COLLEGE SCIENCES
DOCTORAL DE L'INGENIERIE
PAYS DE LA LOIRE ET DES SYSTEMES

CENTRALE NANTES

# MÉMOIRE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE N° 602
*Sciences de l'Ingénierie et des Systèmes*
Spécialité : *Signal, Image, Vision*

Par
**Maria SARKIS**

## Spatial-aware Graph-based Representations for EEG Signals Classification.

**Projet de recherche doctoral présenté et soutenu à l'École Centrale de Nantes,**
**le 21 Novembre 2025**
**Unité de recherche : UMR 6004, Laboratoire des Sciences du Numérique de Nantes (LS2N)**

### Rapporteurs avant soutenance :

Sophie Achard — Directrice de recherche, CNRS – Université Grenoble Alpes
Laurent Oudre — Professeur des universités, École Normale Supérieure Paris-Saclay, Gif-sur-Yvette

### Composition du Jury :

| | | |
|---|---|---|
| Président : | François Cabestaing | Professeur des universités, Université de Lille |
| Examinateurs : | Sophie Achard | Directrice de recherche, CNRS – Université Grenoble Alpes |
| | Laurent Oudre | Professeur des universités, École Normale Supérieure Paris-Saclay, Gif-sur-Yvette |
| | Isabelle Merlet | Chargée de recherche, HDR, Université de Rennes 1 |
| | Sandy Rihana | Professor, Université Saint-Esprit de Kaslik, Liban |
| Directeur de recherches doctorales : | Saïd Moussaoui | Professeur des universités, LS2N, École Centrale Nantes |
| Co-enc. de recherches doctorales : | Mira Rizkallah | Maître de conférences, LS2N, École Centrale Nantes |

### Invité

Vincent Roualdes — Praticien hospitalier, Neurochirurgien, CHU Nantes

# Acknowledgements

Standing at the end of these three years of research, challenges, and growth, I feel truly grateful for the journey and for all the people who supported and inspired me.

To my supervisors: Saïd Moussaoui and Mira Rizkallah, I feel incredibly fortunate and deeply honored to have been supervised by both of you. Thank you for all the opportunities you gave me and for all the trust you have put in me.
Saïd, thank you for your scientific rigor, your guidance, and for teaching me how to take distance from a problem to approach it more clearly and creatively, an attitude that has helped me both in research and in life.
Mira, your energy, patience, kindness, and encouragement have always pushed me to give the best of myself. Thank you for coaching me in so many ways: graphs, teaching, positivity and a lot more.

I would like to express my sincere gratitude to Sophie Achard and Laurent Oudre for accepting to review my thesis and for taking the time to evaluating it carefully, thank you for your constructive comments, and insightful suggestions, which made the reflection more rich. My heartfelt thanks also go to Sandy Rihana, Isabelle Merlet, François Cabestaing, and Vincent Roualdes. I am deeply honored to have such a high quality jury evaluating my thesis. I thank Aurelien Vanlanghenhove for the time he dedicated to my work, for introducing me to EEG and BCI, for the valuable EEG and coding discussions. Our collaborations on projects and teaching were very enriching, and his interest in the field helped to strengthen my own.

I was fortunate to complete my thesis in the exceptional environment of LS2N, surrounded by brilliant and supportive colleagues.
To my office mates Louise and Nils, thank you for sharing the everyday highs and lows, from coding issues to crashing PCs or hard drives. And I'm sorry to say it, but you will still have to call me Dr. Sarkis, even if we're friends and share the same office :p

To my "half-brother" Younes and "half-sister" Dr. Oriane, the organization masters, thank you for making every administrative task and travel preparation so much easier.

You really made me feel like a princess through all the paperwork. Thank you Mehdi for your vibes and your support, your constructive comments and your creative "stress-relief" ideas (nerf gun, spray, lollipop), and sorry I still don't understand all your jokes.

To Dalal, Étienne, Hazel, Roxane, Pierre, Antoine, Gwen, Loriane, Modan, Naser, Osimone, Aubin, Alban- Felix and all the coffee-break friends, thank you for the laughs, the non sense discussions, and the much-needed moments away from the screen. Selma, Nisreen, Yasmine, Vanessa, Charlotte, Mathilde thank you for the amazing food and best afterworks.

To all the PhD students, alumni and permanent members of the SIMS team, I feel truly lucky to have been surrounded by such talented researchers. A special thanks to Barbara Pascal for the rich and stimulating discussions on complex mathematical problems.

Special thanks to the administrative team of LS2N: Laurence, Khrystyna, Sabine, Patrick, Christine, Amandine, Denis. Who handled all the paper work and PC disasters I had during those 3 years. Thank you Patrick Roustang for the scientific reporter experience I had in addition to my research work!

Thank you for all the colleagues in teaching, I learned a lot from you, and to the interns (now PhD students) that I worked with during my thesis: Gracia and Toufic it was really a pleasure I wish you all the best in your respective thesis. My thanks also go to all the students I had the chance to collaborate with: Clément, Xin, Felipe, Guilherme, Diane, Orane, Ilhem, Houda, Zeinab, Dounia, Dina, Hamza, Ilyes, Manil. It was a privilege to accompany each of you in your work.

To my bestfriend back home Marie-Joe, cheers to celebrating every step in life together. Pascale, Carole, Nicole, Pamela, Nay, Kristy, Tia, Marcellin, Sally, Jana, Rita and all my cousins and friends in Lebanon. Can't wait to celebrate this with you.

To my father Assaf, my sister Pia, and my brother Charbel: thank you for being my backbone through every step that led me here, and for the unconditional support especially when I was studying all day all night. Your constant encouragement, your belief in me and your presence in my life are a great blessing.
Mario, thank you for all the love, for standing by my side, caring, helping, reading my papers, and even trying to understand my work. Thank you for being my safe place. I am beyond grateful to have you in my life.

*I dedicate this thesis to the strongest woman I know, my beloved mother, Sabah, who left this world a few months before the defense. She was my guiding light, showing me how to move forward despite the pain of losing her. Keeping her by my side was beyond my control, but making her proud was something I could still choose and this is what gave me the strength to complete this thesis.*

*Mom, I love you.*

# Contents

**Résumé en français** viii

**Notations** xxi

**Acronyms** xxv

**I Introduction and context** 1

**1 Introduction** 3

**2 EEG Signals Acquisition and Analysis for BCI Applications** 9
2.1 Brain activity . . . 10
2.2 Electroencephalography . . . 12
2.3 Brain computer interface . . . 15
2.4 EEG classification framework . . . 15
2.5 Conclusion . . . 23

**3 Foundations of Graph-based modeling for EEG Analysis** 25
3.1 Graph theory and concepts . . . 26
3.2 Graph neural networks . . . 34
3.3 Graph representations for EEG . . . 35
3.4 Conclusion . . . 42

**4 EEG Datasets** 45
4.1 COG-BCI for mental workload classification . . . 46
4.2 HG-MI for mental motor imagery classification . . . 50
4.3 Conclusion . . . 52

**II Graph-Based Representation and Classification** 53

**5 Functional Graph to Image Representation** 55
5.1 Introduction . . . 56
5.2 A graph image classification framework . . . 60

5.3 Experimental setup . . . 67
5.4 Experimental results and analysis . . . 68
5.5 Conclusion . . . 74

**6 Graph Force Learning for EEG Classification 77**
6.1 Force-related work . . . 78
6.2 Hybrid physics data driven graph force learning framework . . . 84
6.3 Experimental results and analysis . . . 93
6.4 Conclusion . . . 104

**7 Brain Connectivity Classification using Graph Neural Networks 107**
7.1 Graph neural networks in EEG analysis . . . 108
7.2 Proposed EEG-GNN framework . . . 114
7.3 Experimental results and analysis . . . 124
7.4 Conclusion . . . 135

**III Conclusions and perspectives 137**

**IV Supplementary material 143**

**A Common Spatial Patterns 144**

**B Per-subject results using all methods 145**

**Bibliography 160**

# Résumé en français

Ce manuscrit présente les travaux de thèse réalisés au Laboratoire des Sciences du Numérique de Nantes (LS2N) au sein de l'équipe Signal, Image et Son (SIMS). Ils portent sur le développement de représentations basées sur des graphes intégrant l'information spatiale pour la classification de signaux électroencéphalographiques (EEG). Ce travail a été soutenu par une bourse doctorale, Contrat Doctoral Etablissement (CDE-ECN) et financé par la Région Pays de la Loire dans le cadre du projet PULSAR (2022-ECN-RIZKALLAH BrainMAGIC).

## Contexte

L'EEG est une technique de mesure non invasive de l'activité cérébrale. Elle consiste à placer des électrodes sur le cuir chevelu afin d'enregistrer l'activité électrique des populations neuronales sous-jacentes. Elle est largement utilisée en raison de sa facilité de mise en œuvre, de son coût relativement faible et de sa résolution temporelle élevée comparée à d'autres modalités d'imagerie. Au cours des dernières années, les interfaces cerveau–ordinateur (ICO) ont suscité un intérêt croissant en tant qu'outils puissants pour interpréter l'activité du système nerveux central et permettre une communication directe entre le cerveau et des dispositifs externes. En enregistrant et en traitant en continu les signaux neuronaux, elles traduisent des activités mentales, telles que l'imagerie motrice, en commandes de contrôle, offrant ainsi un retour en temps-réel sans recourir au système nerveux périphérique. Cette technologie ouvre des perspectives prometteuses, tant pour des applications d'assistance que pour des approches thérapeutiques.

L'utilisation d'un système ICO repose généralement sur deux étapes : (i) une phase hors-ligne, consacrée au calibrage du modèle, et (ii) une phase en-ligne, où les signaux EEG enregistrés par des électrodes sont analysés pour identifier des motifs cérébraux spécifiques et les convertir en commandes exploitables. Ces motifs peuvent être volontairement produits par l'utilisateur, par exemple en imaginant une tâche motrice ou dans le cadre d'autres paradigmes d'interaction. Le système procède alors à des tâches de classification, telles que la reconnaissance du type d'imagerie motrice ou l'évaluation de la charge mentale, permettant de fournir à l'utilisateur un retour adapté dans l'application considérée, qu'il s'agisse de rééducation, d'assistance ou d'interaction homme–machine.

Dans la littérature, les dispositifs d'analyse de l'EEG suivent généralement une chaine de traitement bien établie, comprenant une phase de prétraitement, puis une étape d'extraction de caractéristiques avant la classification. Ces caractéristiques visent à représenter différents aspects de l'activité cérébrale : dans le domaine fréquentiel, elles reposent par exemple sur la puissance dans des bandes spécifiques ; dans le domaine temps–fréquence, elles exploitent des représentations dynamiques obtenues à l'aide d'ondelettes ou de transformées de Fourier à court terme ; enfin, dans le domaine spatial, elles s'appuient sur des projections comme les *Common Spatial Patterns* (CSP). La classification est ensuite réalisée à l'aide de méthodes supervisées, allant des approches classiques telles que les machines à vecteurs supports (SVM), l'analyse discriminante linéaire (LDA) ou les forêts aléatoires, jusqu'aux méthodes les plus récentes d'apprentissage profond, capables d'apprendre directement des représentations discriminantes à partir des signaux bruts.

Au-delà de ces déscripteurs étudiant l'activité mesurée par électrode, un axe de recherche important consiste à étudier la connectivité cérébrale, qui capture les interactions fonctionnelles entre régions du cerveau et offre une description plus riche des mécanismes neuronaux sous-jacents. À partir des signaux EEG, cette connectivité est généralement estimée par des mesures d'interdépendance statistique entre électrodes, telles que la corrélation, l'information mutuelle ou la cohérence de phase (PLV). Ces méthodes aboutissent à des matrices de similarité représentant les relations par paires entre électrodes, mais nécessitent souvent l'introduction de seuils pour éliminer les connexions considérées comme non pertinentes.

Le traitement du signal sur graphe (Graph Signal Processing, GSP) offre un cadre mathématique puissant pour modéliser des réseaux neuronaux. Dans le cas de l'interaction entre régions du cerveau mesuré par l'EEG, cette approche consiste à représenter les électrodes comme des nœuds et leurs relations comme des arêtes pondérées. La structure du graphe peut être définie de différentes manières, par exemple à partir des distances physiques entre électrodes ou à partir de mesures de connectivité fonctionnelle dérivées des signaux. En outre, des caractéristiques propres à chaque électrode, telles que des descripteurs fréquentiels ou topologiques, peuvent être intégrées au niveau des nœuds, enrichissant ainsi la représentation et permettant une analyse plus fine des dynamiques cérébrales.

## Objectifs de la thèse

Malgré ces avancées, l'exploitation explicite de la configuration spatiale des électrodes demeure relativement peu étudiée dans le cadre de la connéctivité cerebrale, alors qu'elle joue un rôle essentiel dans la compréhension de l'organisation fonctionnelle du cerveau.

L'objectif principal de cette thèse est donc de proposer un cadre d'analyse de la connec-

tivité cérébrale intégrant explicitement la disposition spatiale des électrodes pour l'analyse et la classification des signaux EEG. Plus précisément, cet objectif peut être résumé à travers les questions de recherche suivantes :

I. **Apprentissage du graphe :** Comment construire un graphe à partir des données EEG qui reflète la connectivité cérébrale globale, et pas seulement des relations par paires d'électrodes ?

II. **Intégration spatiale :** Une fois ce graphe défini, comment incorporer l'information spatiale liée à la position des électrodes dans la représentation ?

III. **Sélection des informations :** Quels types de descripteurs complémentaires (fréquentiels, spatiaux, topologiques, etc.) doivent être choisis pour enrichir le graphe ?

IV. **Utilisation pour la classification :** Comment exploiter efficacement ce graphe complet dans un cadre de classification des états cérébraux ?

V. **Explicabilité et visualisation :** Existe-t-il des moyens d'expliquer les résultats obtenus et de relier les représentations apprises aux mécanismes neuronaux sous-jacents ?

De plus, à titre d'exemple d'application potentielle, le Centre Hospitalier Universitaire (CHU) de Nantes, en collaboration avec le LS2N, a développé un système de réalité virtuelle piloté par une interface cerveau–ordinateur destiné à accompagner les patients souffrant de douleurs du membre fantôme. En décodant, à partir des signaux EEG, l'imagerie motrice du membre manquant ou affecté et en la traduisant en un retour visuel immersif en temps réel, ce système favorise la plasticité cérébrale grâce à un entraînement mental guidé et contribue à soulager la douleur. Les représentations EEG basées sur les graphes et les stratégies d'apprentissage développées dans cette thèse pourraient améliorer ce type de systèmes en renforçant le décodage de l'imagerie motrice ainsi que la surveillance de la charge mentale, afin d'adapter le processus d'entraînement et de faciliter la progression des séances de rééducation.

# Organisation du manuscrit

Ce manuscrit se structure en deux grandes parties : la première regroupe trois chapitres introductifs qui posent les bases théoriques et méthodologiques des travaux présentés.
**Chapitre 2** présente le contexte général de l'étude, en introduisant les principes de l'électroencéphalographie et des interfaces cerveau–ordinateur, ainsi que les méthodes classiques de traitement du signal et d'apprentissage automatique utilisées dans la littérature pour la classification des signaux EEG.
**Chapitre 3** propose un aperçu détaillé du cadre théorique du traitement de signal sur graphes et des réseaux de neurones sur graphes (GNN), qui constituent la base des approches explorées dans cette thèse. Il inclut également un état de l'art des méthodes

basées sur les graphes appliquées aux signaux EEG. Une attention particulière est portée aux techniques de construction de graphes que nous adaptons et étendons dans nos contributions afin de répondre à la question **I**.
**Chapitre 4** décrit les jeux de données publics utilisés tout au long de ce travail, en détaillant leurs protocoles d'acquisition, leurs paradigmes expérimentaux et les étapes de prétraitement appliquées. Il présente également les méthodes de référence auxquelles nos approches sont comparées, ainsi que leurs résultats rapportés, fournissant ainsi un cadre de référence clair pour l'évaluation des méthodes proposées.
La deuxième partie rassemble les trois chapitres de contributions, chacun visant à répondre aux questions de recherche formulées précédemment. Plus précisément, ces chapitres sont organisés de la manière suivante :

**Chapitre 5** introduit un nouveau cadre pour la classification de la connectivité fonctionnelle issue des signaux EEG. Cette approche répond d'abord à la question **I** en apprenant un Laplacien généralisé du graphe sous contraintes structurelles et de parcimonie, offrant ainsi une représentation de connectivité plus informative que les simples corrélations de Pearson par paires. Pour traiter la question **II**, le graphe appris est transformé en une représentation fonctionnelle *graph-to-image* (*FG2IMG*) qui préserve la disposition spatiale des électrodes, leur importance au sein du graphe ainsi que les poids de connectivité. Ces images sont ensuite exploitées par un réseau de neurones convolutionnel (CNN) pour la classification, répondant ainsi à la question **IV**. Le cadre proposé démontre des performances compétitives : il dépasse les méthodes classiques telles que la distance minimale riemannienne à la moyenne (R-MDM) et s'approche de la précision obtenue avec CSP pour l'imagerie motrice, tout en surpassant clairement CSP et R-MDM pour la classification de la charge mentale. Enfin, ce cadre contribue à la question **V** en améliorant l'interprétabilité, puisqu'il permet la visualisation et l'analyse du comportement du modèle ainsi que des motifs de connectivité fonctionnelle spécifiques aux sujets, confirmant ainsi les avantages de l'apprentissage de la connectivité fondée sur le Laplacien par rapport aux mesures de corrélation traditionnelles.

**Chapitre 6** introduit une seconde chaîne complète de traitement, construite à partir du graphe appris au Chapitre 3. Notre approche répond à la question **II** en modélisant le graphe à travers une représentation inspirée de la physique : les électrodes sont considérées comme des particules soumises à des forces interactives, calculées à partir de leurs positions spatiales et de leur connectivité. Nous étendons ce concept en proposant une formulation généralisée des forces en configurations 2D et 3D, où des paramètres tels que la décroissance avec la distance et l'échelle des forces peuvent être ajustés afin d'optimiser les performances. Pour répondre à la question **III**, les caractéristiques nodales issues de cette modélisation par les forces sont enrichies par des mesures complémentaires issues de

la topologie du graphe et des signaux, constituant ainsi un ensemble de caractéristiques étendu. Ces représentations sont ensuite classifiées à l'aide de méthodes traditionnelles, notamment les SVM, répondant ainsi à la question **IV**. Enfin, cette modélisation par les forces contribue à la question **V** en fournissant des représentations visuellement interprétables et explicables des motifs de connectivité cérébrale sous-jacents. L'approche proposée atteint des performances compétitives sur le jeu de données d'imagerie motrice par rapport aux méthodes de référence et les surpasse dans le cas de la classification de la charge mentale.

**Chapitre 7** introduit une troisième chaîne de traitement fondée sur la modélisation des signaux EEG sous forme de graphes exploitant conjointement l'information portée par les arêtes et par les nœuds. Les arêtes sont définies soit à partir de mesures de connectivité fonctionnelle, soit à partir des distances physiques modulées par ces mesures de connectivité, tandis que les nœuds sont enrichis par des caractéristiques spectrales, spatiales et topologiques. Par ailleurs, ce chapitre compare différentes manières d'intégrer l'information spatiale au niveau des nœuds ou des arêtes en proposant plusieurs stratégies pour coder les relations spatiales entre électrodes. Nous comparons notamment les distances géodésiques sur des surfaces sphériques et ellipsoïdales aux distances cartésiennes projetées sur un plan azimutal 2D, et combinons ces mesures de distance avec diverses fonctions de décroissance pour moduler les poids des arêtes, répondant ainsi aux questions **II** et **III**. Une fois le graphe construit, des réseaux de neurones sur graphes (GNN) sont utilisés pour traiter directement ces représentations dans une tâche de classification EEG, répondant ainsi à la question **IV**. Enfin, il est montré que la modélisation des signaux EEG sous forme de graphes intégrant à la fois la structure spatiale ainsi que des caractéristiques spectrales et topologiques améliore significativement les performances de classification par rapport aux méthodes de référence dans le domaine spatial, tout en offrant une compréhension plus riche des relations spatiales et fonctionnelles sous-jacentes à l'activité cérébrale.

# Publications issues de la thèse

Les travaux de la thèse ont été valorisés à travers des communications suivantes:

— **Chapitre 5:**

- Communication à la conférence internationale *IEEE Engineering in Medicine and Biology Society (EMBC)* qui s'est tenue à Orlando (Floride), en juillet 2024 [Sarkis et al. 2024b].
- Poster présenté au *Colloque Français d'Intelligence Artificielle en Imagerie Biomédicale (IABM)*, qui s'est tenue à Grenoble, en mars 2024

— **Chapitre 6:**

- La première partie de ce travail a donné lieu à une communication au workshop international *Statistical Signal Processing (SSP)*, qui s'est tenu à Édimbourg en juin 2025 [Sarkis et al. 2025c].
- La seconde partie de ce travail fait actuellement l'objet d'une relecture pour publication dans la revue internationale *IEEE Signal Processing Letters (SPL)* [Sarkis et al. 2025a].

— **Chapitre 7:**

- Communication présentée au workshop international *Graph Signal Processing (GSP)*, qui s'est tenu à Delft en juin 2024 [Sarkis et al. 2024a]. ;
- Communication présentée au *Colloque National de Traitement du Signal (GRETSI)*, qui s'est tenu à Strasbourg en août 2025 [Sarkis et al. 2025b].
- Poster présenté au *Colloque Français d'Intelligence Artificielle en Imagerie Biomédicale (IABM)*, qui s'est tenu à Nice en mars 2025.

# List of Figures


2.1 Overview of the Brain different lobes. . . . . 11

2.2 The 10-20 system with front-back (nasion to inion) 10% and 20% electrode distances [DocCheck 2023] . . . . . 13

2.3 Overview of a brain–computer interface system. Brain activity is recorded through electroencephalography and processed through a signal-processing pipeline that includes preprocessing, feature extraction, and classification. The resulting outputs are translated into feedback or control commands depending on the application, such as neurorehabilitation, neuroprosthesis control, or neurogaming. . . . . 16

3.1 Examples of different graph structures with 4 vertices with a self loop on vertex 1: (a) unweighted undirected, (b) directed, and (c) weighted with self-loops, along with their corresponding adjacency and weight matrices. . 27

3.2 An example of $3$-hop neighborhood of a vertex $u$ . . . . . 28

3.3 A graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$; with a set of vertices $\mathcal{V} = \{0, 1, 2, 3, 4, 5, 6, 7\}$ 29

3.4 The left diagram illustrates a regular 2D grid, typical of image data processed by CNN, where each node has a fixed neighborhood. The right diagram represents an irregular graph structure, where nodes are connected based on domain-specific relationships, as used in GNN. . . . . 34

3.5 Message passing process: vertex $u$ receives messages from its $\mathcal{N}_1$ neighbours if one message passing layer is used. These messages, derived from the feature vectors of the neighboring nodes, are then aggregated to update the representation of node $u$. . . . . 35

4.1 The MATB-II tasks extracted from [https://matb.larc.nasa.gov] . . . . . 48

4.2 Decomposing the signal into groups of overlapping windows for COG-BCI dataset. . . . . 49

5.1 Overview of the proposed classification framework with three main modules a) a graph generation module where a multivariate EEG signal is acquired, pre-processed, decomposed into windows from which functional graphs are learned, b) a Graph to Image module where the functional graph is represented by a 2D low resolution image, c) a classification module where a latent representation is learned using a convolutional neural network for a classification task. . . . . 61

5.2 Proposed image creation as a matrix representation of the electrodes spatial positions coupled with Bresenham line algorithm highlighting position, degree and connectivity of the graph. . . . . 62

5.3 Example of a projection of EEG electrode coordinates onto a $61 \times 61$ grid, where red squares indicate the electrode positions in the Euclidean space and the black markers denote the grid points. . . . . 63

5.4 Example of a functional connectivity graph obtained from a selected time window of Subject 7, session 2 (COG-BCI dataset), plotted using NetworkX Python library where edge weights are computed from the Laplacian; edge color and thickness reflect the learned connection strength. . . . . 65

5.5 FG2IMG representation of the functional connectivity graph for a selected time window of Subject 7 Session 2 (COG-BCI dataset). Left: full image including the zero-valued background; right: the background is masked to highlight edge details. . . . . 66

5.6 Example of a functional connectivity graph obtained from a selected time window of Subject 5 in the MI-HG dataset, plotted using NetworkX Python library where edge weights are computed from the Laplacian; edge color and thickness reflect the learned connection strength. . . . . 66

5.7 FG2IMG representation of the functional connectivity graph for a selected time window of Subject 5 (MI-HG dataset). Left: full image including the zero-valued background; right: the background is masked to highlight edge details. . . . . 67

5.8 Convolutional neural network architecture for image-based EEG graph classification. The input image is processed through three convolutional layers with ReLU activation and max pooling, followed by fully connected layers that output the predicted class. . . . . 67

5.9 Visualization of Grad-CAM activation maps across task difficulty levels (easy, medium, and difficult) for subjects chosen randomly from COG-BCI dataset. . . . . 73

6.1 Spring–mass system analogy. Starting from a random configuration, particles linked by springs are iteratively displaced by spring forces until the system converges to a stable, low-energy equilibrium [Kobourov 2012]. . . . 80

6.2 Overview of the proposed graph force learning framework: EEG signals are pre-processed and segmented into overlapping windows of $tF_s$ samples. For each window, a graph is constructed. Using the spatial electrode positions and the graph structure, the forces exerted on each node are then calculated following a formulation in 2D or 3D. The u-, v-, and k-components of these forces are concatenated into a feature vector for each window, which are then classified using an SVM classifier. . . . . 85

6.3 Attractive force between two connected nodes $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_a^{ef}$. Here, $\mathbf{W}_{ef}$ is the connection weight and $d_{ef}$ the Euclidean distance between the two nodes. . . . . 87

6.4 Repulsive force between two unconnected nodes $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_r^{ef}$. Here, $\mathbf{D}_{ee}$ and $\mathbf{D}_{ff}$ are the node degrees, and $d_{ef}$ is the Euclidean distance between the two nodes. . . . . 88

6.5 Example of the resultant 2D force acting on each electrode, expressed as the sum of all attractive and repulsive forces applied to the node. . . . . 88

6.6 Attractive force between two connected vertices $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_a^{ef}$. Here, $\mathbf{W}_{ef}$ is the connection weight, $d_{ef}$ the Euclidean distance between the vertices, and $k_a$ a scaling factor. . . . . 90

6.7 Repulsive force between two unconnected vertices $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_r^{ef}$. Here, $\mathbf{D}_{ee}$ and $\mathbf{D}_{ff}$ are the node degrees, $d_{ef}$ the Euclidean distance, and $k_r$ a scaling factor. . . . . 91

6.8 Example of the resultant 3D force acting on electrodes expressing the sum of attractive and repulsive forces. . . . . 91

6.9 Proportion of selected force, PSD, and structural features for varying total feature counts for COG-BCI dataset. As fewer features are selected, the proportion of force features increases, suggesting their higher discriminative power. Mean classification accuracy with standard deviation is also reported. 99

6.10 Proportion of selected force, PSD, and structural features for varying total feature counts for HG-MI dataset. As fewer features are selected, the proportion of force features increases, suggesting their higher discriminative power. Mean classification accuracy with standard deviation is also reported. 99

6.11 Impact of attractive and repulsive force computation parameters on classification accuracy for subject 12 S2 of COG-BCI dataset. (left) Power parameters $(\lambda_a, \lambda_r)$, (right) scaling parameters $(k_a, k_r = 1 - k_a)$. . . . . 100

6.12 Impact of attractive and repulsive force computation parameters on classification accuracy subject 7 of HG-MI dataset. (left) Power parameters $(\lambda_a, \lambda_r)$, (right) scaling parameters $(k_a, k_r = 1 - k_a)$. . . . . 101

6.13 Topographic maps of EEG-derived features for Participant 1, Session 1, across three workload conditions (easy, medium, difficult). The first row shows the spatial distribution of resultant force magnitudes obtained from the force-based graph representation. The second and third rows display the mean spectral energies in the theta and alpha bands, respectively. . . 102

6.14 Visualization of force-based features for Participant 1, Session 1, across three workload levels (easy, medium, difficult). (Left) Force topography maps showing the spatial distribution of the force magnitudes $||\mathbf{F}||$ over the scalp. (Right) Corresponding spring layout graph representations, where node color encodes the force magnitude. . . . . 103

7.1 From each EEG window, a connectivity graph is learned. Frequency-based features, possibly combined with spatial positions and a topological feature, are used to define the node attributes. The edges reflect either functional connectivity or the inverse of the geodesic distance. The resulting graphs are then processed by a graph neural network (GNN) for classification. . . 114

7.2 Example of a graph with the original coordinates of the electrodes; connections are drawn as lines corresponding to the Cartesian distances between electrodes . . . . . 117

7.3 Illustration of the azimuthal equidistant mapping of the 61 electrodes from the COG-BCI dataset. left: conventional MNE-like projection, right: projection generated by our approach. . . . . 118

7.4 A point on a sphere has coordinates $\bar{\mathbf{c}}_e$. The sphere radius, the azimuth and colatitude are denoted as $r_e$, $\bar{\theta}_e$ and $\bar{\phi}_e$, respectively. . . . . 119

7.5 Example of a graph where the electrodes ,*i.e.* the nodes, are projected onto a sphere with a radius $r$. . . . . 120

7.6 Plot of the Euclidean norms of the electrode positions after centering. The variation in distances suggests that a spherical geometry may not be optimal for representing the head. . . . . 120

7.7 Ellipsoid of revolution properties . . . . . 121

7.8 Projection of electrodes on ellipsoid of revolution . . . . . 122

7.9 GNN model architecture consisting of two GraphConv layers, a global max-pooling layer, a dropout and linear layer. . . . . 124

7.10 Pairwise Euclidean distances in 3D vs. 2D Cartesian coordinates on the COG-BCI dataset. Left panels: histograms of distances with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right panels: corresponding distance matrices displayed with a shared color scale; the colorbar indicates $d$ in cm 129
7.11 Pairwise Euclidean distances in 3D vs. 2D Cartesian coordinates on the HG-MI dataset. Left panels: histograms of distances with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right panels: corresponding distance matrices displayed with a shared color scale; the colorbar indicates $d$ in cm 130
7.12 Spherical/ellipsoidal geodesic distances and sphere-projected Cartesian distances on the COG-BCI dataset. Left column: histograms of pairwise distances $d$ with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right column: corresponding distance matrices displayed with a shared color scale across rows; the colorbar indicates $d$ in cm. . . . . 132
7.13 Spherical/ellipsoidal geodesic distances and sphere-projected Cartesian distances on the HG-MI dataset. Left column: histograms of pairwise distances $d$ with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right column: corresponding distance matrices displayed with a shared color scale across rows; the colorbar indicates $d$ in cm. . . . . 133
7.14 Visual comparison of different electrode projection schemes for the HG-MI dataset (44 channels covering the cortex zone): (a) original 3D positions; (b) MNE-style azimuthal projection onto the 2D plane; (c) spherical projection; and (d) ellipsoidal projection. . . . . 134

B.1 Results per subject of all the proposed approaches on HG-MI dataset . . . 145
B.2 Results per subject of all the proposed approaches on COG-BCI S1 . . . . 146
B.3 Results per subject of all the proposed approaches on COG-BCI S2 . . . . 147

# List of Tables


3.1 Graph metrics for each node of the weighted graph $\mathcal{G}$ in [Fig. 3.3]. . . . . 31
3.2 Summary of common EEG connectivity measures. . . . . 37

4.1 HG-MI dataset: number of epochs per class and participant. . . . . 51

5.1 Overview of deep learning approaches for EEG classification. . . . . 60
5.2 COG-BCI dataset [Hinss et al. 2023] classification accuracy (mean ± std; 5-fold CV) across 29 participants for two sessions. Best result per row is **bold**. . . . . 70
5.3 HG-MI dataset [Schirrmeister et al. 2017] classification accuracy (mean ± std; 5-fold CV) across 14 participants for different approaches used with FG2IMG and classical baseline results. Only results on 80/20 % splits for baseline approaches are available. Best result per row is **bold**. . . . . 70
5.4 COG-BCI [Hinss et al. 2023] dataset classification accuracy (mean ± std; 5-fold CV) across 29 participants for two sessions, using Riemannian-MDM variants, CSP-OVR [W. Wu et al. 2006], and the proposed FG2IMG approach under different train/test splits. Best result per row is **bold**. . . . . 72

6.1 Overview of the Force-directed models and their force formulations. . . . . 83
6.2 SVM classification accuracy (mean ± std) for different node feature representations, graph structure is computed using the Learned Laplacian. Best results per dataset are in **bold**. . . . . 94
6.3 SVM classification accuracy (mean ± std) for different graph construction strategies and node feature representations across two datasets. Best results per column for each dataset is **bold** . . . . . 95
6.4 Classification accuracy (mean ± std) of different algorithms across graph construction strategies. Best results per row and column are **bold** . . . . . 96
6.5 Comparison against baseline approaches for EEG classification. Best results in column is **bold**. . . . . 97
6.6 Classification accuracy (mean ± std) of different signal-related feature sets on the two datasets. . . . . 98

7.1 Summary of the reviewed methods for EEG graph construction and GNN modeling. . . . . 113
7.2 Comparison of connectivity-driven graph construction strategies (mean ± std). . . . . 126
7.3 Classification accuracy (mean ± std) using different distance decay functions and spatial representations (2D vs. 3D Cartesian). Node features are fixed to topological and spectral features. . . . . 128
7.4 Classification accuracy (mean ± std) using different distance decay functions applied to geodesic distances under various projection types. Node features are fixed to topological and spectral features. . . . . 131
7.5 Comparison of node and edge attribute configurations (mean ± std). . . . 132
7.6 Comparison with state-of-the-art methods using spatial features (mean ± std). . . . . 135

# Notations

## Notation

| | |
|---|---|
| $\mathbf{X} \in \mathbb{R}^{n \times tF_s}$ | Multivariate EEG signal |
| $\mathbf{x_e}$ | EEG signal measured by electrode e |
| $\mathsf{x}, \mathsf{y}, \mathsf{z}$ | Cartesian coordinates of a node |
| $X, Y, Z$ | Random variables |
| $\mathbf{A}$ | Adjacency matrix of the graph |
| $\mathbf{W}$ | Weight matrix |
| $\hat{\mathbf{W}}$ | Normalized weight matrix |
| $\mathbf{L}$ | Graph Laplacian |
| $\mathbf{L}_{\text{norm}}$ | Normalized Laplacian |
| $\mathbf{D}$ | Weighted degree matrix |
| $\mathbf{deg}$ | Degree matrix of the unweighted graph |
| $\mathbf{S}$ | Covariance matrix |
| $\mathbf{\Omega}$ | Precision matrix |
| $\mathbf{I}$ | Identity matrix |
| $\bar{\mathbf{h}}_u$ | Feature vector of node $u$ |
| $\mathbf{h} \in \mathbb{R}^{n \times 1}$ | a signal on graph |
| $h(u)$ | Value of a graph signal $\mathbf{h}$ at node $u$ |
| $\hat{h}(l)$ | Projection of $\mathbf{h}$ onto the $l^{\text{th}}$ eigenvector |
| $\hat{h}(l)^2$ | Energy of the signal on the $l^{\text{th}}$ eigenvector |
| $\mathbf{H}$ | Regularization matrix |
| $\bar{\mathbf{H}}^{(\mathbf{0})}$ | Feature matrix at layer 0 |
| $\mathbf{u}_u$ | Eigenvector associated with vertex $u$ |
| $\lambda$ | Graph Laplacian eigenvalue |
| $\phi_{\mathbf{x}_e}(k)$ | Instantaneous phase of signal $\mathbf{x}_e$ at time $k$ |

| | |
|---|---|
| $\bar{f}$ | Frequency bin |
| $PSD_{\mathbf{x}_e}^{(\bar{f})}$ | Power spectral density of signal $\mathbf{x}_e$ at $\bar{f}$ |
| $G_{\mathbf{x}_e\mathbf{x}_f}^{(\bar{f})}$ | Cross-spectral density between $\mathbf{x}_e$ and $\mathbf{x}_f$ at frequency bin $\bar{f}$ |
| $\mu_e$ | Mean of signal $\mathbf{x}_e$ |
| $tF_s$ | Number of time samples ($t$ duration $\times$ sampling frequency $F_s$) |
| $\alpha_0$ | Positive constant (regularization weight) |
| $\zeta$ | Sparsity-controlling parameter |
| $T$ | Threshold value |
| $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ | Graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ |
| $\mathcal{N}_k(u)$ | $k$-hop neighbourhood of node $u$ |
| $d(u, v)$ | Shortest path length between nodes $u$ and $v$ |
| $CC_u$ | Closeness centrality of vertex $u$ |
| $CB_u$ | Betweenness centrality of vertex $u$ |
| $\#d(s, t)$ | Total shortest paths between $s$ and $t$ |
| $\#d(s, t)_v$ | Total shortest paths between $s$ and $t$ passing through $v$ |
| $C_u$ | Clustering coefficient of vertex $u$ |
| $\mathcal{Y}$ | Set of different classes |
| $\mathcal{L}(\mathbf{A})$ | Set of admissible adjacency matrices |
| $\Psi(\cdot)$ | Aggregation function |
| $P(\mathbf{X} \mid \mathcal{G})$ | Likelihood of data $\mathbf{X}$ given graph $\mathcal{G}$ |
| $P(\mathbf{\Omega})$ | Prior distribution on graph structures |
| $P(\mathbf{X})$ | Marginal likelihood (normalizing constant) |
| $\mathbf{\Omega} \succ 0$ | Positive-definite constraint on $\mathbf{\Omega}$ |
| $\mathrm{sgn}(a - b)$ | Sign of the difference $a - b$ |
| $(\cdot)^T$ | Transpose |
| $(\cdot)^*$ | Complex conjugate |
| $\|\cdot\|_1$ | Element-wise $\ell_1$ norm (Laplace prior) |
| $\|\cdot\|_F$ | Frobenius norm |
| $\|\cdot\|$ | Module |
| $\|\|$ | Concatenate |
| $\delta^{EEG}$ | Delta rhythm |
| $\theta^{EEG}$ | Theta rhythm |
| $\alpha^{EEG}$ | Alpha rhythm |
| $\beta^{EEG}$ | Beta rhythm |
| $\gamma^{EEG}$ | Gamma rhythm |

# Notation related to chapter 5

| | |
|---|---|
| $\mathcal{C}$ | Set of $(\mathsf{x}, \mathsf{y}, \mathsf{z})$ spatial coordinates associated with a node |
| $\mathbf{M}$ | Two-dimensional image of size $m \times m$ |
| $\Delta\mathsf{x}$ | Distance between two grid elements in the $\mathsf{x}$ direction |
| $\Delta\mathsf{y}$ | Distance between two grid elements in the $\mathsf{y}$ direction |
| $\mathcal{P}$ | List of pixel indices |
| $p_e = \mathbf{M}_{(\mathsf{i}_e, \mathsf{j}_e)}$ | Pixel in the image associated with node or edge element $e$ |
| $\mathcal{D}(p_e, p_f)$ | Set of pixel indices forming the discrete line connecting $p_e$ and $p_f$ |
| $\mathcal{W}_p$ | Value of pixel $p$ in the image |
| $\Delta\mathsf{i}$ | Difference between two pixel indices in the direction of $\mathsf{i}$ |
| $\Delta\mathsf{j}$ | Difference between two pixel indices in the direction of $\mathsf{j}$ |
| $s_\mathsf{i}$ | Sign of the difference of pixel indices in the direction of $\mathsf{i}$ |
| $s_\mathsf{j}$ | Sign of the difference of pixel indices in the direction of $\mathsf{j}$ |
| **3-NN** | Graph with connections only to the three spatially closest neighbors |

# Notation related to chapter 6

| | |
|---|---|
| $\mathbf{F}_e^a$ | Attractive force acting on node e |
| $\mathbf{F}_e^r$ | Repulsive force acting on node e |
| $\mathbf{F}_e$ | Resultant force acting on node e |
| $\beta_{ef}$ | Angle of the force vectors between node e and f in the plane |
| $\beta_F$ | Angle of the resultant force vector |
| $k_s$ | Spring constant |
| $k_a$ | Scaling constant related to attractive force |
| $k_r$ | Scaling constant related to repulsive force |
| $d_0$ | Ideal edge length |
| $\eta_a$ | Constant controlling learning rate of attractive force |
| $\eta_r$ | Constant controlling learning rate for repulsive force |
| $\mathcal{L}(e)$ | Loss function of node e |
| $\bar{\mathsf{x}}$ | Displacement from equilibrium position |
| $l_{e,f}$ | Desired distance between node e and node f |
| $E$ | Global energy function |

| | |
|---|---|
| $\delta$ | Parameter to activate or not the weights in force computation |
| $\psi$ | Similarity function |
| $\mathcal{C}^{2D}$ | Set of $(\mathsf{x}, \mathsf{y})$ spatial coordinates associated with the electrodes or nodes |
| $\lambda_a$ | Power parameter modulating the effect of distance on the attractive force |
| $\lambda_r$ | Power parameter modulating the effect of distance on the repulsive force |
| $\bar{\phi}$ | Polar angle |
| $\bar{\theta}$ | Azimuthal angle |
| $\mathbf{F}_{graph}$ | Feature vector representing the graph node forces features |
| $\gamma_{RBF}$ | Hyperparameter in SVM classifier |

# Notation related to chapter 7

| | |
|---|---|
| $\mathcal{F}(\cdot)$ | Decay function of distance |
| $\mathbf{R}$ | Graph structure encoded as physical distances between nodes |
| $r$ | Radius of sphere |
| $\rho$ | Radial distance |
| $\bar{\mathsf{c}}$ | Projected coordinates |
| $\mathsf{a}, \mathsf{b}$ | Ellipsoid of revolution semi-axes |
| $\xi$ | Flattening of ellipsoid |
| $\epsilon$ | Ellipsoid first eccentricity |
| $\varphi$ | Latitude of a point on an ellipsoid |
| $\Lambda$ | Longitude of a point on ellipsoid |
| $\mathsf{h}$ | Height of a point relative to the ellipsoid surface |
| $\kappa$ | Parameter related to coordinates of a point on the ellipsoid |
| $\mathbf{\Theta}$ | Transformation matrix in GNN |

# Acronyms

| | |
|---|---|
| **AR** | Autoregressive modeling |
| **BCI** | Brain–Computer Interface |
| **BOLD** | Blood-Oxygen-Level-Dependent |
| **BSS** | Blind Source Separation |
| **CAR** | Common Average Referencing |
| **CNN** | Convolutional Neural Network |
| **CNS** | Central Nervous System |
| **COG-BCI** | Cognitive Brain–Computer Interface |
| **COMM** | Communication |
| **CSP** | Common Spatial Patterns |
| **CV** | Cross Validation |
| **DL** | Deep Learning |
| **DSP** | Digital Signal Processing |
| **DTI** | Diffusion Tensor Imaging |
| **ECG** | Electrocardiogram |
| **EEG** | Electroencephalography |
| **EMG** | Electromyogram |
| **EOG** | Electrooculogram |
| **FBCSP** | Filter Bank Common Spatial Pattern |
| **FG2IMG** | Functional Graph-to-Image |
| **FFT** | Fast Fourier Transform |
| **FIR** | Finite Impulse Response Filter |
| **fMRI** | Functional Magnetic Resonance Imaging |
| **GAT** | Graph Attention Network |
| **GCN** | Graph Convolutional Network |
| **GFT** | Graph Fourier Transform |

| | |
|---|---|
| **GGL** | Generalized Graph Laplacian |
| **Grad-CAM** | Gradient-Weighted Class Activation Mapping |
| **GNN** | Graph Neural Network |
| **GIN** | Graph Isomorphism Network |
| **GSP** | Graph Signal Processing |
| **HG-MI** | High Gamma Motor Imagery |
| **ICA** | Independent Component Analysis |
| **IWBW** | Intensity-Weighted Bandwidth |
| **IWMF** | Intensity-Weighted Mean Frequency |
| **JAD** | Joint Approximate Diagonalization |
| **LDA** | Linear Discriminant Analysis |
| **LSTM** | Long Short-Term Memory |
| **MAP** | Maximum A Posteriori |
| **MATB-II** | Multi-Attribute Task Battery-II |
| **OVR** | One-Versus-Rest |
| **PCA** | Principal Component Analysis |
| **PC** | Pearson Correlation |
| **PLI** | Phase Lag Index |
| **PLV** | Phase-Locking Value |
| **PSD** | Power Spectral Density |
| **PSP** | Postsynaptic Potential |
| **PVT** | Psychomotor Vigilance Task |
| **RESMAN** | Resource Management |
| **RF** | Random Forest |
| **R-MDM** | Riemannian Minimum Distance to Mean |
| **RGB** | Red Green Blue |
| **SEF** | Spectral Edge Frequency |
| **SOBI** | Second Order Blind Identification |
| **SPD** | Symmetric Positive Definite |
| **STFT** | Short-Time Fourier Transform |
| **SVM** | Support Vector Machine |
| **SYSMON** | System Monitoring |
| **TCSGSP** | Temporally Constrained Sparse Group Spatial Patterns |
| **VR** | Virtual Reality |
| **XGBoost** | Extreme Gradient Boosting |

Part I

# Introduction and context

# Chapter 1

# Introduction

This manuscript presents the doctoral research carried out in the Laboratoire des Sciences du Numérique de Nantes (LS2N), within the Signal, Image and Sound (SIMS) group. The related topic focuses on the development of graph-based representations that integrate spatial information for the classification of electroencephalographic (EEG) signals. This work was supported by a doctoral fellowship Contrat Doctoral Établissement (CDE-ECN) and funded by the Pays de la Loire Region as part of the PULSAR project (2022-ECN-RIZKALLAH BrainMAGIC).

## Context

Electroencephalography (EEG) is a non-invasive brain sensing technique that involves placing electrodes on the scalp to capture the electrical activity of underlying neuronal populations. It is widely used due to its ease of implementation, relatively low cost, and high temporal resolution compared to other sensing modalities. In recent years, EEG-based brain–computer interfaces (BCIs) have attracted growing interest as powerful tools for interpreting the activity of the central nervous system (CNS) and enabling direct communication between the brain and external devices. By continuously recording and processing neural signals, they translate mental activities, such as motor imagery, into control commands, thus providing real-time feedback without relying on the peripheral nervous system. This technology opens promising perspectives for both assistive applications and therapeutic approaches.

The setting of a BCI system generally involves two stages: (i) an offline phase dedicated to model calibration, and (ii) an online phase, where EEG signals recorded by scalp electrodes are analyzed to identify specific brain patterns and convert them into actionable commands. These patterns can be voluntarily produced by the user, for example, by imagining a motor imagery task or within other interaction paradigms. The system then performs classification tasks, such as recognizing the type of motor imagery or assessing mental workload, to provide the user with appropriate feedback in the given application, whether for rehabilitation, assistance, or human–machine interaction.

In the literature, EEG processing pipelines generally follow a well-established sequence consisting of a pre-processing phase, followed by feature extraction and then classification. The pre-processing step is related to signal cleaning, artifact removal and noise reduction. The extracted features intends to represent different aspects of brain activity: in the frequency domain, they rely, for example, on the power within specific bands; in the time–frequency domain, they use dynamic representations obtained through wavelets or short-time Fourier transforms; and in the spatial domain, they rely on projections such as *Common Spatial Patterns* (CSP). The classification step is then performed using supervised methods, ranging from classical approaches such as Support-Vector Machines, (SVM), Linear Discriminant Analysis (LDA) and Random Forests (RF) to more recent deep learning (DL) methods capable of directly learning discriminative representations from raw signals.

Beyond these features, focused on activity measured at individual electrodes, an important research direction is the study of *brain functional connectivity*, which corresponds to the functional interactions between brain regions and provides a richer description of the underlying neural mechanisms. From EEG signals, this connectivity is generally inferred using statistical mutual independence measures between electrodes, such as pairwise correlation, mutual information, phase-locking value (PLV) and others. These methods result in similarity matrices representing pairwise relationships between electrodes but often require thresholding to remove connections considered not relevant.

Alternatively, Graph Signal Processing (GSP) provides a powerful mathematical framework for modeling EEG signals. In fact, in the context of interactions between brain regions measured by EEG, this approach consists in representing the electrodes as nodes and their relationships as weighted edges. The graph structure can be defined in different ways, for example from the physical distances between electrodes or from functional connectivity measures derived from the signals. In addition, features specific to each electrode, such as frequency-based or topological features, can be integrated at the node level, thus enriching the representation and enabling a more detailed analysis of brain dynamics.

# Thesis objectives

Despite these advances in EEG signal analysis techniques, the explicit use of the spatial configuration of electrodes remains relatively underexplored, even though it plays a crucial role in understanding the functional activity of the brain. The main objective of this thesis is therefore to **propose a brain connectivity analysis framework that explicitly incorporate the spatial arrangement of the electrodes for the analysis and the classification EEG signals.** More precisely, this goal can be summarized through the following research questions:

I. **Graph learning:** How can we construct a graph from EEG data that reflects the overall brain connectivity, rather than only pairwise relationships between electrodes?

II. **Spatial integration:** Once this graph is defined, how can the spatial information related to the electrode positions be incorporated into the representation?

III. **Information selection:** Which types of complementary features (frequency-based, spatial, topological, etc.) should be chosen to enrich the graph?

IV. **Use for classification:** How can this complete graph be effectively exploited within a brain state classification framework?

V. **Explainability and visualization:** Are there ways to explain the obtained results and relate the learned representations to the underlying neural mechanisms?

Moreover, as an example of a potential application, the Nantes University Hospital (CHU), in collaboration with the LS2N, has developed a BCI-driven VR system to support patients with phantom limb pain. By decoding motor imagery of the missing or affected limb from EEG signals and translating it into real-time immersive visual feedback, the system promotes brain plasticity, through guided mental training, and helps relieve pain. Our graph-based EEG representations and learning strategies could enhance such systems by improving the decoding of motor imagery and mental workload monitoring to adapt the training process and ease the progression of rehabilitation sessions.

# Organization of the manuscript

This manuscript is organised into two main parts: the first part is composed of three introductory chapters that establish the theoretical and methodological foundations of the work.

**Chapter 2** presents the general context of the study, introducing the principles of electroencephalography and brain–computer interfaces, as well as the classical signal processing and machine learning methods that have historically been used for EEG classification.

**Chapter 3** provides a detailed overview of the theoretical framework of graph signal processing and graph neural networks (GNNs), which together form the basis for the approaches explored in this thesis. It also includes a state-of-the-art review of graph-based methods applied to EEG. Particular attention is given to the graph construction techniques that we adapt and extend in our proposed contributions to address the question **I**.

**Chapter 4** describes the publicly available datasets used throughout this work, detailing their acquisition protocols, experimental paradigms, and preprocessing steps. It also contains the baseline methods against which our approaches are compared, along with their reported results, providing a clear reference framework for the evaluation of the proposed methods.

The second part of the manuscript brings together the three contribution chapters, each of them aims to address the research questions formulated earlier. More specifically, these chapters are organized as follows:

**Chapter 5** introduces a novel framework for the classification of EEG functional connectivity. This approach first tackles question **I** by learning a generalized graph Laplacian under structural and sparsity constraints, providing a more informative connectivity representation than simple pairwise Pearson correlations. To address question **II**, the learned graph is transformed into a functional graph-to-image (*FG2IMG*) representation that preserves the spatial arrangement of the electrodes, their importance within the graph, and the connectivity patterns. These images are then exploited by a Convolutional Neural Network (CNN) for classification, thereby answering question **IV**. The proposed framework demonstrates competitive performance: it surpasses classical pipelines such as Riemannian Minimum Distance to Mean (R-MDM) and approaches the accuracy of CSP on motor imagery, while clearly outperforming both CSP and R-MDM for mental workload classification. Finally, the framework contributes to question **V** by enhancing interpretability, as it enables the visualization and analysis of the model's behavior and the subject-specific functional connectivity patterns, confirming the advantages of Laplacian-based connectivity learning over traditional correlation measures.

**Chapter 6** introduces a second complete processing pipeline, built upon the graph learned in Chapter 3. Our approach tackles question **II** by modeling the graph through a physically inspired force representation: electrodes are considered as particles subject to interactive forces, computed from their spatial positions and connectivity. We extend this concept by proposing a generalized formulation of forces in both 2D and 3D settings, where parameters such as distance decay and force scaling can be tuned to optimize performance. To address question **III**, the resulting force-based node features are enriched with complementary graph-topological and signal-derived measures, forming an extended feature set. These representations are then classified using traditional classifiers including Support Vector Machines (SVM), thereby answering question **IV**. Finally, this force-based modeling contributes to question **V** by providing visually interpretable and explainable representations of the underlying brain connectivity patterns. The proposed approach achieved competitive performance on the motor imagery dataset compared to the baseline and clearly outperformed it in the case of mental workload classification.

**Chapter 7** introduces a third processing pipeline based on modeling EEG signals as graphs that jointly exploit edge- and node-level information. Edges are defined based on the functional connectivity or based on the physical distances guided by the functional connectivity measures, while nodes are enriched with spectral, spatial, and topological

features. Furthermore, this chapter compares different ways of integrating spatial information at the node level or at the edge level by proposing several strategies for encoding spatial relationships between electrodes, comparing geodesic distances on spherical and ellipsoidal surfaces with Cartesian distances projected onto a 2D azimuthal plane, and combining these distance measures with various decay functions to modulate edge weights addressing questions **II** and **III**. Once the graph is built, GNNs are leveraged to process it directly for EEG classification, thereby answering question **IV**. Finally, it is found that modeling EEG signals as graphs that integrate spatial structure with spectral and topological features significantly improves classification performance compared to traditional spatial-domain baselines and provides a richer understanding of the spatial and functional relationships underlying brain activity.

# Thesis publications

The work carried out in this thesis has been disseminated through the following contributions:

**Chapter 5**

— M. Sarkis, M. Rizkallah, S. Moussaoui, *"Functional Graph Image Representation Applied to EEG-based Mental Workload Classification,"* in *Proceedings of the 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Orlando, FL, USA, 2024.

— Poster presented at the *French Conference on Artificial Intelligence in Biomedical Imaging (IABM)*, Grenoble, France, 2024.

**Chapter 6**

— M. Sarkis, M. Rizkallah, S. Moussaoui, *"Graph Force Learning from EEG Signals for Mental Workload Classification,"* in *IEEE Statistical Signal Processing Workshop (SSP)*, Edinburgh, Great Britain, 2025.

— M. Sarkis, M. Rizkallah, S. Moussaoui, *"Force-Driven Graph Learning and Classification,"* submitted to *IEEE Signal Processing Letters (SPL)*, July 2025, under review.

**Chapter 7**

— M. Sarkis, M. Rizkallah, S. Moussaoui, *"A Graph Signal Processing Framework Based on Graph Learning and Graph Neural Networks for Mental Workload Classification from EEG Signals,"* in *7th Graph Signal Processing Workshop (GSP)*, Delft, Netherlands, 2024.

— M. Sarkis, M. Rizkallah, S. Moussaoui, *« Réseaux de neurones sur graphes pour l'analyse de signaux EEG »,* in *30e Colloque sur le Traitement du Signal et des Images (GRETSI)*, Strasbourg, France, 2025.

— Poster presented at the *French Conference on Artificial Intelligence in Biomedical Imaging (IABM)*, Nice, France, 2025.

# Chapter 2

# EEG Signals Acquisition and Analysis for BCI Applications


2.1 Brain activity . . . 10
2.2 Electroencephalography . . . 12
2.2.1 Electrode placement . . . 12
2.2.2 EEG rythms . . . 12
2.2.3 EEG and BCI Applications . . . 14
2.3 Brain computer interface . . . 15
2.4 EEG classification framework . . . 15
2.4.1 Preprocessing . . . 16
2.4.2 Feature extraction . . . 17
2.4.2.1 Time domain features . . . 17
2.4.2.2 Frequency domain features . . . 17
2.4.2.3 Time-frequency domain features . . . 18
2.4.2.4 Spatial domain features . . . 18
2.4.3 Classification . . . 21
2.4.3.1 Conventional machine learning classifiers . . . 21
2.4.3.2 Deep learning approaches . . . 22
2.5 Conclusion . . . 23

This chapter establishes the scientific context of the thesis, tracing the path from the fundamental mechanisms of neuronal activity to the classification of mental states from EEG recordings. It begins with an overview of the organization of the brain and the generation of neural activity, followed by a description of how this activity can be non-invasively captured using EEG. The advantages of EEG are discussed, along with the nature and information contained in EEG recordings. Particular attention is given to their exploitation in the framework of BCIs, where neural activity is translated into actionable outputs for various applications. Finally, we review state-of-the-art approaches to EEG signal processing in BCIs, outlining the main stages of a typical pipeline, signal preprocessing, feature extraction, and classification.

## 2.1 Brain activity

The brain is a highly complex organ that governs all aspects of bodily function, including thought, memory, motor coordination, vision, respiration, temperature regulation, hunger and others. It comprises a network of blood vessels and nerves, including neurons and various supporting cells [Sultana et al. 2024]. Functioning as the command center of the body, the brain transmits and receives chemical and electrical signals. These signals regulate several physiological processes. Some may induce sensations such as fatigue or pain, while others influence movement or perception. Some remain within the brain, while others are transmitted through the spinal cord and dispersed across the body's extensive network of nerves to reach distant organs and extremities.

At a high level, the brain can be divided into the cerebrum, brainstem, a cerebellum, each being responsible for a task. (a) the cerebellum for example means the 'little brain', it contributes in the control of movements and posture. It also plays a role in motor learning and some cognitive functions (b) the brainstem controls everything that we do automatically without having to think about, like heartbeat, sense of equilibrium and breathing. Furthermore, (c) the cerebrum is the largest part of the brain that initiates and coordinates movements, it contains white and gray matter. A cerebrum is divided into four lobes (occipital, temporal, parietal and frontal) as shown in [Fig. 2.1], each has right and left counterparts: The right hemisphere is linked to imagination and creativity, whereas the left hemisphere is linked to logical abilities.

— **Frontal lobe.** is the largest lobe of the human brain and is located at the front of the head. It plays a central role in a wide range of higher cognitive functions, including decision-making, long-term planning, interpretation, idea organization, and problem-solving. It is also essential for aspects of personality, behavior, and voluntary movement.

— **Parietal lobe.** located in the middle part of the brain, is involved in object recog-

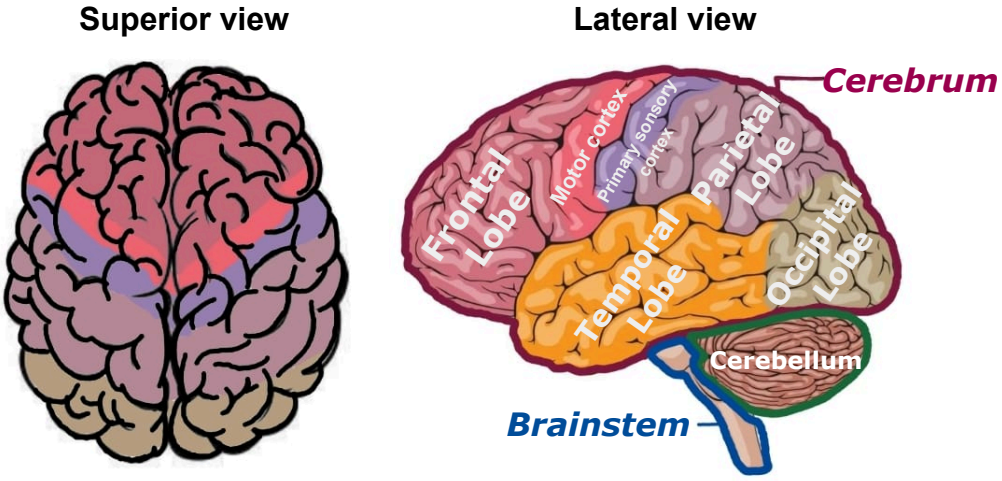



Figure 2.1 – Overview of the Brain different lobes.

nition and understanding spatial relationships, helping individuals perceive the position of their body in relation to surrounding objects. It also plays a key role in processing sensory information such as touch, pressure, and pain.

— **Occipital lobe.** is the back part of the brain that is involved in vision.

— **Temporal lobe.** The sides of the brain, temporal lobes are involved in short-term memory, speech, musical rhythm and some degree of smell recognition.

While each lobe of the brain has its own specialized functions, all of them rely on the same underlying communication system. The brain doesn't work as separate parts in isolation, rather it is billions of neurons constantly sending signals to each other. These tiny electrical and chemical messages are what allow the lobes to carry out their roles, from processing sights and sounds to controlling movement and decision-making [Britton et al. 2016]. In the nervous system, neurons communicate through synapses, which are specialized structures that allow the transfer of electrical or chemical signals to neighboring neurons. Each neuron maintains a voltage gradient across its membrane, and when incoming signals raise the membrane potential beyond a specific threshold, an action potential is triggered. This pulse travels along the neuron's axon and is transmitted across the synapse to the next neuron [Sporns 2002]. A postsynaptic potential (PSP) is a small change in a neuron's membrane voltage that occurs when it receives input from another neuron. These changes can be excitatory or inhibitory, and their combined effect determines whether the neuron will fire an action potential. In contrast to action potentials, PSPs are graded and accumulate over time and space within the neuron.

From this arises the definition of brain activity as the transmission of electrical impulses, or action potentials, between neurons and the resulting postsynaptic responses. As

neurons communicate and synchronize with each other, these action potentials accumulate to form signals that can be detected and measured [Müller-Putz 2020].

## 2.2 Electroencephalography

Electroencephalography has been used for nearly a century as a non-invasive method to measure the brain's electrical activity. It was first introduced by Hans Berger, a German neurologist and psychiatrist who recorded the brain's electrical signals from the scalp [İnce et al. 2021]. EEG is known by its high temporal resolution, but suffers from low spatial resolution since it is limited by the space covered by the electrodes. Electrodes are placed on the scalp and connected to an amplifier that sends the numerical signal to a computer. For each electrode, the voltage potential difference with another electrode (named reference) is recorded. Usually the same reference is used for the system, allowing for the detection of brain potentials generated by neuronal activity. However, due to signal attenuation through the skull and surrounding tissues, EEG can only reliably capture synchronous activity, that is, electrical signals that arise when large groups of neurons fire together.

### 2.2.1 Electrode placement

In EEG recordings, electrode placement follows standardized international systems to ensure consistency and reproducibility. One of the most widely adopted methods is the 10–20 system, which ensures equal inter-electrode spacing [Fig. 2.2]. The name "10–20" refers to the fixed distances 10% or 20% of the total length used to place electrodes along the anterior–posterior axis (from the nasion, located at the bridge of the nose, to the inion, the prominent bump at the back of the head), as well as between the left and right auricular points.

Each electrode label begins with a letter indicating the underlying cortical region: F (frontal), C (central), P (parietal), T (temporal), and O (occipital). Electrodes placed between these primary regions are labeled with two letters, such as FC for the frontal-central area. The letters are followed by either a number or the letter "z": odd and even numbers denote electrodes on the left and right hemisphere, respectively. The midline electrodes aligned along the nasion–inion axis are marked with a "z" (e.g., Cz indicates the central midline site). Furthermore, the numerical labels increase progressively with distance from the midline [Britton et al. 2016].

### 2.2.2 EEG rythms

When large groups of neurons fire in coordinated patterns, their collective activity produces rhythmic oscillations. These oscillations, captured by EEG, appear as distinct

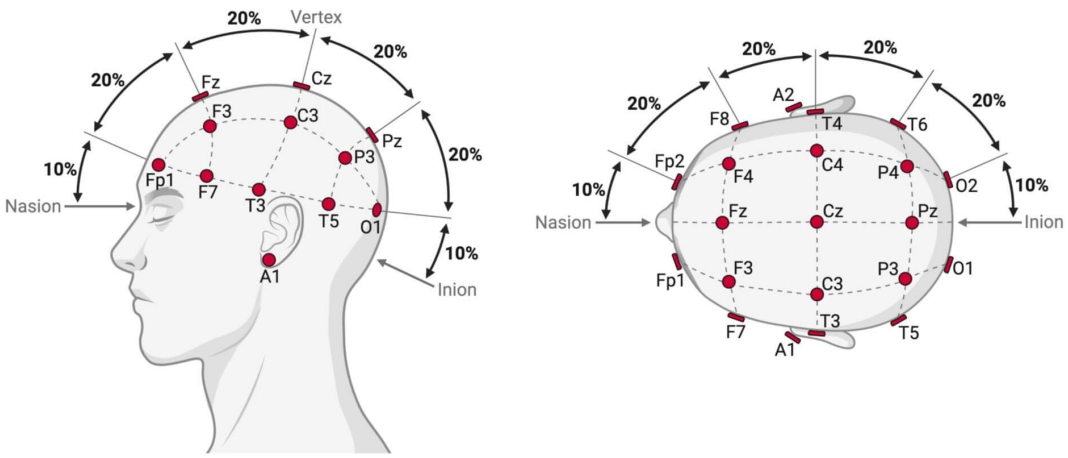



Figure 2.2 – The 10-20 system with front-back (nasion to inion) 10% and 20% electrode distances [DocCheck 2023]

brain rhythms or frequency bands, each associated with different cognitive states and levels of arousal. The relative importance level of each rhythm varies with both the individual's age and behavioral state. The principal EEG rhythms include [Blinowska et al. 2006]:

— $\delta^{EEG}$ (0.5–4 Hz): this rhythm predominates during deep sleep. Delta activity tends to be synchronous and widespread across the scalp.

— $\theta^{EEG}$ (4–8 Hz): Theta activity can appear during certain emotional or cognitive states, and is typically dominant during light sleep, deep relaxation, meditation, and dreaming.

— $\alpha^{EEG}$ (8–13 Hz): Alpha rhythms are most prominent during wakeful rest, especially when the eyes are closed and the individual is relaxed. They are strongest in the posterior regions of the head and are suppressed or attenuated by visual attention or mental effort.

— $\beta^{EEG}$ (13–30 Hz): Beta activity is associated with increased alertness, focused attention, concentration, anxiety, mental activity, and problem-solving. It typically reflects active cognitive engagement and decision-making processes.

— $\gamma^{EEG}$ (above 30 Hz): Gamma rhythms are involved in information processing, such as sensory perception, and are also linked to the initiation of voluntary movements.

Analyzing EEG rhythms offers valuable insights into the brain's functional organization. In particular, in this thesis, we will use the relative energy of each frequency band to characterize the signal recorded at each electrode in Chapters 6 and 7.

### 2.2.3 EEG and BCI Applications

EEG serves as a cornerstone in clinical practice, primarily used for diagnosing and confirming neurological conditions such as neonatal seizures, encephalopathy, epilepsy, and coma [Britton et al. 2016]. Its applications extend beyond diagnostics into intraoperative monitoring, where it helps the assessment of the depth of anesthesia during surgical interventions. More broadly, EEG signals provide a foundation not only for the identification of neurological disorders but also for informing treatment approaches. In particular, BCIs, described in detail in the following section, harness EEG data to transform neural activity into actionable commands, enabling therapeutic and assistive applications in real time.

One of the most prominent application areas of BCIs is motor rehabilitation and assistance for individuals with physical impairments but preserved cognitive function. Originally developed to support patients with motor disabilities, BCIs are now applied in rehabilitation after amputations, strokes, or nervous system injuries [Y.-N. Ma et al. 2025]. They enable the control of neuroprosthetic devices that artificially trigger muscle contractions, restoring functions such as hand grasping for quadriplegic patients [Pfurtscheller et al. 2008]. Noninvasive EEG-based systems, such as electrically actuated hand orthoses, have demonstrated promising outcomes in supporting impaired motor functions [Devi et al. 2025]. Beyond medicine, BCIs are increasingly applied in non-clinical domains like gaming and entertainment [Van Erp et al. 2012], where EEG signals allow users to interact with virtual environments, for example, controlling elements in a BCI-adapted "tower defense" game through steady-state visually evoked potentials (SSVEPs) [Van Vliet et al. 2012].

Another significant domain of BCI application lies in pain management, particularly for neuropathic conditions like phantom limb pain, which remains difficult to treat and affects around 7% of the European population [Kooijman et al. 2000]. Traditional approaches such as mirror therapy [Ramachandran et al. 1996] and mental motor imagery have shown encouraging results, though their effectiveness is limited by the subjective variability of patients' imagery ability and motivation. To address these challenges, BCI systems provide real-time neurofeedback by decoding EEG signals and projecting them into virtual environments, thereby reinforcing mental imagery and supporting rehabilitation [Ortiz-Catalan et al. 2016].

The work presented in this thesis falls within the field of BCIs, which can be applied to neurorehabilitation, for example, to help relieve patients suffering from phantom limb pain. In this context, we focus on the classification of two types of mental tasks: mental workload and motor imagery. Although these paradigms serve as the main use cases in this work, the proposed methods are generic and can be applied to other types of mental task classification. An overview of a typical BCI processing pipeline, from the acquisition

of EEG signals to the generation of feedback, which depends on the specific application is shown in [Fig. 2.3]. In our case, the contribution of this thesis lies primarily in the classification stage, where EEG signals are processed and decoded to distinguish between different mental states.

## 2.3 Brain computer interface

Formally, a Brain–Computer Interface is a system composed of three components: recording brain activity, decoding this activity to transform it into commands, and providing feedback to the user. The concept was first proposed by Jacques Vidal, a Belgian scientist, in 1973 [Vidal 1973]. A BCI system typically operates in two stages [Clerc et al. 2016]: (a) Open-loop stage: this is an offline calibration phase during which system parameters are defined and the models are trained. (b) Closed-loop stage: this is the online phase, where brain activity is recorded, decoded into actionable commands, and fed back to the user in real time.

The closed-loop stage encompasses the full process from brain activity recording to user feedback. Electrical signals can be measured using techniques such as electroencephalography (EEG) as detailed in [Sec. 2.1]). A signal processing pipeline is applied on the recorded signals, beginning with preprocessing to remove artifacts (e.g., cardiac activity, muscle movements, electromagnetic interference). This is followed by feature extraction and classification, the primary focus of this thesis, where relevant signal features are identified and categorized. Although not addressed here, the command translation step converts classification outputs into actionable commands for patient use. The final output is delivered according to the application, including computer-based feedback, prosthetic control, or interaction within virtual environments.

In this following, we present the overall signal processing pipeline adopted for EEG classification. The discussion begins with an examination of the principal stages of the classification framework with a particular emphasis on the methodological approaches employed at each stage.

## 2.4 EEG classification framework

A conventional machine learning framework for electroencephalography (EEG) signal classification is presented in [Fig. 2.3]. This framework is composed of three principal stages: pre-processing and filtering, feature extraction and classification. Specifically, through the processes of pre-processing and feature extraction, raw EEG signals are transformed into structured and informative representations that are suitable for subsequent decision-making tasks. The latter are subsequently provided as input to a classification

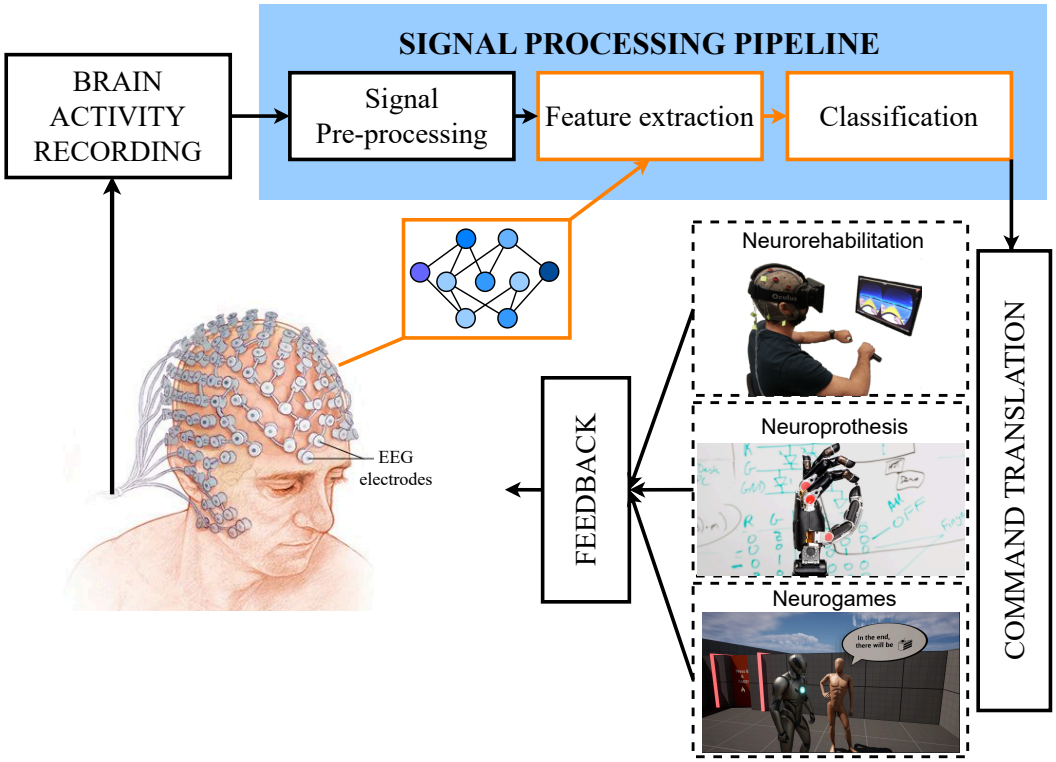



Figure 2.3 – Overview of a brain–computer interface system. Brain activity is recorded through electroencephalography and processed through a signal-processing pipeline that includes preprocessing, feature extraction, and classification. The resulting outputs are translated into feedback or control commands depending on the application, such as neurorehabilitation, neuroprosthesis control, or neurogaming.

model. Each component of this framework will be elaborated upon in the following sections.

### 2.4.1 Preprocessing

Because EEG signals are often contaminated by various physiological (i.e., originating from the body) and non-physiological (i.e., caused by external factors) artifacts—such as cardiac activity, eye movements and blinks, muscle activity, and head or cable movements, the preprocessing stage is crucial for artifact removal and signal denoising [X. Jiang et al. 2019]. Environmental artifacts can often be eliminated through simple filtering, as their frequency content is typically inconsistent with that of the EEG signal. To this end, EEG signals are commonly filtered in both the temporal domain (using band-pass filters) and the spatial domain (using spatial filters). For instance, notch filters are applied to suppress interference from electrical power lines [Plechawska-Wójcik et al. 2019], high-pass filters are employed to remove low-frequency drifts and other non-stationary components, and low-pass filters are used to attenuate high-frequency noise. To reduce eye-blink artifacts, spatial filtering techniques such as Independent Component Analysis (ICA) are

frequently applied. Moreover, blind source separation (BSS) methods [Fitzgibbon et al. 2007], particularly Principal Component Analysis (PCA) [Turnip et al. 2014], have proven effective in removing muscle-related artifacts. In typical EEG acquisition, one electrode is often dedicated to recording cardiac activity; such artifacts can therefore be more easily removed using a reference waveform. Although a comprehensive examination of preprocessing techniques falls outside the scope of this thesis, it is important to acknowledge their critical role in establishing a robust foundation for reliable EEG analysis.

### 2.4.2 Feature extraction

Once clean EEG signals have been obtained, they are transformed into informative feature representations that are suitable for subsequent decision-making tasks. These features may originate from different domains, and multiple types of features can be combined to enhance the representation of the signal, as detailed below.

#### 2.4.2.1 Time domain features

Time-domain feature extraction is one of the most fundamental techniques, in which EEG signals are analyzed directly as a function of time. This approach enables the quantification of how a signal evolves over time. Among the methods in this category is the autoregressive (AR) modeling, where the signal is represented by AR coefficients that form the feature vector. For example, [Duhan et al. 2020] investigated EEG-based biometric identification, aiming to recognize individuals from their EEG signals. In their study, an autoregressive model of order three was employed to extract time-domain features, which were subsequently evaluated using multiple classifiers. Time-domain features have also been integrated into deep learning frameworks to capture temporal dependencies in EEG data. In this context, [A. Wu et al. 2024] proposed an autoregressive pre-training approach, demonstrating that such pre-training enables models to more accurately capture the temporal characteristics of EEG signals, thereby improving classification performance. Other time-domain feature extraction methods are simpler and rely on statistical measures of the signal, such as variance, first- and second-order derivatives, zero-crossing counts for the derivatives, and the number of local minima and maxima [Abdulhussien et al. 2022]. These statistical features have been applied, for instance, in seizure detection, where they were combined with frequency-domain features obtained via a sliding-window Fourier transform.

#### 2.4.2.2 Frequency domain features

Extending beyond time-domain approaches, frequency-domain analysis techniques focus on decomposing the EEG signal into its constituent sinusoidal components, thereby enabling the extraction of features that characterize the underlying rhythmic activity of

the brain. The process typically begins with transforming the signal from the time domain to the frequency domain, most commonly via the Fast Fourier Transform (FFT). Such analysis is crucial for identifying the characteristic frequency bands of brain oscillations, which are known to correspond to different mental states, as discussed in [Section 2.2.2]. Examining the EEG spectrum enables the quantification of the signal's energy distribution across these bands within a specified time window. In this context, the Power Spectral Density (PSD) and its normalized form are fundamental tools, allowing the derivation of secondary features such as the intensity-weighted mean frequency (IWMF), intensity-weighted bandwidth (IWBW), and spectral edge frequency (SEF) [Boonyakitanont et al. 2020]. For example, the authors of [Tran et al. 2023] extracted PSD features from the theta, alpha, and beta bands to classify two emotional states: happy and sad, demonstrating a clear correlation between PSD values and emotional states. Their results also indicated that all EEG sub-bands contribute to emotion discrimination at varying levels. Similarly, [Alam et al. 2021] employed PSD features from the alpha and beta bands to classify two motor imagery tasks in a BCI setting. In their approach, the FFT was first applied to obtain the spectral representation of the EEG signals, followed by the Welch method [Solomon Jr 1991] to estimate the PSD.

#### 2.4.2.3 Time-frequency domain features

While time-domain features provide insights into the temporal characteristics of EEG signals, and frequency-domain features capture their spectral composition, both approaches are limited when dealing with the inherently non-stationary nature of EEG activity. To overcome these limitations, time–frequency methods have been developed, allowing the simultaneous examination of signal dynamics across both time and frequency. This enables a more comprehensive characterization of transient neural oscillations and event-related changes. This can be effectively achieved using the wavelet transform. In the literature, both continuous [Puranik et al. 2018] and discrete wavelet transforms ([Aliyu et al. 2023],[Kumar et al. 2017]) have been studied. They are frequently applied in the context of epilepsy research, where the goal is to characterize both the timing and frequency content of epileptic spikes. Among the various wavelet families explored, Daubechies-3 wavelets [Kumar et al. 2017] and Daubechies-2 wavelets [Aliyu et al. 2023] have demonstrated a high efficiency in detecting various EEG signal variations. Similarly, [Gosala et al. 2023] applied the wavelet scattering transform for schizophrenia classification, comparing the continuous and discrete wavelet transforms alongside various classifiers.

#### 2.4.2.4 Spatial domain features

In EEG analysis, while the time and frequency domains capture how brain activity evolves over time and oscillatory components, the spatial domain focuses on how this ac-

tivity is distributed across electrode locations on the scalp, highlighting variations across space. To characterize these spatial patterns, various approaches can be employed. In this section, we focus on two state-of-the-art methods: the first is based on spatial filtering, specifically the Common Spatial Pattern technique, while the second exploits the riemannian geometry of the covariance matrices manifold.

**Common Spatial Pattern (CSP)** A spatial filter is a mathematical method that produces new values by taking weighted combinations of signals from other electrodes. Among these methods, the CSP algorithm, a supervised spatial filter, is particularly popular. It was originally introduced by [Koles 1991] to compute spatial filters that transform multichannel time-series data from two classes into a new space, such that the variance for one class is maximized and for the other minimized. After computing the two covariance matrices corresponding to the two classes, the simultaneous diagonalization of those two matrices, also called generalized eigenvalue problem is realized. The resulting eigenvectors form the columns of the CSP projection matrix $\mathbf{W}_{CSP}$. When the EEG signals are projected using a set of the first and last columns of $\mathbf{W}_{CSP}$, the variance of the transformed signals is maximized for one class and minimized for the other, providing features that are highly discriminative for classification. A more comprehensive overview of the original CSP algorithm and the generalized eigenvalue problem can be found in [Appendix A].

Many variants of the original CSP algorithm have been proposed to enhance subject-specific performance. Filter Bank Common Spatial Pattern (FBCSP) [Ang et al. 2012] improves CSP by automatically selecting discriminative frequency bands for EEG band-pass filtering. It includes four stages starting with band-pass filtering to decompose EEG into multiple frequency bands, spatial filtering using CSP to extract band-specific features, then feature selection, and classification. Temporally Constrained Sparse Group Spatial Patterns (TCSGSP) [Y. Zhang et al. 2018] extends CSP to jointly optimize features across frequency bands and time, capturing temporal dynamics. CSP+ [Barachant et al. 2010b] enhances CSP by interpreting it in Riemannian geometry, adding Riemannian mean computation [eq.A.1] to improve spatial feature representation.

CSP is inherently binary. When dealing with more than two classes, multiclass CSP [Grosse-Wentrup et al. 2008] uses joint approximate diagonalization (JAD) to extract components maximizing mutual information with class labels. An alternative, which we adopt in this work, is the one-versus-rest (OVR) strategy [W. Wu et al. 2006]. For each class $\mathcal{Y}_q,\ q \in \mathbb{N}\}$, we group all trials from class $\mathcal{Y}_q$ as the *positive* class and all trials from the remaining classes as the *negative* class. A CSP projection matrix $\mathbf{W}_{CSP}^{(c)}$ is computed for each binary partition $(c, \text{rest})$. The resulting features from all $\mathbf{W}_{CSP}^{(c)}$ are concatenated to form the final feature vector.

**Covariances and Riemannian geometry** As described earlier, the CSP algorithm relies on the covariance matrices of different classes to derive optimal spatial filters. An alternative approach is to directly exploit the distances between covariance matrices for classification. Given two symmetric positive definite (SPD) matrices corresponding to the covariances of two classes, $\mathbf{S}_1$ and $\mathbf{S}_2$, their *Euclidean distance* is defined as the Frobenius norm of their difference:

$$\delta_E(\mathbf{S}_1, \mathbf{S}_2) = \|\mathbf{S}_1 - \mathbf{S}_2\|_F.$$

The corresponding *arithmetic mean* based on this distance is expressed as:

$$\mathfrak{A}_e(\mathbf{S}_1, \ldots, \mathbf{S}_I) = \arg \min_{\mathbf{S} \in \mathcal{SPD}(n)} \sum_{i=1}^{I} \delta_E^2(\mathbf{S}, \mathbf{S}_i) = \frac{1}{I} \sum_{i=1}^{I} \mathbf{S}_i. \tag{2.1}$$

where $\mathcal{SPD}(n)$ denotes the space of $n \times n$ SPD matrices. However, since sample covariance matrices are SPD, they do lie on a Riemannian manifold $\mathcal{M}$ rather than in a flat Euclidean space. Consequently, treating them as ordinary vectors in Euclidean space can lead to an underestimation of the true distances between them. To overcome this limitation, [Barachant et al. 2012] proposed handling SPD matrices directly in their native space by computing distances on the manifold.

A Riemannian manifold is a differentiable manifold whose tangent space at each point is a finite-dimensional Euclidean space. The shortest path between two points on such a manifold is called a geodesic, and the Riemannian distance between these points is defined as the length of this curve. In the case of SPD matrices, the affine-invariant Riemannian distance is given by:

$$\delta_R(\mathbf{S}_1, \mathbf{S}_2) = \left\| \log\left(\mathbf{S}_1^{-1}\mathbf{S}_2\right) \right\|_F = \left[ \sum_{i=1}^{k} \log^2 \lambda_i \right]^{1/2}, \tag{2.2}$$

where $\lambda_i$ for $i = \{1, \ldots, k\}$ are the real positive eigenvalues of $\mathbf{S}_1^{-1}\mathbf{S}_2$. The Riemannian mean of a set of SPD matrices $\{\mathbf{S}_1, \ldots, \mathbf{S}_I\}$ is defined as the matrix that minimizes the sum of squared Riemannian distances to all matrices in the set:

$$\mathfrak{A}_r(\mathbf{S}_1, \ldots, \mathbf{S}_I) = \arg \min_{\mathbf{S} \in \mathcal{SPD}(n)} \sum_{i=1}^{I} \delta_R^2(\mathbf{S}, \mathbf{S}_i), \tag{2.3}$$

By adopting this approach, spatial filtering is no longer required, as the spatial information is already embedded in the covariance matrix. For classification, any distance-based classifier can be applied using the Riemannian distance. When combined with the Minimum Distance to Mean (MDM) classifier, this method is considered a baseline in the remainder of this manuscript.

An alternative method, proposed by [Barachant et al. 2010a], projects the covariance matrices onto the tangent plane of the Riemannian manifold, where they can be treated

as vectors and compared using Euclidean distances.
Covariance matrices have been widely used as features in the literature. For instance, [Fernandez et al. 2024] integrated covariance matrices with case-based reasoning to enhance the explainability of model decisions, allowing the identification of relationships within the data that contribute to classification. Their approach was applied to Alzheimer's and Parkinson's disease diagnosis. Similarly, [Gopan K et al. 2022] proposed a classification framework based on covariance matrices, comparing a strategy that used matrix properties of the covariance with one based on Riemannian geometry via tangent-plane projection, for the classification of eyes-open versus eyes-closed EEG signals. Finally, [Carrara et al. 2024] introduced an augmented covariance matrix method that incorporates both spatial and temporal information. They performed classification using Riemannian geometry and demonstrated that their approach outperformed the state-of-the-art in motor imagery tasks.

### 2.4.3 Classification

Following feature extraction, EEG classification can be approached using either traditional machine learning methods or more recent deep learning architectures.

#### 2.4.3.1 Conventional machine learning classifiers

Traditional machine learning pipelines rely on handcrafted features derived from the raw EEG signals, such as band-power, time–frequency representations, spatial filters (e.g., CSP), or covariance matrices [Lotte, Congedo, et al. 2007]. Linear classifiers are discriminative models that separate data classes using linear decision boundaries. They are among the most widely applied algorithms in BCI research. Among them, the two most commonly employed are LDA and SVMs. LDA classifies data by projecting it onto a line or hyperplane that maximizes the separation between class means while minimizing within-class variance. It assumes normally distributed data with equal class covariances. It has a low computation requirement which makes it suitable for online BCI. The most widely adopted pipeline is CSP combined with Linear Discriminant Analysis (CSP+LDA) [S.-L. Wu et al. 2013, Alotaiby et al. 2017,Soler et al. 2024]. Support Vector Machines (SVMs) also classify data using a discriminant hyperplane. Unlike LDA, SVM selects the hyperplane that maximizes the distance to the nearest training samples, which typically improves the model's generalization ability. The use of this classifier in litterature was reported to achieve good performance for EEG classification [Guler et al. 2007, Jothiraj et al. 2025]. Random Forest classifiers are ensembles of decision trees, each trained on randomly selected samples and features. This randomness increases tree diversity, enhancing the model's robustness and generalization. RFs have been widely applied in EEG

studies [Edla et al. 2018, Kalousios et al. 2025]. A particularly successful geometry-based approach is the MDM classifier [Yger et al. 2016], which, when applied to covariance matrices in the Riemannian framework, exploits the affine-invariant geometry of symmetric positive definite matrices and remains a robust state-of-the-art baseline in many BCI paradigms. Other classifiers including K-Nearest Neighbors (KNN), which classifies a new sample by assigning it the most dominant class among its closest neighbors in the training set were also used for EEG classification [Manjusha et al. 2016]. It is also possible to aggregate different classifiers together using different strategies, for instance the voting scheme in which case the final predicted class will correspond to that of the majority [Rakotomamonjy et al. 2005].

#### 2.4.3.2 Deep learning approaches

Deep learning methods differ from traditional pipelines by integrating feature extraction and classification into a single end-to-end framework, often operating directly on raw or minimally preprocessed EEG. Architectures such as Convolutional Neural Networks and Graph Neural Networks learn hierarchical representations, capturing local temporal and spatial dynamics and progressively more abstract neural patterns. CNN-based approaches vary in architecture and input, using either raw signals or derived EEG features; a more detailed review of these methods is presented in Chapter 5. Although deep learning has transformed fields such as computer vision and speech recognition, systematic reviews ([Lotte, Bougrain, et al. 2018]) report that its superiority over traditional pipelines (e.g., CSP+LDA, Riemannian MDM) in EEG–BCI classification is not yet consistently demonstrated. Recent advances have shown promise, but deep learning still faces challenges such as limited data availability, high variability across subjects, and the absence of a consensus baseline [X. Wang et al. 2024]. Nevertheless, research continues to explore its potential for scalable and fully automated EEG decoding. In parallel, a growing body of work investigates graph-based neural networks, which model functional connectivity and exploit the complex spatial relationships of EEG sensors. These GNNs offer a principled way to integrate connectivity information into learning [Klepl, M. Wu, et al. 2024]; an in-depth review of such approaches is provided in Chapter 7.

The contributions, subsequently detailed in this manuscript, present a series of EEG classification pipelines built on diverse feature extraction strategies, spanning traditional machine learning classifiers to modern deep learning models. Particular emphasis is placed on convolutional neural networks and graph neural networks, whose performance is systematically compared against established machine learning baselines.

## 2.5 Conclusion

In this chapter, we introduced the fundamental concepts of brain activity and electroencephalography, explaining how EEG measures neural dynamics and describing the different frequency bands that characterize brain signals. We then detailed the standard EEG signal processing pipeline employed in brain–computer interface systems, covering the main stages of preprocessing, feature extraction and classification. A review of the different methods used at each stage was provided, together with an overview of representative BCI applications.

While much of the literature has focused on analyzing brain activity through local signal characteristics, another complementary line of research investigates brain connectivity, which captures the functional interactions and dependencies between different brain regions. Modeling this connectivity provides a richer and more structured view of neural dynamics than treating electrodes as independent channels.

Capturing the underlying structure of brain connectivity from EEG naturally leads to representing it as a graph, where nodes correspond to electrodes or brain regions and edges encode spatial or functional relationships. In the next chapter, we will formally introduce the concept of a graph and present methods to define and learn its structure directly from EEG data, establishing the foundation for the graph-based approaches explored in the remainder of this work.

# Chapter 3

# Foundations of Graph-based modeling for EEG Analysis


3.1 Graph theory and concepts . . . 26
3.1.1 Definitions . . . 26
3.1.2 Graph spectral theory . . . 30
3.1.3 Graph topology construction . . . 32
3.2 Graph neural networks . . . 34
3.2.1 Aggregation and message passing . . . 34
3.3 Graph representations for EEG . . . 35
3.3.1 Handcrafted graphs . . . 36
3.3.2 Learned graphs . . . 37
3.3.3 Graph learning under Laplacian and structural constraints . . . 40
3.4 Conclusion . . . 42

In this chapter, we review the main principles of Graph Signal Processing. We first explore the basics of graph structures and signals lying on irregular domains. We further present the different techniques of defining the underlying graph structure, i.e. graph learning, used in the literature. With the aid of the aforementioned notions, we then review the graph neural networks strategies tailored for inference problems based on graphs. This chapter is inspired from different references in the domain: [Stankovic et al. 2019], [Dong et al. 2019] and [Shuman et al. 2013].

## 3.1 Graph theory and concepts

It all started in the 18th century, when Leonhard Euler addressed the famous problem of the Seven Bridges of Königsberg. The city of Königsberg, divided by the Pregel River, featured seven bridges connecting four distinct land areas. The central question posed by its citizens was whether it was possible to walk through the city in such a way that each bridge was crossed exactly once. Euler abstracted the problem by representing land areas as nodes and bridges as edges. He demonstrated that no such path exists, since more than two nodes had an odd number of connecting edges, and that if a graph is connected and has no odd number of edges for all nodes, it has at least one path. This pioneering work, *"Solutio problematis ad geometriam situs pertinentis."* published in 1736 [Euler 1736], laid the foundation for graph theory. Eversince, graphs started being used as a powerful tool to represent complex networks such as social networks, biological networks, patent networks, traffic networks, citation networks, communication networks and others.

### 3.1.1 Definitions

Graphs are powerful mathematical tools that are widely used to represent relationships between entities. A graph, denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consists of two fundamental components: a set of $n$ vertices (or nodes) denoted by $\mathcal{V} = \{1, 2, \ldots, n\}$, and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, which represent the relationships or connections between pairs of vertices. One of the most common ways to represent a graph is through its adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$. In this matrix, the entry $\mathbf{A}_{uv}$ indicates whether there is an edge between nodes $u$ and $v$:

$$\mathbf{A}_{uv} = \begin{cases} 1, & \text{if } (u, v) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

In the case of an undirected graph, the edge relation is symmetric; if node $u$ is connected to node $v$, then node $v$ is also connected to node $u$, consequently, the adjacency matrix is symmetric (see [Fig. 3.1.a]). In contrast, a directed graph consists of oriented edges, where an edge from node $u$ to node $v$ does not necessarily imply an edge from $v$

to $u$. In this case, the adjacency matrix $\mathbf{A}$ is generally asymmetric, which means that $\mathbf{A}_{uv} \neq \mathbf{A}_{vu}$ in general. Directed graphs are used to model asymmetric relationships, such as citations between papers, follower networks, or signal flows [Jannoud et al. 2014]. This type of structure is illustrated in [Fig. 3.1.b]. Graphs can also encode additional information through weights on edges, representing the strength of connections. These are modeled using a weight matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, where $\mathbf{W}_{uv}$ denotes the weight of the edge from node $u$ to node $v$ (see [Fig. 3.1.c]). Moreover, a node may be connected to itself via an edge known as a self-loop. Self-loops appear as non-zero entries on the diagonal of the adjacency or weight matrix. For example, in [Fig. 3.1], node 1 has a self-loop, which is reflected in the corresponding diagonal entry.

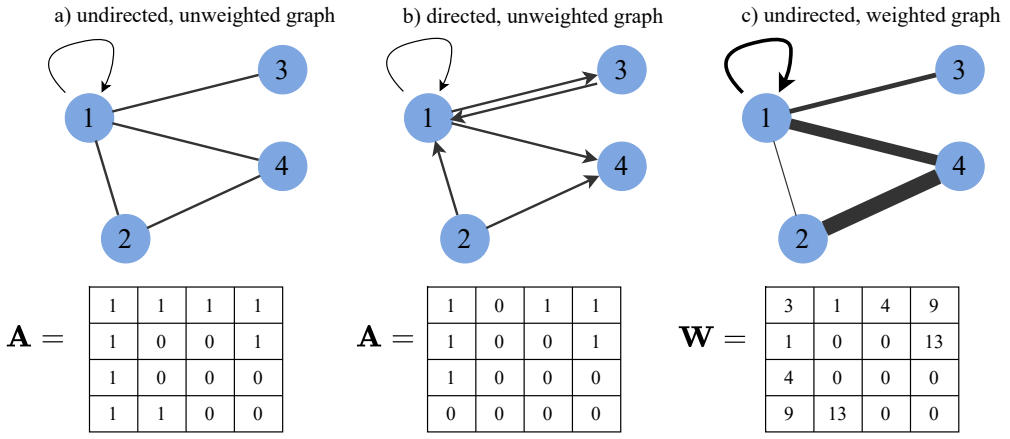



Figure 3.1 – Examples of different graph structures with 4 vertices with a self loop on vertex 1: (a) unweighted undirected, (b) directed, and (c) weighted with self-loops, along with their corresponding adjacency and weight matrices.

**Neighbourhood.** The neighborhood of a vertex describes the set of nodes that can be reached from it within a certain number of steps (or hops). Two vertices are said to be 1-hop neighbours if they are directly connected by an edge. That is, if $(u, v) \in \mathcal{E}$, then node $v$ belongs to the 1-hop neighbourhood of node $u$, denoted by $\mathcal{N}_1(u)$. This concept extends naturally to define $k$-hop neighbourhoods, which include all nodes that can be reached from a given node within $k$ steps along the edges of the graph. To formalize this, let $d(u, v)$ represent the shortest path distance (measured in number of hops) between nodes $u$ and $v$. Then, the $k$-hop neighborhood of node $u$ is defined as:

$$\mathcal{N}_k(u) = \{v \in \mathcal{V} \mid d(u, v) \leq k\} \tag{3.2}$$

In a connected graph, every pair of nodes is reachable by some finite path, and hence, for each node, its $k$-hop neighborhood grows with $k$ until it eventually includes the entire

graph.

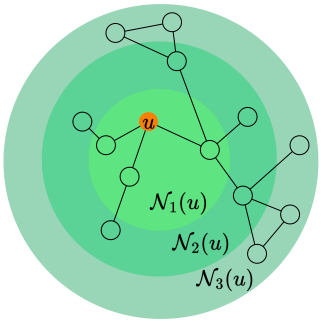



Figure 3.2 – An example of *3*-hop neighborhood of a vertex $u$

**Degree.** In an unweighted graph, the degree of a node $u$, is another important structural property. Denoted as $\mathbf{deg}_u$, it corresponds to the number of *1*-hop neighbors it has, that is, the number of edges incident to it. In contrast, for a weighted graph, we define the weighted degree of node $u$ as the sum of the weights of all edges connected to it. This is equivalent to summing the entries in the $u^{\text{th}}$ row (or column, for undirected graphs) of the weight matrix $\mathbf{W}$. The degree matrix $\mathbf{D}$ is a diagonal matrix that stores these degrees on its diagonal. Formally, the diagonal entry corresponding to node $u$ is given by:

$$\mathbf{D}_{uu} = \sum_{v \in \mathcal{N}_1(u)} \mathbf{W}_{uv}. \tag{3.3}$$

Let's take the graph in [Fig. 3.3], as an example, in this case:

$$\mathbf{D} = \begin{bmatrix} 1.21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.81 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.59 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.82 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.66 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.64 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.61 \end{bmatrix}; \quad \mathbf{deg} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

.

**Closeness centrality.** It quantifies how close a vertex is to all other vertices in the graph. It reflects the average distance required for information to spread from a given node to all others. The closeness centrality of a vertex $u$, denoted $CC_u$, is defined as the inverse of the average shortest path distances from $u$ to all other nodes [Freeman 1978]:

$$CC_u = \frac{N-1}{\sum_{v=1}^{N-1} d(v,u)}, \tag{3.4}$$

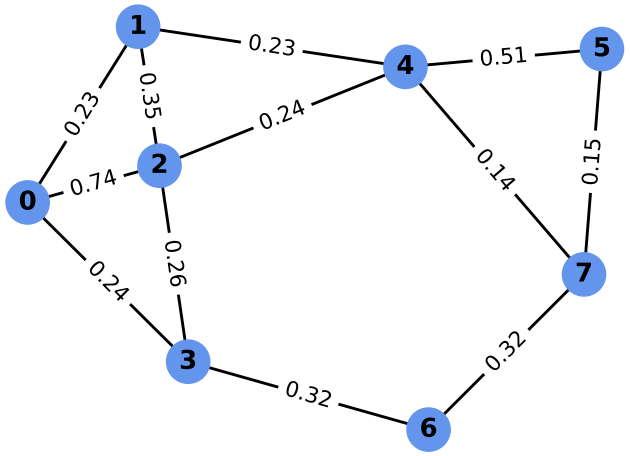



Figure 3.3 – A graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$; with a set of vertices $\mathcal{V} = \{0, 1, 2, 3, 4, 5, 6, 7\}$

where $N - 1$ is the number of nodes reachable from $u$. A higher closeness centrality indicates that the node is, on average, nearer to the rest of the graph and can potentially disseminate information more efficiently.

**Betweenness centrality** measures the extent to which a vertex lies on the shortest paths between other pairs of vertices. A node with high betweenness centrality acts as a bridge or bottleneck for the flow of information through the graph. The betweenness centrality of a vertex $v$ is defined as [Brandes and Pich 2007]:

$$CB_v = \sum_{s,t \in \mathcal{V}} \sum_{s \neq v \neq t} \frac{\#d(s,t)_v}{\#d(s,t)}. \tag{3.5}$$

Here, $\#d(s,t)$ denotes the total number of shortest paths between vertices $s$ and $t$, and $\#d(s,t)_v$ is the number of those paths that pass through node $v$. The more frequently a node appears on shortest paths, the higher its betweenness centrality.

**The clustering coefficient** of a graph quantifies the tendency of a node's neighbors to create triangles in the graph. For a given vertex $u$, the clustering coefficient measures the likelihood that two of its neighbors are also connected to each other. In the case of a weighted graph, the clustering coefficient incorporates edge weights to reflect the strength of connections. It can be expressed as follows [Onnela et al. 2005]:

$$C_u = \frac{1}{\mathbf{deg}_{uu}\big(\,\mathbf{deg}_{uu} - 1\big)} \sum_{vw} \Big(\hat{\mathbf{W}}_{uv}\, \hat{\mathbf{W}}_{uw}\, \hat{\mathbf{W}}_{vw}\Big)^{1/3}. \tag{3.6}$$

where $\hat{\mathbf{W}}_{uv} = \frac{\mathbf{W}_{uv}}{\max(\mathbf{W})}$ is the normalized weight by the maximum weight in the graph. A higher value of $C_u$ indicates that the neighbors of node $u$ are more densely interconnected, implying a higher tendency toward local clustering in the graph.

**Eigenvector centrality** is a measure of node importance in a network that assigns relative scores to all nodes based on the principle that connections to high-scoring nodes contribute more to the score of a node than connections to low-scoring ones. Mathematically, a vertex $u$ has a high eigenvector centrality if it is connected to many other nodes that themselves have high centrality scores. This idea is captured mathematically by defining the centrality score of a node $u$ as the $u^{th}$ entry of the eigenvector $\mathbf{u}$ corresponding to the largest positive eigenvalue $\lambda$ (in absolute value) of the adjacency matrix $\mathbf{A}$. The latter satisfies the following eigenvalue equation:

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}. \tag{3.7}$$

In component-wise form, the centrality of each node $u$ satisfies the following property:

$$\mathbf{u}_u = \frac{1}{\lambda}\sum_{v\in\mathcal{V}} \mathbf{A}_{uv}\mathbf{u}_v, \tag{3.8}$$

showing that the centrality of a node depends recursively on the centralities of the neighboring nodes.

The topological features of the nodes in chapters 6 and 7 of this manuscript correspond to a subset of the graph metrics described in this section. [Tab. 3.1] highlights the numerical values of those metrics in the simplified case of [Fig. 3.3]. In this graph, node 2 has the highest weighted degree (1.59) and the highest eigenvector centrality (0.61), indicating that it is the most strongly connected and influential node overall. Node 4 shows the greatest weighted closeness centrality (3.02) and also the highest weighted betweenness centrality (0.38), highlighting its role as a central bridge that lies on many of the shortest weighted paths. Finally, node 0 exhibits the highest weighted clustering coefficient (0.34), meaning its neighbors are the most densely interconnected compared to other nodes.

### 3.1.2 Graph spectral theory

In graph theory, the graph Laplacian matrix $\mathbf{L}$ is a fundamental way to represent a graph in matrix form, yet it serves as a bridge between digital signal processing (DSP) and graph signal processing because it generalizes key concepts from DSP, like the Fourier transform, filtering, and frequency content, to irregular domains, instead of regular grids or time series. Unweighted combinatorial graph Laplacian represents the most basic formulation where $\mathbf{L} = \mathbf{deg} - \mathbf{A}$; in the case of weighed un-directed graph, it is formulated

Table 3.1 – Graph metrics for each node of the weighted graph $\mathcal{G}$ in [Fig. 3.3].

| **Node** $u$ | $\mathbf{D}_{uu}$ | $CC_u$ | $CB_u$ | $C_u$ | $EC_u$ |
|---|---|---|---|---|---|
| 0 | 1.21 | 2.10 | 0.05 | 0.34 | 0.54 |
| 1 | 0.81 | 2.45 | 0.14 | 0.30 | 0.36 |
| 2 | 1.59 | 2.46 | 0.10 | 0.23 | 0.61 |
| 3 | 0.82 | 2.17 | 0.14 | 0.16 | 0.29 |
| 4 | 1.12 | 3.02 | 0.38 | 0.11 | 0.28 |
| 5 | 0.66 | 2.00 | 0.00 | 0.30 | 0.14 |
| 6 | 0.64 | 2.06 | 0.05 | 0.00 | 0.11 |
| 7 | 0.61 | 2.69 | 0.38 | 0.10 | 0.09 |

as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. More advanced formulations include the self-loop matrix as:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} + \mathbf{V}. \tag{3.9}$$

The Laplacian corresponding to [Fig. 3.3] would be the following:

$$\mathbf{L} = \begin{bmatrix} 1.21 & -0.23 & -0.74 & -0.24 & 0 & 0 & 0 & 0 \\ -0.23 & 0.81 & -0.35 & 0 & -0.23 & 0 & 0 & 0 \\ -0.74 & -0.35 & 1.59 & -0.26 & -0.24 & 0 & 0 & 0 \\ -0.24 & 0 & -0.26 & 0.82 & 0 & 0 & -0.32 & 0 \\ 0 & -0.23 & -0.24 & 0 & 1.12 & -0.51 & 0 & -0.14 \\ 0 & 0 & 0 & 0 & -0.51 & 0.66 & 0 & -0.15 \\ 0 & 0 & 0 & -0.32 & 0 & 0 & 0.64 & -0.32 \\ 0 & 0 & 0 & 0 & -0.14 & -0.15 & -0.32 & 0.61 \end{bmatrix}$$

As the weights are usually positive values, the matrix $\mathbf{L}$ is symmetric, real, and positive semi-definite, with non-negative diagonal entries and non-positive off-diagonal entries. To account for variations in node connectivity, a normalized version is often used:

$$\mathbf{L}_{norm} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}, \tag{3.10}$$

which refers to normalizing each edge weight $\mathbf{W}_{ij}$ by $\frac{1}{\sqrt{\mathbf{D}_{ii}\mathbf{D}_{jj}}}$. This formulation balances the influence of nodes with varying connectivity, ensuring that the Laplacian reflects relative differences in the graph more fairly, especially when degrees of nodes differ significantly. The graph Laplacian is the discrete analogue of the Laplacian operator in calculus, which quantifies the deviation of a function from its average value within a local neighborhood. In the context of graphs, it serves an analogous role by characterizing how a function defined on the vertices (e.g., a signal or scalar field) varies in relation to the

values on adjacent nodes.
The spectral theory of graphs is founded upon the mathematical properties of this matrix. As it is real symmetric positive semi-definite, it has a complete set of orthonormal eigenvectors $\{\mathbf{u}_l\}_{l=0}^{n-1}$ associated to non-negative eigenvalues $\{\lambda_l\}_{l=0}^{n-1}$ satisfying $\mathbf{L}\mathbf{u}_l = \lambda_l \mathbf{u}_l$, for $l = 0, 1, \ldots, n-1$. The multiplicity of the zero eigenvalue equals the number of connected components of the graph; in particular, a connected graph has a single zero eigenvalue. The first eigenvalue $\lambda_0$ is always equal to zero, which corresponds to a constant eigenvector $\mathbf{u}_0$, larger $\lambda_l$ indicates higher "frequency" eigenvectors. In analogy with the discrete Fourier transform on regular domains, the Graph Fourier Transform (GFT) is defined by taking the eigenvectors $\mathbf{u}_l$ of $\mathbf{L}$ as the Graph Fourier basis while the associated eigenvalues $\lambda_l$ serve as the graph frequencies.

A simple case of a graph signal is a vector $\mathbf{h} \in \mathbb{R}^{n\times 1}$, where each vertex $v \in \mathcal{V}$ in $\mathcal{G}$ is assigned a real value $h(v)$. The local variation of the signal on vertex $u$, is expressed as

$$\mathbf{L}\mathbf{h}(u) = \sum_{v\in\mathcal{N}_u} \mathbf{W}_{u,v}(h(u) - h(v)). \tag{3.11}$$

The global variation (or smoothness) of the signal is captured by the quadratic form:

$$\mathbf{h}^\top \mathbf{L}\mathbf{h} = \sum_{(u,v)\in\mathcal{E}} \mathbf{W}_{u,v}(h(u) - h(v))^2. \tag{3.12}$$

This expression can be rewritten using the spectral decomposition:

$$\mathbf{h}^\top \mathbf{L}\mathbf{h} = \sum_{l=0}^{n-1} \lambda_l \hat{h}(l)^2, \tag{3.13}$$

where $\hat{h}(l) = \mathbf{u}_l^\top \mathbf{x}$ is the projection of the signal onto the $l^{th}$ eigenvector, and $\hat{h}(l)^2$ is the energy of the signal in that mode. If $\mathbf{h}^\top \mathbf{L}\mathbf{h}$ is small, this means the signal is smooth: neighboring vertices have similar values. A large $\mathbf{h}^\top \mathbf{L}\mathbf{h}$ means that the signal changes sharply across connected vertices.

Having introduced the graph and its characteristics, a natural question emerges: what is the origin of this graph? The following section addresses this by examining various methodologies for constructing or inferring the graph structure from data.

### 3.1.3 Graph topology construction

Depending on the nature of the problem, the graph structure may be either known in advance, defined based on physical proximity, or requires computational inference from signal data. These scenarios can be broadly grouped as follows:
**(1) Graphs with known, domain-defined structures**: In many applications, the graph topology is explicitly defined by the structure of the system itself. For instance, in

molecular graphs, nodes represent atoms and edges represent chemical bonds [Duvenaud et al. 2015]. In social networks, edges correspond to relationships such as friendships or follower links [Easley et al. 2010], while in web graphs, hyperlinks define the connections between webpages [J. M. Kleinberg 1999]. Citation networks model references between academic publications [Ali et al. 2020], and Wikipedia edit graphs are constructed based on editing interactions among contributors [Brandes, Kenis, et al. 2009]. Similarly, in infrastructure networks such as transportation systems or power grids, nodes and edges correspond to physically observable components such as intersections, roads, substations, and transmission lines [X. Ma et al. 2021]. In the biomedical domain, Diffusion Tensor Imaging (DTI) on the brain produces anatomical brain networks by tracing white matter tracts between gray matter regions [Osipowicz et al. 2016], providing a clear, structurally defined connectivity map.

**(2) Graphs based on spatial proximity or distance:** In some sensor-based applications, the graph structure is usually defined using geographical or physical distances between nodes. For instance, in air quality monitoring, a graph can be built by connecting sensors based on their geodesic distance [Do et al. 2019]. Similarly, in traffic sensor networks, nodes may be linked according to their spatial layout (e.g. adjacent intersections or road segments) [Z. Zhang et al. 2021]. Likewise, in magnetoencephalography-based functional networks, the connectivity can be defined using geodesic distances between sensor locations on the dewar containing them [Mouches et al. 2024]. These applications rely on spatial priors to establish the graph structure without directly using signal-based measures.

**(3) Graphs inferred from signal-based computations:** In other scenarios, the graph topology is not explicitly known and must be derived through computational analysis of signal similarity or interaction. A prominent example is resting-state Functional magnetic resonance imaging (fMRI), where functional connectivity graphs are constructed by computing correlations between blood-oxygen-level-dependent (BOLD) signals across brain regions, reflecting synchronized neural activity [H. Tang et al. 2023]. Similarly, in dynamic traffic modeling, graphs can be inferred from signal similarity (e.g., flow or speed correlations) between sensor locations [Z. Zhang et al. 2021]. These approaches require graph learning or estimation techniques to uncover the latent connectivity structure that governs signal's behavior.

In the remainder of this manuscript, we focus on graph construction strategies corresponding to the latter two categories. In order to effectively process the constructed graphs, specialized neural network architectures are required. Unlike traditional data structures, these brain graphs are non-Euclidean, irregular, and lack the fixed local neighborhoods present in images or sequences. This motivates the use of graph-based learning models [Bronstein et al. 2017] which can naturally operate on such irregular structures.

## 3.2 Graph neural networks

Conventional deep learning models are typically designed for Euclidean domains such as sequences or images. Architectures like CNNs operate on regular grids like images, where local neighborhoods are fixed and uniformly structured (see [Fig. 3.4], left). However, many real-world domains involve complex relationships that are more naturally represented as graphs, where vertices are connected through non-uniform, domain-specific topologies. Graph Neural Networks extend the core principles of CNNs by enabling message passing and feature aggregation over irregular neighborhoods defined by the graph's topology (see [Fig. 3.4], right). For a more thorough analysis, the reader can refer to a comprehensive survey on graph neural networks in [Z. Wu et al. 2020].
In this manuscript, we focus on graph-level classification, where the objective is to make predictions based on the entire graph structure. Unlike node-level or edge-level tasks, this requires capturing global patterns and high-order interactions across the graph. Achieving this often relies on understanding and propagating local information between nodes. This brings us to the concepts of aggregation and message passing, which provide the fundamental mechanisms for exchanging information across nodes and building rich node representations that can later be pooled to form graph-level embeddings.

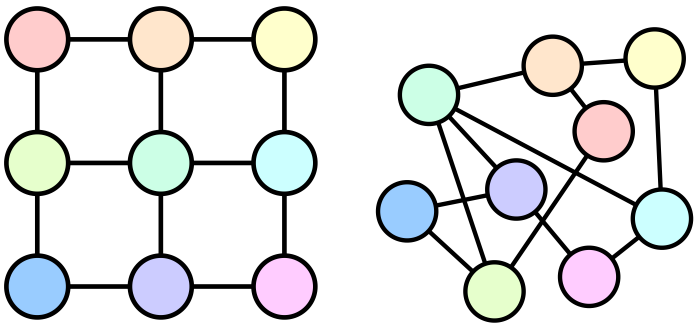


Figure 3.4 – The left diagram illustrates a regular 2D grid, typical of image data processed by CNN, where each node has a fixed neighborhood. The right diagram represents an irregular graph structure, where nodes are connected based on domain-specific relationships, as used in GNN.

### 3.2.1 Aggregation and message passing

**Definition 3.2.1 (Node message or feature vector)** *A node message, also called a node feature vector or graph signal, can be viewed as a signal defined on the graph. Instead of assigning a single scalar value to each node, it assigns a vector of values* $\bar{\mathbf{h}}_u$*, capturing multiple attributes or features associated with that node.*

The fundamental component of a GNN is the Message Passing Layer, which generalizes the convolution operation to graph-structured data. Unlike traditional convolutions

that operate on fixed-size neighborhoods in grid-like structures, message passing in GNNs accommodates variable neighborhood sizes, as each node may have a different number of neighbors. During message passing, each node aggregates information from its *1-hop* neighborhood to update its own representation as illustrated in [Fig. 3.5]. Specifically, for a given target node, messages are computed from the features of its neighboring nodes and possibly the edges connecting them. These messages are then combined using a permutation-invariant aggregation function $\Psi(.)$ that can be a sum, mean, or max, ensuring that the operation is independent of the order of the neighbors. Combining $k$ GNN layers means that each node is exchanging information with other nodes in its *k-hop* neighborhood. However, stacking a lot of message passing layers might cause oversmoothing.

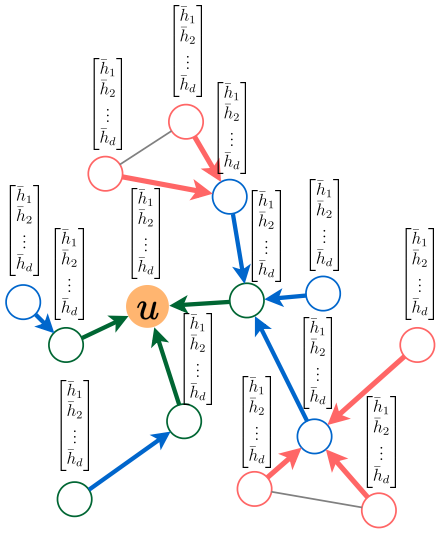



Figure 3.5 – Message passing process: vertex $u$ receives messages from its $\mathcal{N}_1$ neighbours if one message passing layer is used. These messages, derived from the feature vectors of the neighboring nodes, are then aggregated to update the representation of node $u$.

## 3.3 Graph representations for EEG

When inferring a graph representation from EEG signals, two main approaches can be distinguished: handcrafted and learned. Traditionally, handcrafted graphs rely on domain knowledge or predefined models of connectivity. However, these approaches are limited by prior assumptions and may fail to capture subject-specific or dynamic patterns. In contrast, learned graphs are estimated directly from the data, allowing for more flexible and adaptive representations. In the following sections, we introduce the theoretical foundations of both approaches.

### 3.3.1 Handcrafted graphs

To model EEG-recorded brain activity using graphs, researchers have extensively leveraged handcrafted connectivity measures. In particular, two forms of connectivity are frequently distinguished: effective connectivity, which assesses causality, and functional connectivity, which measures statistical dependencies between brain regions. Effective connectivity characterizes directed causal relationships by analyzing the temporal dependencies of neural signals. Since it falls outside the scope of the present thesis, only its definition is provided here, and the interested reader is directed to comprehensive reviews and tool descriptions for a deeper exploration [F. Huang et al. 2025]. The contributions presented in this thesis focus primarily on functional connectivity, which quantifies the statistical interdependence of physiological time series recorded from different brain regions. Given the high temporal resolution of EEG, functional connectivity is particularly suitable for studying brain network dynamics.

Formally, the multivariate EEG signal over a time segment $t$ is denoted as $\mathbf{X} \in \mathbb{R}^{n \times tF_s}$, where $n$ indexes the electrodes and $F_s$ is the sampling frequency. The signal recorded at electrode $e$ is denoted by $\mathbf{x}_e$. Relying on the multivariate EEG signals, a graph is constructed where a node corresponds to an electrode. The functional connectivity is then represented by a matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. The weights are estimated using a variety of metrics, typically grouped into categories including time-domain [L. Wei et al. 2025], frequency-domain [Nolte et al. 2004], phase synchronization metrics [Briels et al. 2020], information-theoretic approaches [Martínez Vásquez et al. 2023], and other measures [Bastos et al. 2016, Maitin et al. 2023]. A summary of common metrics is shown in [Tab. 3.2].

**Notation:**

- $\mathbf{x}_e$, $\mathbf{x}_f$: EEG signals at electrodes $e$ and $f$.
- $\mathbf{S}_{ef}$: covariance between $\mathbf{x}_e$ and $\mathbf{x}_f$.
- $\mathbf{S}_{ee}$: variance of $\mathbf{x}_e$.
- $tF_s$: number of time samples (signal length $\times$ sampling frequency).
- $\mu_e$: mean of signal $\mathbf{x}_e$.
- $(\cdot)^*$: complex conjugate.
- $\bar{f}$: frequency bin.
- $PSD_{\mathbf{x}_e}^{(f)}$: power spectral density of $\mathbf{x}_e$ at frequency $\bar{f}$.
- $G_{\mathbf{x}_e\mathbf{x}_f}^{(f)}$: cross-spectral density between $\mathbf{x}_e$ and $\mathbf{x}_f$ at frequency $\bar{f}$.
- $\phi_{\mathbf{x}_e(k)}$: instantaneous phase of $\mathbf{x}_e$ at time $k$ (via Hilbert transform).
- $u$, $v$: bin from the discretized range of $\mathbf{x}_e$ and $\mathbf{x}_f$, respectively.
- $P(u)$, $P(v)$: marginal probability distributions of discretized signals $\mathbf{x}_e$ and $\mathbf{x}_f$.
- $P(u, v)$: joint probability distribution of $\mathbf{x}_e$ and $\mathbf{x}_f$.

Table 3.2 – Summary of common EEG connectivity measures.

| | Measure | Description / Formula |
|---|---|---|
| *Time* | Pearson correlation | Instantaneous linear similarity between two signals. $\mathbf{W}_{ef} = \frac{\mathbf{S}_{ef}}{\sqrt{\mathbf{S}_{ee}\mathbf{S}_{ff}}}$ |
| | Cross-correlation | Similarity as a function of lag $m$ between signals. $\mathbf{W}_{ef}^{(m)} = \sum_{k=0}^{tF_s-1}(\mathbf{x}_e{}^{(k)} - \mu_e)(\mathbf{x}_f{}^{(k-m)} - \mu_f)^*$ |
| *Frequency* | Squared coherence | Linear coupling at frequency $\bar{f}$ via normalized cross-spectral density. $\mathbf{W}_{ef}^{(\bar{f})} = \frac{\lvert G_{\mathbf{x}_e\mathbf{x}_f}^{(\bar{f})}\rvert^2}{PSD_{\mathbf{x}_f}^{(\bar{f})} PSD_{\mathbf{x}_j}^{(\bar{f})}}$ |
| | Imaginary coherence | Uses only imaginary part of coherence to reduce volume conduction. $\mathbf{W}_{ef}^{(\bar{f})} = \frac{\mathrm{Im}\{G_{\mathbf{x}_e\mathbf{x}_f}^{(\bar{f})}\}}{\sqrt{PSD_{\mathbf{x}_e}^{(\bar{f})} PSD_{\mathbf{x}_f}^{(\bar{f})}}}$ |
| *Phase* | Phase locking value | Consistency of phase difference across samples. $\mathbf{W}_{ef} = \left\lvert \frac{1}{tF_s}\sum_{k=1}^{tF_s} e^{-\mathrm{i}(\phi_{\mathbf{x}_e(k)} - \phi_{\mathbf{x}_f(k)})}\right\rvert$ |
| | Phase lag index | Non-zero phase locking based on sign of phase differences. $\mathbf{W}_{ef} = \left\lvert \frac{1}{tF_s}\sum_{k=1}^{tF_s} \mathrm{sign}[\sin(\phi_{\mathbf{x}_e(k)} - \phi_{\mathbf{x}_f(k)})]\right\rvert$ |
| *Info* | Mutual information | Reduction in uncertainty of $\mathbf{h}_e$ given $\mathbf{h}_f$. $\mathbf{W}_{ef} = \sum_{u\in\mathbf{x}_e}\sum_{v\in\mathbf{x}_f} P(u,v)\,\log\left(\frac{P(u,v)}{P(u)P(v)}\right)$ |

Functional connectivity matrices derived from measures like Pearson correlation, PLI, and others listed in [Tab. 3.2] are often binary-thresholded (e.g., selecting the top 10% of connections) to build sparse graph representations by selecting only important connections [Adamovich et al. 2022]. Subsequent graph-theoretic metrics (e.g., clustering coefficient, local efficiency, path length) allow characterization of network topology [Rubinov et al. 2010].

### 3.3.2 Learned graphs

More recently, the focus has shifted to data-driven or learned graphs, where the structure is estimated directly from the observed multivariate data. To set the stage for discussing how graphs can be learned from data, such as EEG signals, we first introduce some foundational definitions.

**Definition 3.3.1 (Multivariate process)** *A multivariate process involves multiple interdependent variables. In the context of EEG, the multichannel signal can be modeled as a multivariate process, where each channel (electrode) corresponds to a variable.*

**Definition 3.3.2 (Independence)** *Let $X$ and $Y$ be two random variables. They are*

*said to be* ***statistically independent*** *if and only if their joint probability factorizes as:*

$$P_{XY}(x, y) = P_X(x)\, P_Y(y),$$

*where $P_X$ and $P_Y$ are the marginal distributions of $X$ and $Y$, respectively.*

**Definition 3.3.3 (Conditional independence)** *Let $X$, $Y$, and $Z$ be three random variables. $X$ and $Y$ are said to be* ***conditionally independent given*** *$Z$ if:*

$$P_{XY|Z}(x, y \mid z) = P_{X|Z}(x \mid z)\, P_{Y|Z}(y \mid z),$$

*In a multivariate setting, $Z$ typically represents all variables in the system except $X$ and $Y$, and conditional independence captures the absence of direct interaction between $X$ and $Y$ given the rest.*

**Definition 3.3.4 (Sparse matrix)** *A matrix is said to be sparse if most of its elements are zero. In brain connectivity modeling, sparsity implies that only a small subset of electrode pairs are conditionally dependent.*

**Definition 3.3.5 (Precision matrix)** *In Gaussian graphical models (GGMs), the key quantity is the precision matrix $\mathbf{\Omega}$ defined as the inverse of the covariance matrix $\mathbf{S}$. Zeros in $\mathbf{\Omega}$, correspond to conditional independencies between variables.*

We aim to estimate the underlying brain connectivity graph from multivariate EEG signals by modeling the data as a sample from a probabilistic graphical model. Given the multivariate matrix $\mathbf{X} \in \mathbb{R}^{n \times tF_s}$, the goal is to infer the most probable graph $\mathcal{G}$ explaining the data. This can be formalized as computing the posterior distribution over graphs:

$$P(\mathcal{G} \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid \mathcal{G}) P(\mathcal{G})}{P(\mathbf{X})}, \tag{3.14}$$

where:

— $P(\mathbf{X} \mid \mathcal{G})$ is the likelihood of the data given the graph,

— $P(\mathcal{G})$ is a prior distribution over possible graph structures,

— $P(\mathbf{X})$ is the marginal likelihood (normalization constant).

Under the assumption that the data follows a *multivariate Gaussian distribution* with zero mean, the graph structure can be fully encoded by the sparsity pattern of the *precision matrix* $\mathbf{\Omega} = \mathbf{S}^{-1}$, where $\mathbf{S}$ is the covariance matrix of the distribution. The non-zero entries of $\mathbf{\Omega}$ correspond to edges in the graph, representing direct (conditional) dependencies between variables.

The classical approach to graph structure learning originates with [Dempster 1972], who proposed introducing zeros into the inverse covariance matrix to model conditional

independencies. This method requires inverting the empirical covariance matrix, which can be singular or poorly conditioned. To overcome this, other approaches impose sparsity through regularization. The graph inference task in [Eq. 3.14] can be recast as a maximum a posteriori (MAP) estimation problem:

$$\mathbf{\Omega}^{\star} = \arg\max_{\mathbf{\Omega} \succ 0} \; P(\mathbf{\Omega} \mid \mathbf{X}) = \arg\max_{\mathbf{\Omega} \succ 0} \; P(\mathbf{X} \mid \mathbf{\Omega})\, P(\mathbf{\Omega}), \tag{3.15}$$

where:

— $P(\mathbf{X} \mid \mathbf{\Omega})$ is the likelihood under the multivariate Gaussian model,

— $P(\mathbf{\Omega})$ is a prior that encourages sparse solutions.

The probability density function for a signal $\mathbf{x}_i$ is written as:

$$P(\mathbf{x}_i \mid \mathbf{\Omega}) = \frac{1}{(2\pi)^{n/2} |\mathbf{\Omega}^{-1}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{x}_i^{\top} \mathbf{\Omega} \mathbf{x}_i\right). \tag{3.16}$$

Assuming $tF_s$ independent and identically distributed (i.i.d.) samples, the likelihood of the multivariate matrix $P(\mathbf{X} \mid \mathbf{\Omega})$ becomes:

$$P(\mathbf{X} \mid \mathbf{\Omega}) = \prod_{i=1}^{tF_s} \frac{1}{(2\pi)^{n/2} |\mathbf{\Omega}^{-1}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{x_i}^{\top} \mathbf{\Omega} \mathbf{x_i}\right), \tag{3.17}$$

and the log-likelihood of the multivariate matrix becomes:

$$\begin{aligned}
\log P(\mathbf{X} \mid \mathbf{\Omega}) &= \sum_{i=1}^{tF_s} \Big[ -\tfrac{n}{2} \log(2\pi) - \tfrac{1}{2} \log |\mathbf{\Omega}^{-1}| - \tfrac{1}{2} \mathbf{x_i}^{\top} \mathbf{\Omega} \mathbf{x_i} \Big] \\
&= -\tfrac{tF_s n}{2} \log(2\pi) - \tfrac{tF_s}{2} \log |\mathbf{\Omega}^{-1}| - \tfrac{1}{2} \sum_{i=1}^{tF_s} \mathbf{x_i}^{\top} \mathbf{\Omega} \mathbf{x_i}, \\
&= -\tfrac{tF_s n}{2} \log(2\pi) + \tfrac{tF_s}{2} \log\det(\mathbf{\Omega}) - \tfrac{tF_s}{2} \operatorname{tr}(\mathbf{S}\mathbf{\Omega}), \\
&= \tfrac{tF_s}{2} \Big( \log\det(\mathbf{\Omega}) - \operatorname{tr}(\mathbf{S}\mathbf{\Omega}) \Big) + \text{const.}
\end{aligned}$$

where $\mathbf{S} = \frac{1}{tF_s} \sum_{i=1}^{tF_s} \mathbf{x_i} \mathbf{x_i}^{\top}$ is the sample covariance matrix, and $tr()$ refers to the trace operator $\sum_{i=1}^{tF_s} \mathbf{x_i}^{\top} \mathbf{\Omega} \mathbf{x_i} = \operatorname{tr}\Big(\mathbf{\Omega} \sum_{i=1}^{tF_s} \mathbf{x_i} \mathbf{x_i}^{\top}\Big) = tF_s \operatorname{tr}(\mathbf{\Omega}\mathbf{S})$. To promote sparsity in the precision matrix, a Laplace ($l_1$) prior is imposed on its entries. This formulation yields a MAP estimation problem commonly known as the *Graphical Lasso* [Friedman et al. 2008], which can be expressed as the following convex optimization problem:

$$\mathbf{\Omega}^{\star} = \arg\max_{\mathbf{\Omega} \succ 0} \; \log\det \mathbf{\Omega} - \operatorname{tr}(\mathbf{S}\mathbf{\Omega}) - \zeta \|\mathbf{\Omega}\|_1, \tag{3.18}$$

where $\|\mathbf{\Omega}\|_1$ denotes the element-wise $\ell_1$-norm, and $\zeta > 0$ controls the level of sparsity and $\mathbf{\Omega} \succ 0$ ensures that the precision matrix is symmetric and positive definite.

### 3.3.3 Graph learning under Laplacian and structural constraints

The standard *Graphical Lasso* formulation estimates a positive definite precision matrix $\mathbf{\Omega}$ without imposing any explicit constraints on the sign of its off-diagonal elements. Consequently, the resulting precision matrix may contain both positive and negative entries. However, in the context of EEG analysis, we are primarily interested in modeling positive interactions between vertices (electrodes), which correspond to the presence of edges in the underlying graph. To enforce this desirable property, it is necessary to introduce additional constraints during the graph-learning process.

To address this limitation, we adopt the approach proposed by [Egilmez et al. 2017], more precisely, the generalized graph Laplacian (GGL) framework. In this formulation, the graph structure represented by the precision matrix $\mathbf{\Omega}$ is learned directly from the data while enforcing both Laplacian and structural constraints.

**Laplacian constraint.** Recall that a *generalized graph Laplacian* $\mathbf{L}$ is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{W} + \mathbf{V},$$

where $\mathbf{W}$ is the weighted adjacency matrix, $\mathbf{D}$ is the degree matrix, and $\mathbf{V}$ is a diagonal matrix of self-loops. By construction, graph Laplacians are symmetric positive semidefinite and have *non-positive off-diagonal entries* ($\mathbf{L}_{ef} = -\mathbf{W}_{ef}$), which implies that the corresponding edge weights $\mathbf{W}_{ef}$ are **non-negative**. Enforcing a Laplacian structure ensures that the learned graph naturally encodes **positive edge weights**, consistent with the notion of positive functional interactions between EEG electrodes:

$$\mathcal{L} = \left\{ \mathbf{L} \;\middle|\; \mathbf{L} \succeq 0,\ \mathbf{L}_{ef} \leq 0 \text{ for } e \neq f \right\}. \tag{3.19}$$

**Structural constraint.** Additional prior knowledge about the graph structure (e.g., electrode layout or known connectivity) can be incorporated via an *adjacency matrix* $\mathbf{A}$, which specifies the set of admissible edges. When $\mathbf{A}$ is provided, the learning process is restricted to estimate weights only on allowed edges: if $\mathbf{A}_{ef} = 0$, the algorithm enforces $\mathbf{\Omega}_{ef} = 0$. If the underlying adjacency is unknown, the algorithm will compute the graph structure as well as the edge weights. The feasible set of precision matrices becomes:

$$\mathcal{L}(\mathbf{A}) = \left\{ \mathbf{\Omega} \in \mathcal{L} \;\middle|\; \mathbf{\Omega}_{ef} \leq 0 \text{ if } \mathbf{A}_{ef} = 1,\ \mathbf{\Omega}_{ef} = 0 \text{ if } \mathbf{A}_{ef} = 0,\ e \neq f \right\}. \tag{3.20}$$

**Optimization formulation.** Similar to [Eq. 3.18], the GGL framework solves

$$\mathbf{\Omega}^{\star} = \arg\min_{\mathbf{\Omega} \succ 0} \underbrace{\operatorname{tr}(\mathbf{S}\mathbf{\Omega}) - \log\det(\mathbf{\Omega})}_{\text{Data fidelity}} + \underbrace{\|\mathbf{\Omega} \odot \mathbf{H}\|_1}_{\text{Sparsity / regularization}}, \tag{3.21}$$

where $\mathbf{S}$ is the sample covariance matrix and $\mathbf{H}$ is a regularization matrix weighting the $\ell_1$ penalty. The regularization term can be expressed as

$$\|\mathbf{\Omega} \odot \mathbf{H}\|_1 = \sum_{i,j} \mathbf{H}_{ij} \, |\mathbf{\Omega}_{ij}|,$$

which is non-smooth at $\mathbf{\Omega}_{ij} = 0$. To obtain a smooth approximation, $\mathbf{H}$ is often chosen as

$$\mathbf{H} = \alpha_0 \sqrt{\|\mathbf{S}\|_F} \cdot \left[\mathbf{I} - \mathbf{1}_N \mathbf{1}_N^T\right],$$

where $\sqrt{\|\mathbf{S}\|_F}$ is chosen to normalize the penalty to the scale of the data. Where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathbf{I}$ is the identity matrix, and $\mathbf{1}_N \mathbf{1}_N^T$ is the all-ones matrix, and $\alpha_0$ a positive constant. With this choice, the regularization term simplifies to $\mathrm{tr}(\mathbf{\Omega H})$.

The trace operator being linear, the non smooth function in [Eq. 3.21] becomes smooth and is expressed as:

$$\mathbf{\Omega}^\star = \underset{\mathbf{\Omega} \in \mathcal{L}(\mathbf{A})}{\arg\min} \; \mathrm{tr}(\mathbf{\Omega K}) - \log\det(\mathbf{\Omega}), \tag{3.22}$$

where $\mathbf{K} = \mathbf{S} + \mathbf{H}$. The sparsity pattern of the resulting precision matrix (interpreted as a *generalized graph Laplacian*) provides insights into the *conditional independence* relationships between EEG signals [C. Zhang et al. 2015]. Non-zero off-diagonal elements $\mathbf{\Omega}_{ef}$ indicate positive partial correlations between the corresponding electrodes $e$ and $f$.

**Solving the problem.** The aforementioned optimization problem is convex; however, solving it directly using standard gradient descent is computationally expensive, as each iteration would require computing the inverse of the current Laplacian estimate. To address this issue, the authors in [Egilmez et al. 2017] propose decomposing the original problem into simpler subproblems by employing an *iterative block-coordinate descent* algorithm. At each iteration, the algorithm updates one row and the corresponding column of the Laplacian $\mathbf{\Omega}$, as well as the associated entries of its inverse, while keeping the remaining blocks fixed. Each block update satisfies an optimality condition that leads to a closed-form expression for the subproblem, which reduces to a *non-negative quadratic program* solvable by efficient dedicated solvers. Projections onto the Laplacian constraint set are applied when necessary to ensure that the updated precision matrix remains a valid graph Laplacian. For further details on the solver and its derivation, the reader is referred to Section V in [Egilmez et al. 2017].

In the frameworks proposed in this manuscript, we rely on this algorithm as the graph-learning backbone. While the original implementation was provided in MATLAB, we re-implemented it in Python to integrate it into our processing pipelines and further reduce the computation time by introducing parallelization. For each multivariate EEG window, a graph Laplacian $\mathbf{L}$ is learned as described in [Sec. 5.2.2], using $\alpha_0 = 0.01$, with probability and inner tolerances set to $10^{-2}$ and $10^{-6}$, respectively.

## 3.4 Conclusion

In this chapter, we introduced the theoretical foundations of *Graph Signal Processing* and its role in extending classical signal analysis to data supported on irregular domains. We reviewed the fundamental concepts of graphs and graph signals, discussed strategies for defining or learning the underlying graph topology, and presented graph neural network architectures designed to exploit these representations for inference tasks. Building on these principles, we detailed how graphs can be employed to model EEG signals, either by computing pairwise statistical connectivity measures or by learning the graph structure directly from data under structural and Laplacian constraints. This framework enables richer representations of the brain's functional organization and supports more effective classification of EEG-based tasks. The methods and insights outlined here form the groundwork for the following chapters. Before introducing the main contributions of this thesis, the next chapter presents the datasets processed and used in our experiments.

# Chapter 4

# EEG Datasets


4.1 COG-BCI for mental workload classification . . . . . 46
4.1.1 Experimental setting . . . . . 46
4.1.2 Signal pre-processing . . . . . 48
4.1.3 Baseline results reported in [Roy et al. 2022] . . . . . 48
4.1.4 Our experimental setup and baseline results . . . . . 49
4.2 HG-MI for mental motor imagery classification . . . . . 50
4.2.1 Experimental setting . . . . . 50
4.2.2 Signal pre-processing . . . . . 50
4.2.3 Baseline results . . . . . 51
4.3 Conclusion . . . . . 52

The proposed frameworks are evaluated on two distinct datasets, each corresponding to one of the primary applications addressed in this thesis: mental workload and motor imagery classifications. The details of each dataset are presented in the following sections.

## 4.1 COG-BCI for mental workload classification

The first dataset used to evaluate the proposed frameworks originates from the 2021 Neuroergonomics Passive BCI Hackathon [Roy et al. 2022]. It is a publicly available EEG dataset that includes multiple participants recorded in several sessions. The objective of the hackathon was to classify mental workload across sessions, that is, to predict a subject's workload in one session using data from the other two sessions. The complete dataset, later released in [Hinss et al. 2023], includes recordings from 29 participants (11 female, 18 male) with a mean age of 23.9 years. Most were students, some were employees; 14 participants had a bachelor's degree and 13 a master's degree.

### 4.1.1 Experimental setting

The experiment was carried out over three sessions, each spaced one week apart. During each session, participants performed four cognitive tasks: the N-back task, the Multi-Attribute Task Battery-II (MATB-II), the Psychomotor Vigilance Task (PVT), and the arrow-based Eriksen flanker task. To ensure consistent electrode placement and account for within- and between-subject variability, a 3D image of each participant wearing the EEG cap was taken at the start of every session. Each session began with brief training on the four tasks (detailed later), followed by two minutes of resting-state: one minute with eyes open and one minute with eyes closed. The tasks were then performed twice in a pseudorandomized order. Upon completion, participants were given a break. Task execution lasted between 65 and 80 minutes, depending on break duration, making each session approximately two hours long, with the first session typically taking more time due to initial setup. In this thesis, we focus on classifying mental workload - corresponding to the difficulty of the task - based on the MATB-II task. However, for completeness, we briefly describe the other tasks performed during the experimental sessions:

**N-back Task.** In this task, participants are shown a sequence of letters on a computer screen, each displayed for a short duration. They are asked to press a button whenever the current letter matches the one presented N steps earlier in the sequence. The value of N determines the task's difficulty, with higher values requiring greater working memory load.

**Flanker Task.** This task assesses response inhibition and conflict resolution. In the arrow-based version, participants are shown a horizontal row of five arrows (e.g., <<><< or >>>>>) and must respond based on the direction of the central arrow while ignoring

the surrounding flankers (> and > in this example). The task induces cognitive conflict when the flanking arrows point in the opposite direction from the target.
**PVT.** The PVT measures sustained attention and reaction time. Over a 10-minute period, participants are instructed to press the space bar as quickly as possible whenever a visual timer appears on the screen.
**MATB-II Task.** Developed by NASA, it induces three levels of mental workload by varying the number of concurrent sub-tasks the participant must perform. For this dataset, four specific sub-tasks were selected shown in [Fig. 4.1]:

1. **TRACK:** Participants are presented with a moving target within a window and must use a joystick to keep the target inside a central square area. Task difficulty can be adjusted by modifying the target's speed and movement pattern.
2. **SYSMON (System Monitoring):** Participants monitor gauges and warning lights. They must react in three situations: (i) absence of green lights, (ii) presence of red lights, and (iii) deviations of four moving dials from a midpoint. Reactions are performed using keys F5, F6, and F1–F4, respectively. Difficulty increases with the number of events requiring a response.
3. **COMM (Communication):** Participants listen to radio messages and respond only to those directed at them by selecting specific radio channels and adjusting frequencies. Responses are made via joystick-based button presses. Task complexity is increased by varying the number of relevant and irrelevant messages.
4. **RESMAN (Resource Management):** Participants manage an interface displaying two main fuel tanks (A and B) and four subsidiary tanks (C to F), interconnected through eight pumps (controlled via keys 1–8). The objective is to maintain appropriate fuel levels in the main tanks by activating or deactivating pumps. Task difficulty can be raised by introducing events such as pump failures.

All participants completed three independent runs, each lasting five minutes and corresponding to a different difficulty level: *easy* (including SYSMON and TRACK), *medium* (SYSMON, TRACK, and RESMAN), and *hard* (all four tasks: SYSMON, TRACK, COMM, and RESMAN).

**Signal aquisition** EEG signals were acquired using 64 Ag-AgCl electrodes and an amplifier. Since the Cz electrode was not recorded for the first nine participants, it was excluded for all participants to ensure consistency. One electrode was dedicated to recording electrocardiographic (ECG) activity. After preprocessing, a total of 61 electrodes were retained for analysis. At the beginning of each session, a 3D scan of the participant's head was performed, and the precise positions of the electrodes (following the *10-20 system*) were recorded as $(\mathsf{x}, \mathsf{y}, \mathsf{z})$ coordinates.

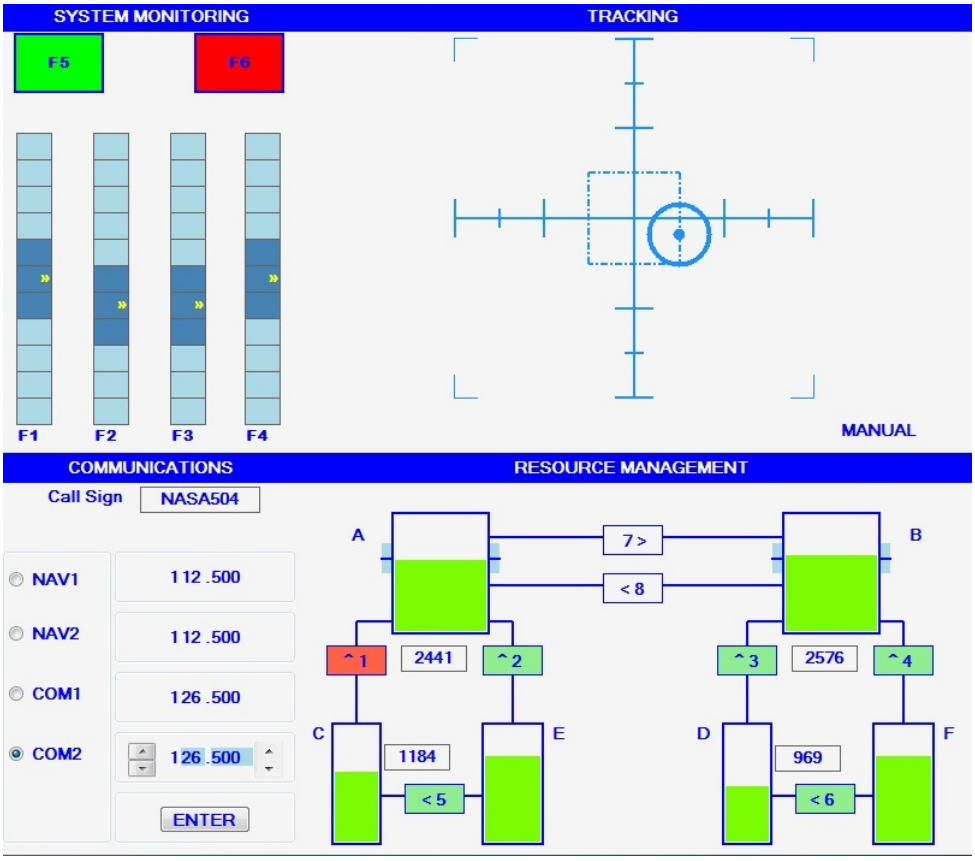



Figure 4.1 – The MATB-II tasks extracted from [https://matb.larc.nasa.gov]

### 4.1.2 Signal pre-processing

As this manuscript does not address EEG signal denoising or artifact removal, we adopted the same pre-processing pipeline used by the authors in [ Roy et al. 2022]. These steps were designed to enhance the signal-to-noise ratio and include the following: epoching into non-overlapping 2-second segments; referencing to the right mastoid electrode; high-pass filtering at 1 Hz using a FIR filter; electrode rejection based on abnormal amplitudes (i.e., average amplitude exceeding twice the standard deviation across channels), followed by spherical interpolation; application of SOBI (Second Order Blind Identification), a blind source separation method based on second-order statistics, with automatic rejection of independent components corresponding to muscle, heart, or eye activity using a 95% confidence threshold; low-pass filtering at 40 Hz with a FIR filter to reduce contamination from motion-related electrooculographic (EOG) and electromyographic (EMG) artifacts; common average referencing (CAR); and finally, downsampling to 250 Hz.

### 4.1.3 Baseline results reported in [Roy et al. 2022]

In their study, the authors proposed a three-class mental workload classification approach based on MATB-II tasks using a R-MDM classifier applied to covariance matrices

computed from EEG signal windows. They selected a subset of 10 electrodes (F3, Fz, F4, FCz, C3, C4, CPz, P3, Pz, P4) covering frontal and posterior cortical areas. Using the R-MDM classifier, they achieved an overall accuracy of $0.69 \pm 0.12$. They further analyzed performance in specific frequency bands, obtaining accuracies of $0.68 \pm 0.12$ in the alpha band and $0.66 \pm 0.13$ in the theta band. They reported that, the classification accuracy was higher in the second session $0.70 \pm 0.13$ compared to the first session $0.64 \pm 0.12$.

### 4.1.4 Our experimental setup and baseline results

In order to increase the size of the dataset for all the methods presented in this manuscript and tested on the COG-BCI dataset, the cleaned EEG signal windows are reassembled and then segmented into 2-second windows with a 1-second overlap. To avoid data leakage between training and test sets for each subject, these overlapping windows are grouped based on their sequential order. The final window in each group is excluded from the experiments to maintain clear boundaries, see [Fig. 4.2] for a visual representation. Model performance is evaluated using 5-fold cross-validation, where shuffling and train-test splits are applied at the group level rather than directly on individual windows. Furthermore, the groups are fixed and used the same across the different approaches for a fair comparison.

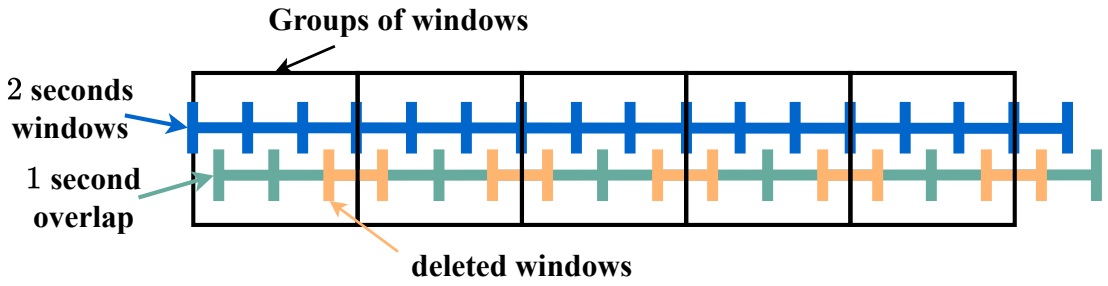



Figure 4.2 – Decomposing the signal into groups of overlapping windows for COG-BCI dataset.

For a fair comparison with our proposed approaches, we implemented the same pipeline based on the R-MDM classifier using the overlapping window grouping strategy described above. We obtained slightly lower results, with an accuracy of $0.68 \pm 0.11$ for session 2 and $0.65 \pm 0.06$ for session 1, confirming the trend that session 2 yields better performance than session 1. Band-specific results showed accuracies of $0.59 \pm 0.09$ in the Theta band and $0.60 \pm 0.11$ in the Alpha band. Additionally, when using all available electrodes instead of a subset, the classification accuracy reached $0.62 \pm 0.08$. We also reported the performances of another state of the art approach: the CSP method combined with an OVR strategy and a LDA classifier. Applied to the same window groups, this method achieved an accuracy of $0.75 \pm 0.14$ subject-wise.

## 4.2 HG-MI for mental motor imagery classification

The second dataset used to evaluate the methods proposed in this manuscript is the High-Gamma Motor Imagery dataset [Schirrmeister et al. 2017], which includes EEG recordings from 14 healthy participants (6 female, 2 left-handed; mean age: $27.2 \pm 3.6$ years). Participants performed four classes of motor tasks: left-hand movement, right-hand movement, bilateral foot movement, and rest, over approximately 1000 trials per subject. Each trial lasted 4 seconds and trials were distributed across 13 runs.

### 4.2.1 Experimental setting

Participants were seated comfortably in an armchair within a dedicated cabin. Visual cues were displayed on a monitor located outside the cabin and viewed through a shielded window. A fixation point was placed at the center of the screen, and participants were instructed to stay relaxed, fixate on the point, and remain as still as possible throughout the motor execution tasks. They were instructed to relax, fixate the fixation mark and to keep as still as possible during the motor execution task. Blinking and swallowing was restricted to the inter-trial intervals.
The task consisted of responding to directional gray arrows presented against a black background. Depending on the arrow's orientation, participants were instructed to: clench their toes (downward arrow), perform sequential finger tapping with the left hand (leftward arrow), with the right hand (rightward arrow), or remain relaxed (upward arrow). These movements were chosen to minimize proximal muscle activity while remaining cognitively engaging.
Each trial lasted 4 seconds, during which the indicated movement was performed continuously at the participant's own pace. A total of 80 trials were presented per run, each followed by a randomized inter-trial interval lasting between 3 and 4 seconds. The arrow presentation was pseudo-randomized, ensuring that all four movement conditions were represented once every four trials.

**Signal acquisition** EEG signals were recorded using 128 electrodes at a sampling rate of $F_s = 500$ Hz. Only the 44 electrodes covering the motor cortex were selected for analysis. Specifically, all central electrodes (45 in total) were considered, except for the Cz electrode, which served as the recording reference.

### 4.2.2 Signal pre-processing

The dataset was downloaded and preprocessed using the Mother Of All BCI Benchmarks (MOABB), an open-source Python library that standardizes the evaluation of BCI algorithms on EEG datasets [Aristimunha et al. 2025]. Pre-processing depends on the

selected paradigm, which defines how raw EEG recordings are segmented into trials suitable for machine learning. This includes bandpass filtering, epoch extraction based on stimulus events, and optional resampling to match the requirements of specific pipelines. To ensure fair algorithm comparisons, we adopted the motor imagery (MI) paradigm with four classes, applying a bandpass filter in the $[8, 32]Hz$ range, and maintained the same size of windows. Note that in this case since we do not have a continuous signal for each task, it is not possible to have overlapping windows as it is the case for COG-BCI dataset, therefore we use non overlapping windows for this dataset.

### 4.2.3 Baseline results

After pre-processing, the number of windows per class for each subject is shown in [Tab. 4.1] Results are reported for different pipelines on this dataset. R-MDM classifier

Table 4.1 – HG-MI dataset: number of epochs per class and participant.

| **Participant** | **Feet** | **Left Hand** | **Rest** | **Right Hand** | **Total** |
|---|---|---|---|---|---|
| P1 | 120 | 120 | 120 | 120 | 480 |
| P2 | 244 | 242 | 243 | 244 | 973 |
| P3 | 260 | 260 | 260 | 260 | 1040 |
| P4 | 265 | 264 | 264 | 264 | 1057 |
| P5 | 220 | 220 | 220 | 220 | 880 |
| P6 | 260 | 260 | 260 | 260 | 1040 |
| P7 | 260 | 260 | 260 | 260 | 1040 |
| P8 | 203 | 204 | 203 | 204 | 814 |
| P9 | 260 | 260 | 260 | 260 | 1040 |
| P10 | 260 | 260 | 260 | 260 | 1040 |
| P11 | 260 | 260 | 260 | 260 | 1040 |
| P12 | 260 | 260 | 260 | 260 | 1040 |
| P13 | 240 | 240 | 240 | 240 | 960 |
| P14 | 260 | 260 | 260 | 260 | 1040 |
| **Mean ± std** | 240.9 ± 39.2 | 240.7 ± 39.1 | 240.7 ± 39.2 | 240.9 ± 39.1 | 963.1 ± 156.6 |

achieved an accuracy of $0.52 \pm 0.10$ [Barachant et al. 2012]. The CSP method combined with a LDA classifier reached $0.73 \pm 0.10$, while CSP combined with a SVM achieved $0.76 \pm 0.11$. The highest performance on this dataset was obtained using the tangent space method, with an accuracy of $0.85 \pm 0.09$ [Corsi et al. 2022].

## 4.3 Conclusion

In this chapter, we have reviewed the two EEG datasets that will provide the basis for evaluating our proposed graph representations and classification pipelines. We described the experimental settings, signal acquisition procedures, and pre-processing steps, along with summarizing the baseline classification results reported in the literature. Having established both the theoretical foundations of EEG signals and graph-based representations, as well as the practical context of the datasets, we now move on to the core contributions of this thesis. Specifically, we focus on how the intrinsic structure of brain connectivity, as reflected in EEG data, can be effectively modeled and classified, through graph representations.

Part II

# Graph-Based Representation and Classification

# Chapter 5

# Functional Graph to Image Representation


5.1 Introduction . . . 56
5.1.1 Classification based on raw EEG signals . . . 57
5.1.2 Classification based on two-dimensional EEG representations . . . 57
5.1.3 Classification on connectivity-based EEG representations . . . 58
5.1.4 Discussion . . . 60
5.2 A graph image classification framework . . . 60
5.2.1 Functional graph construction . . . 61
5.2.2 Functional graph to image representation . . . 62
5.2.2.1 Finding the pixel indices corresponding to the electrode coordinates . . . 62
5.2.2.2 Defining the pixel values . . . 63
5.2.3 Graph image classification . . . 65
5.3 Experimental setup . . . 67
5.4 Experimental results and analysis . . . 68
5.4.1 Impact of the of the structural constraint on the Laplacian . . . 68
5.4.2 Pearson correlation vs partial correlation based connectivity . . . 69
5.4.3 Comparison against baseline classification pipelines . . . 70
5.4.4 Interpretability and explainability . . . 72
5.5 Conclusion . . . 74

In this chapter, we turn to the first proposed pipeline, based on a Functional Graph to Image (FG2IMG) representation, which addresses EEG classification through functional connectivity. This pipeline is built around two key components. First, we introduce the method for constructing graphs from EEG data based on graph Laplacian learning, where the nodes correspond to electrodes and the edges capture functional relationships between them. Second, these learned graphs are transformed into two-dimensional image representations that explicitly highlight both the spatial arrangement of the electrodes and their connectivity structure. The resulting images are then used as inputs to a convolutional neural network, which performs the classification task.

Since CNN-based classification of EEG signals has been widely explored in the literature, we begin this chapter by reviewing existing pipelines that exploit different domains: time, frequency, time–frequency, spatial, and connectivity for generating CNN-compatible representations. We then present our approach in detail: starting with Laplacian learning from EEG signals, followed by the graph-to-image transformation, and concluding with the chosen CNN architecture. Finally, we provide experimental results and comparisons to demonstrate the feasibility and effectiveness of the FG2IMG framework in the context of both mental workload and motor imagery classification.

## 5.1 Introduction

Classical machine learning approaches, presented in [Sec. 2.4], rely on manually extracting features from the data prior to classification. However, with the increasing availability of EEG datasets and the rapid progress of deep learning methods, research has progressively moved toward end-to-end deep learning approaches for EEG classification. These methods are particularly appealing, as they enable the automatic learning of hierarchical feature representations directly from the data, although some studies have combined handcrafted feature extraction with deep learning classifiers [Craik et al. 2019].

Within deep learning methods, CNNs have been among the most widely adopted, with numerous studies proposing image-based representations of EEG signals. A substantial body of work has emerged in this direction, introducing different strategies for constructing CNN-compatible inputs from EEG data. These approaches can be broadly grouped into three categories: (i) methods that use raw EEG signals directly as input to CNNs; (ii) methods that transform EEG signals into two-dimensional representations such as time–frequency maps or topographic images prior to classification; and (iii) methods that rely on functional connectivity and connectivity-based representations to generate CNN inputs. In the following, we provide a literature review structured around these three categories.

### 5.1.1 Classification based on raw EEG signals

This category encompasses approaches that directly feed raw EEG time series into deep learning models, thereby bypassing explicit feature engineering. For example, [Hussein et al. 2019] employed non-overlapping EEG segments as input to a Long Short-Term Memory (LSTM) network, allowing the capture of both short- and long-term temporal dependencies, and reported superior classification performance compared to baseline methods on the same dataset for seizure detection. Similarly, [Yanagimoto et al. 2016] evaluated the use of raw EEG signals, without any artifact removal, as input to a CNN model for emotion recognition. Their results showed that training on data pooled from multiple subjects yielded better performance than training on individual subjects, thereby emphasizing the importance of large training datasets in this context.
Other studies explored architectures that combine spatial and temporal information. In [Salama et al. 2018], a 3D-CNN was proposed for emotion recognition, where the input consisted of a 3D volume: each frame represented a 2D spatial matrix of multi-channel EEG samples, and successive frames were stacked along the third dimension to capture temporal dynamics. Along similar lines, [X. Wei et al. 2018] defined inputs as 2D images of electrode signals and further proposed a 3D multichannel representation obtained by fusing images of spatially adjacent electrodes. This multichannel approach was shown to outperform both 2D CNNs and traditional signal-processing methods for seizure detection. Additionally, [Zhao et al. 2019] introduced a 3D representation that explicitly preserved electrode spatial positions. Instead of arranging raw signals as a [channels $\times$ time samples] matrix, they constructed a 2D matrix in which electrode values were placed at their corresponding spatial locations, with the remaining positions padded with zeros. This was then expanded into a 3D volume by stacking temporal segments along the third dimension. A multi-branch 3D CNN was subsequently employed to jointly extract spatial and temporal features for the classification of four motor imagery classes.

### 5.1.2 Classification based on two-dimensional EEG representations

Another line of research has focused on transforming EEG signals into two-dimensional representations, such as spectrograms or topographic maps, which are naturally suited for processing with CNNs.

**Spectral and temporal representations.** A common strategy is to transform EEG into time–frequency representations. For example, [Z. Wang et al. 2018] applied the short-time Fourier transform (STFT) to obtain time–frequency representations, rendered them as images, and used these as input to neural networks. Their results showed that CNNs performed better than LSTMs on these images, although both fell slightly below a

CSP+SVM baseline on the same dataset.
Similarly, [Vrbancic et al. 2018] classified spectrogram images of individual channels using a LeNet-based CNN, where classification results are obtained by a majority voting rule, in the context of detecting neural impairment disorders. [Vilamala et al. 2017] also relied on spectrograms, but employed multitaper spectral estimation to reduce both bias and variance in the spectral estimation, followed by a logarithmic scaling, segmentation into epochs, and mapping into RGB matrices using a colormap. This representation was then used for sleep stage classification via transfer learning.

**Classification based on combined spatio-spectral-temporal representations.** Beyond pure spectrograms, some works have integrated spectral information with the spatial arrangement of electrodes. [Bashivan et al. 2015] introduced a multispectral representation that preserved spectral and spatial information. They projected scalp electrodes into the plane using an azimuthal equidistant projection and interpolated power values over the $\theta^{EEG}$, $\alpha^{EEG}$, $\beta^{EEG}$ bands to form three-channel images. The temporal dimension was further incorporated by representing sequences of images from consecutive time windows, which were processed with a ConvNet followed by an LSTM to capture temporal dependencies; a similar approach was later reported in [Kuanar et al. 2018]. Extending this line of work, [Jiao et al. 2018] used the same multichannel formulation and developed single-channel representations. The single-channel representation was designed to preserve only temporal information by forming a 2D map with electrode names on one axis and power magnitudes in the three bands on the other. Interestingly, they found that shuffling electrode positions degraded classification performance in the multichannel representation but not in the single-channel case which highlights the effect of spatial location of the electrodes on the classification. Finally, they proposed a fusion strategy that combined both representations, showing that this approach achieved competitive results with fewer parameters compared to recurrent neural networks for mental workload classification.

### 5.1.3 Classification on connectivity-based EEG representations

Connectivity-based approaches explicitly model inter-electrode relationships and use them as CNN-compatible inputs. This category includes methods that classify connectivity matrices directly, as well as those that generate enriched representations from connectivity measures prior to classification.

**Directly classifying connectivity.** [Azad et al. 2025] proposed a Chimera Synchronization Matrix-based Network, where connectivity matrices were constructed by measuring the Euclidean distance between each electrode's amplitude and a reference. These

matrices were displayed as 2D images and directly classified by a CNN into three seizure-related states: pre-ictal, ictal, and post-ictal. Despite being trained on only 10 epochs, this approach achieved high accuracy.
Along similar lines, [Almohammadi et al. 2024] investigated two complementary strategies for motor imagery classification. The first was spatial: a binary adjacency matrix was built such that only spatially neighboring electrodes were connected (edge = 1), with all others set to zero. This adjacency matrix was then multiplied with the EEG signals and fed to a CNN. The second approach was functional: EEG signals were bandpass-filtered into different rhythms, PLV matrices were computed, and these were used as CNN inputs. Their experiments demonstrated that functional connectivity matrices capture long-range inter-electrode interactions more effectively than spatial adjacency, thereby improving classification performance.

**Connectivity-based representations prior to classification.** Moving beyond direct use of matrices, [Moon et al. 2020] reported being among the first to classify connectivity representations with deep learning. They computed connectivity using PLV and PC measures and proposed different strategies to order the electrodes before CNN classification. Two ordering schemes were tested: (i) starting from the left-frontal electrode and iteratively adding the nearest neighbor, and (ii) a hemispheric asymmetry strategy that restricted additions to the same hemisphere unless no electrodes remained. In addition, they introduced a data-driven ordering based on global and local connectivity interactions via one-dimensional scaling, which was shown to improve performance in emotional video recognition. Moreover, [Modak et al. 2025] combined a PC-based connectivity representation with a ResNet to distinguish patients with and without schizophrenia. Their method filtered EEG signals into five rhythms (delta, theta, alpha, beta, gamma), converted each time series into a multiplex weighted visibility graph (nodes are the time points, edges are weighted by inverse Euclidean distance), and evaluated mean local efficiency per electrode and the PC values between the electrodes were then derived from these measures. The resulting PC matrices for each rhythm were encoded as RGB images and classified with a CNN, showing superior performance compared to other connectivity-based deep learning approaches.
Finally, [Nogales et al. 2023] proposed generating 2D graph images based on connectivity metrics, including squared coherence, PC, directed transfer function, and corrected cross-correlation, with these measures leading to the best results among those tested. The resulting images had a size of $170 \times 1650$, where edge weights were represented by the thickness of connections. In a variant of their method, EEG signals were filtered into three rhythms (theta, alpha, and beta), with frequency-band-specific connections encoded by different colors. A CNN was then employed for image classification, showing promising results for two-class epilepsy seizure classification.

### 5.1.4 Discussion

Table 5.1 – Overview of deep learning approaches for EEG classification.

| Input | Model | Task | Reference |
|---|---|---|---|
| Raw EEG | LSTM | Seizure detection | [Hussein et al. 2019] |
| Raw EEG | CNN | Emotion recognition | [Yanagimoto et al. 2016] |
| 3D EEG volumes | 3D−CNN | Emotion recognition | [Salama et al. 2018] |
| 2D/3D signals images | CNN | Seizure detection | [X. Wei et al. 2018] |
| 3D electrode−position maps | 3D−CNN | Motor imagery | [Zhao et al. 2019] |
| Time−frequency images | CNN, LSTM | Seizure detection | [Z. Wang et al. 2018] |
| Spectrograms (per channel) | LeNet−CNN | Neural impairment | [Vrbancic et al. 2018] |
| Spectrograms (multitaper) | CNN | Sleep staging | [Vilamala et al. 2017] |
| Multispectral topomaps | CNN + LSTM | Temporal dynamics | [Bashivan et al. 2015; Kuanar et al. 2018] |
| Single/multi−channel maps | CNN + fusion | Mental workload | [Jiao et al. 2018] |
| Chimera sync. matrices | CNN | Seizure classification | [Azad et al. 2025] |
| PLV, PC | CNN | Emotion recognition | [Moon et al. 2020] |
| PC + multiplex graphs | ResNet | Schizophrenia | [Modak et al. 2025] |
| Spatial vs functional conn. | CNN | Motor imagery | [Almohammadi et al. 2024] |
| 2D graph images (coh., PC, DTF) | CNN | Epilepsy | [Nogales et al. 2023] |

In [Tab. 5.1], we provide an overview of the reviewed studies. While numerous approaches incorporate temporal, spectral, or spatial dimensions of EEG [Bashivan et al. 2015; Jiao et al. 2018; Kuanar et al. 2018; Vilamala et al. 2017; Vrbancic et al. 2018; Z. Wang et al. 2018], a common limitation arises in the treatment of connectivity. Specifically, some methods process connectivity matrices without considering the explicit spatial arrangement of electrodes [Azad et al. 2025; Modak et al. 2025; Moon et al. 2020], whereas others model electrode positions without explicitly including functional connectivity [Almohammadi et al. 2024; Z. Wang et al. 2018; Zhao et al. 2019]. Yet, both spatial organization and functional interactions are essential for EEG classification, and their joint integration remains largely unexplored despite its potential to improve performance. Among the reviewed studies, the work of [Nogales et al. 2023] is the most closely related to ours, as it also generates graph-based images from connectivity metrics prior to classification. The key distinction, however, lies in the representation: while their method produces large-scale graph images, our approach focuses on deriving compact, low-dimensional images that jointly encode spatial arrangement and connectivity information.

## 5.2 A graph image classification framework

Considering a set of multivariate EEG signals of $t$ seconds recorded by $n$ electrodes at a sampling frequency $F_s$, positioned on the scalp of a subject in known spatial positions

$\mathcal{C} \in \mathbb{R}^{n\times 3}$ , and stacked in a matrix $\mathbf{X} \in \mathbb{R}^{n\times(tF_s)}$, the proposed graph image representation learning framework is shown in [Fig. 5.1]. It admits the EEG signals as input data and predicts a class $\mathcal{Y} = [1, ..k]$ among a set of $k$ classes. It consists of three main steps: (1) functional graph construction, (2) functional graph to image (FG2IMG) representation and (3) classification based on a functional graph image. These steps are detailed below.

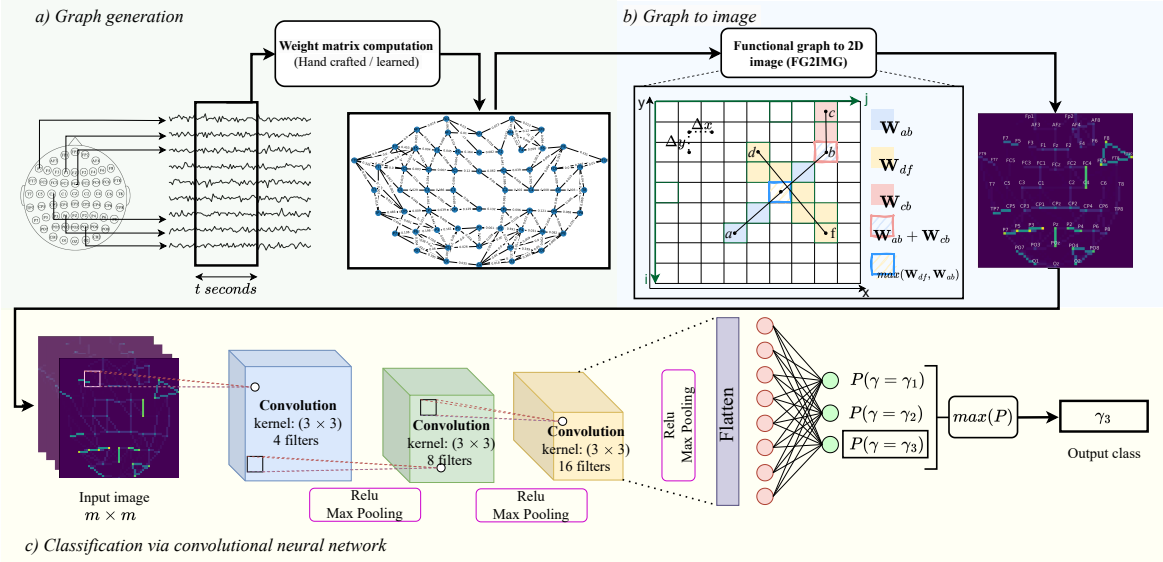



Figure 5.1 – Overview of the proposed classification framework with three main modules a) a graph generation module where a multivariate EEG signal is acquired, pre-processed, decomposed into windows from which functional graphs are learned, b) a Graph to Image module where the functional graph is represented by a 2D low resolution image, c) a classification module where a latent representation is learned using a convolutional neural network for a classification task.

### 5.2.1 Functional graph construction

Let us first recall that an undirected weighted graph is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$, where $\mathcal{V}$ is the set of vertices, $\mathcal{E}$ is the set of edges, and $\mathbf{W}$ is the weight matrix. In the context of EEG signals, the vertices correspond to the electrodes, and an edge between two electrodes $e$ and $f$ indicates the presence of an interaction between their recorded signals. The strength of the *dependency* between electrodes $e$ and $f$ is encoded in the entry $\mathbf{W}_{ef}$ of the weight matrix $\mathbf{W}$.

Several approaches have been proposed to quantify the interaction between electrodes. Many rely on pairwise similarity measures, often referred to as *handcrafted connectivity metrics* (see [Sec. 3.3.1]). These methods compute pairwise relationships by comparing signal characteristics (e.g., correlation, coherence, phase synchrony) but do not account for the influence of other signals in the network when estimating the dependency between

two electrodes.
To address this limitation, we propose to model the inter-electrode dependency using the precision matrix (inverse covariance), which captures *conditional dependencies* between signals. This formulation naturally leads to the problem of learning a *sparse* graph Laplacian $\mathbf{L}$, under structural constraints, as detailed in [Sec. 3.3.3]. Once the Laplacian $\mathbf{L}$ is learned, the weight matrix $\mathbf{W}$ is obtained by setting the diagonal entries to zero and taking the negative of the off-diagonal elements.

### 5.2.2 Functional graph to image representation

In order to optimize functional graph analysis and classification, our proposal is to exploit its weight matrix $\mathbf{W}$ to create a two-dimensional image $\mathbf{M}$ of size $m \times m$ pixels. This image provides a structured and compact representation of the graph while preserving the spatial relationships between electrodes. The proposed approach is summarized in [Fig. 5.2] and can be divided into two steps:

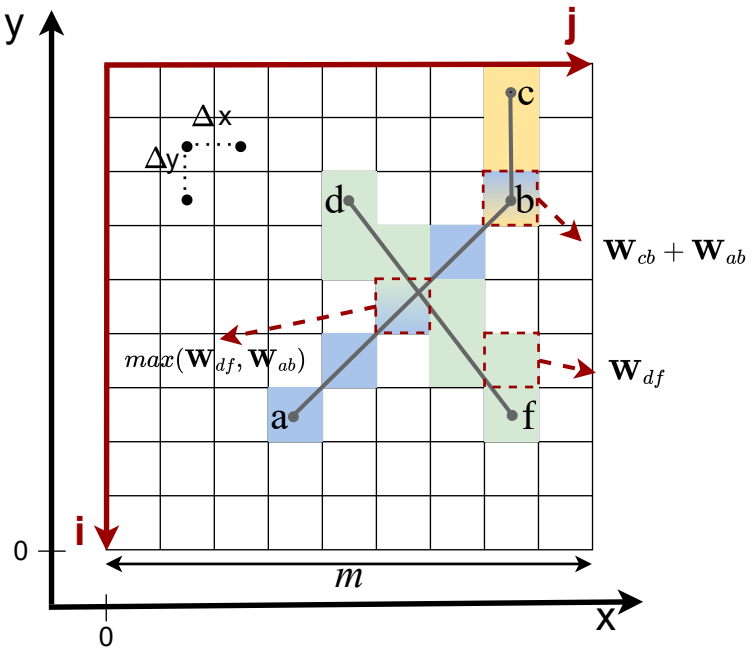



Figure 5.2 – Proposed image creation as a matrix representation of the electrodes spatial positions coupled with Bresenham line algorithm highlighting position, degree and connectivity of the graph.

#### 5.2.2.1 Finding the pixel indices corresponding to the electrode coordinates

Let $\mathcal{V}$ denote the set of electrodes, each associated with 2D Euclidean coordinates $\mathcal{C} = \{(\mathsf{x}_v, \mathsf{y}_v) \mid v \in \mathcal{V}\}$. To determine the pixel indices $\mathcal{P} = \{(\mathsf{i}_v, \mathsf{j}_v) \mid v \in \mathcal{V}\}$ of each electrode

in the image, we first define a regular grid that partitions the original 2D Euclidean space into $m$ cells along both the horizontal ($\mathsf{x}$) and vertical ($\mathsf{y}$) directions, spanning from the minimum $\mathsf{x}_v^{\min}$ to the maximum $\mathsf{x}_v^{\max}$ electrode coordinates.

The distance between of two consecutive grid elements in Euclidean space is computed as:

$$\Delta\mathsf{x} = \frac{\mathsf{x}_v^{\max} - \mathsf{x}_v^{\min}}{m}, \quad \Delta\mathsf{y} = \frac{\mathsf{y}_v^{\max} - \mathsf{y}_v^{\min}}{m}. \tag{5.1}$$

Once this grid is defined, each electrode is projected onto the matrix $\mathbf{M}$ by counting how many grid cells separate its position from the origin $(0, 0)$. The corresponding pixel indices can thus be computed as

$$\mathsf{i}_v = (m-1) - \frac{\mathsf{y}_v - \mathsf{y}_v^{\min}}{\Delta\mathsf{y}}, \qquad \mathsf{j}_v = \frac{\mathsf{x}_v - \mathsf{x}_v^{\min}}{\Delta\mathsf{x}}, \tag{5.2}$$

The term $(m-1)$ added to the vertical index compensates for the inverted $\mathsf{y}$-axis in matrix indexing and ensures non-negative coordinates.
This is illustrated in [Fig. 5.3], where $m = 61$ and the red points represent the electrodes in the real Euclidean space and the black markers separate the grid elements.

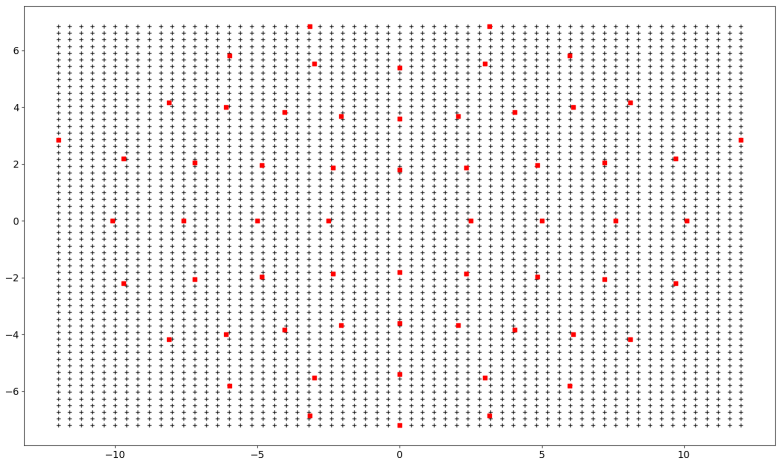



Figure 5.3 – Example of a projection of EEG electrode coordinates onto a $61 \times 61$ grid, where red squares indicate the electrode positions in the Euclidean space and the black markers denote the grid points.

#### 5.2.2.2 Defining the pixel values

Once the electrodes are projected from the 2D Euclidean space onto the matrix $\mathbf{M}$, the next step is to assign a value to each pixel in the image. We set the background pixels to zero and represent the connections between electrodes by drawing lines between the

pixels corresponding to nodes that are connected in the graph. These lines encode the edges of the graph within the image, where the pixel intensities along each line reflect the presence and strength of the corresponding connection.
To draw the connections between electrodes on the image, we use Bresenham's algorithm [Bresenham 1965], which relies on an integer incremental error scheme to efficiently perform line drawing in all octants. This method balances the positive and negative error between the $i$ and $j$ indices, allowing us to trace a discrete line that closely follows the ideal continuous segment. Let two pixels in the image be $p_e = \mathbf{M}_{(\mathsf{i}_e,\mathsf{j}_e)}$ and $p_f = \mathbf{M}_{(\mathsf{i}_f,\mathsf{j}_f)}$, corresponding to two different electrodes $e, f \in \mathcal{V}$. The set of pixel coordinates forming the discrete line $\mathcal{D}(p_e, p_f)$ that connects $p_e$ and $p_f$ is defined as

$$\mathcal{D}(p_e, p_f) = \Big\{ (\mathsf{i}_k, \mathsf{j}_k) \,\Big|\, \mathsf{i}_k \in [\mathsf{i}_e,\, \mathsf{i}_f],\ \mathsf{j}_k \in [\mathsf{j}_e,\, \mathsf{j}_f] \Big\},$$

where each new point $(\mathsf{i}_{k+1}, \mathsf{j}_{k+1})$ is obtained recursively from the previous one $(\mathsf{i}_k, \mathsf{j}_k)$. We define

$$\Delta\mathsf{i} = |\mathsf{i}_f - \mathsf{i}_e|, \quad \Delta\mathsf{j} = |\mathsf{j}_f - \mathsf{j}_e|, \quad s_\mathsf{i} = \operatorname{sgn}(\mathsf{i}_f - \mathsf{i}_e), \quad s_\mathsf{j} = \operatorname{sgn}(\mathsf{j}_f - \mathsf{j}_e),$$

and initialize the error variable as $\mathrm{err} = \Delta\mathsf{i} - \Delta j$. At each iteration, the doubled error is updated and the next pixel is chosen according to:

$$(\mathsf{i}_{k+1}, \mathsf{j}_{k+1}) = \begin{cases} (\mathsf{i}_k + s_\mathsf{i},\, \mathsf{j}_k), & \text{if } 2 \cdot \mathrm{err} > -\Delta\mathsf{j}, \quad \mathrm{err} \leftarrow \mathrm{err} - \Delta\mathsf{j}, \\ (\mathsf{i}_k,\, \mathsf{j}_k + s_\mathsf{j}), & \text{if } 2 \cdot \mathrm{err} < \Delta\mathsf{i}, \quad \mathrm{err} \leftarrow \mathrm{err} + \Delta\mathsf{i}. \end{cases} \tag{5.3}$$

This integer-only recurrence efficiently generates the sequence of discrete pixel coordinates that best approximates the straight line between $p_e$ and $p_f$, while supporting all octants without requiring floating-point operations.
In order to define the value of each pixel $p$, we associate it with the set of edge weights passing through it:

$$\mathcal{W}_p = \Big\{ \mathbf{W}_{ef} \,\Big|\, p \in \mathcal{D}(p_e, p_f),\ e, f \in \mathcal{V},\ e \neq f \Big\}. \tag{5.4}$$

For pixels corresponding to electrodes, the value is set to the node degree (i.e., the sum of the weights of all edges incident to that electrode), highlighting its importance in the network. For all other pixels (those crossed by one or more edges), the value is set to $\max(\mathcal{W}_p)$, representing the strongest connection passing through that location.

[Fig. 5.2] illustrates the process described in this section on an example of few electrodes. For instance, the electrodes $a$, $b$, $c$, $d$, and $f$ are first projected from their Euclidean positions $(\mathsf{x}, \mathsf{y})$ onto discrete pixel indices $(\mathsf{i}, \mathsf{j})$ in the image. Pixels are then selected whenever a connecting line between two electrodes passes through them (e.g., the lines linking

$a$ to $b$ and $d$ to $f$). The pixel values are updated according to the associated edge weights $\mathbf{W}$: for example, the pixel at $b$ accumulates the contributions of $\mathbf{W}_{ab}$ and $\mathbf{W}_{cb}$, while pixels along the line between $d$ and $f$ are set to $\max(\mathbf{W}_{df}, \mathbf{W}_{ab})$.

Examples of functional connectivity graphs are shown in [Fig. 5.4] and [Fig. 5.6], corresponding to selected signal windows from the COG-BCI and MI-HG datasets, respectively. The graphs are plotted using the electrodes' spatial positions projected into the azimuthal plane with the NetworkX Python library [Hagberg et al. 2008]. Their corresponding image representations generated by the FG2IMG framework are presented in [Fig. 5.5] and [Fig. 5.7]. In these images, the background is filled with zeros, and the color intensity encodes the strength of the dependencies between electrodes. In the left plots, zero-valued pixels are masked to better highlight weaker connections with lower edge weights.

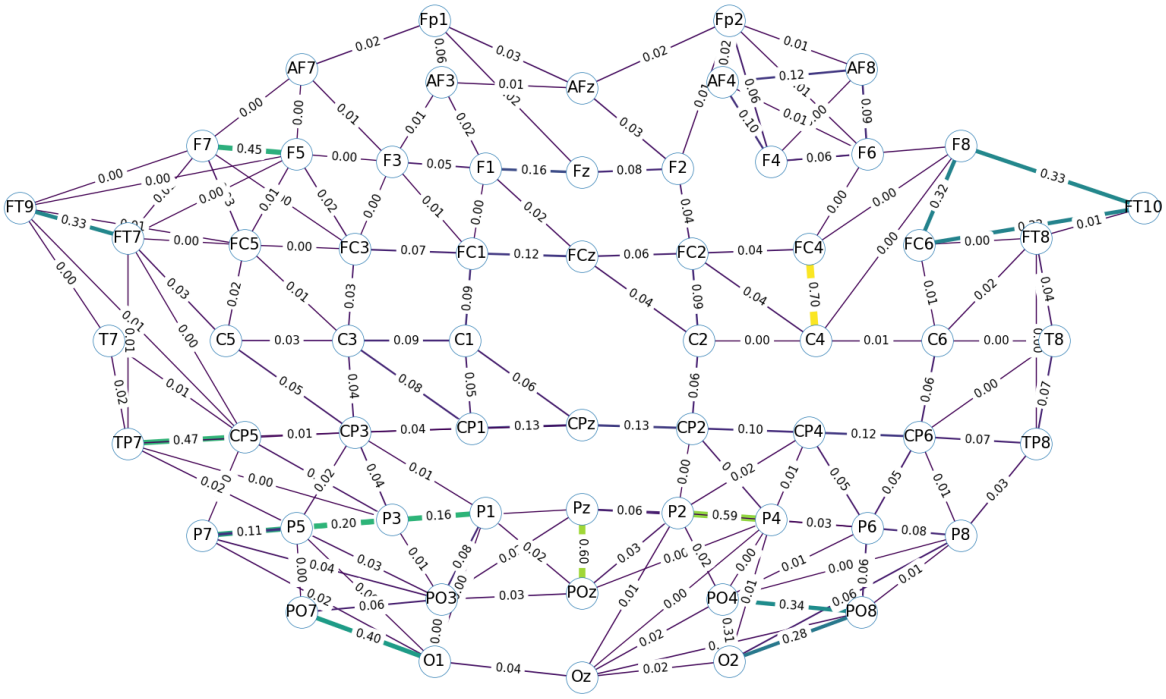



Figure 5.4 – Example of a functional connectivity graph obtained from a selected time window of Subject 7, session 2 (COG-BCI dataset), plotted using NetworkX Python library where edge weights are computed from the Laplacian; edge color and thickness reflect the learned connection strength.

### 5.2.3 Graph image classification

To infer a class from a 2D functional graph image, several deep learning architectures adapted to image classification can be applied. The retained architecture is a CNN, consisting of 3 convolutional blocks. Each one consisting of a convolutional layer, ReLU as an activation function and a max pooling operation. The convolutional layers have 4,

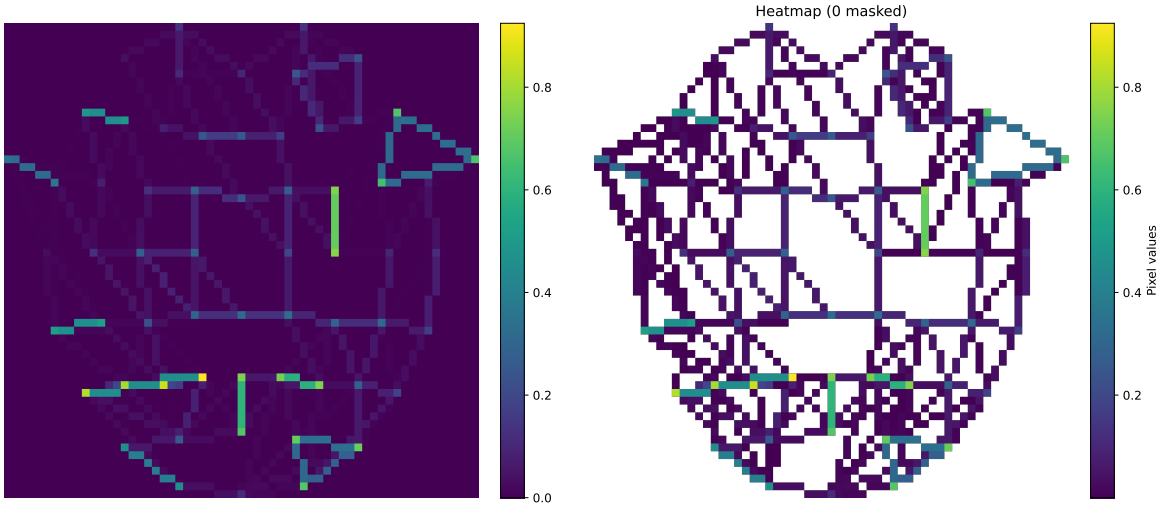



Figure 5.5 – FG2IMG representation of the functional connectivity graph for a selected time window of Subject 7 Session 2 (COG-BCI dataset). Left: full image including the zero-valued background; right: the background is masked to highlight edge details.

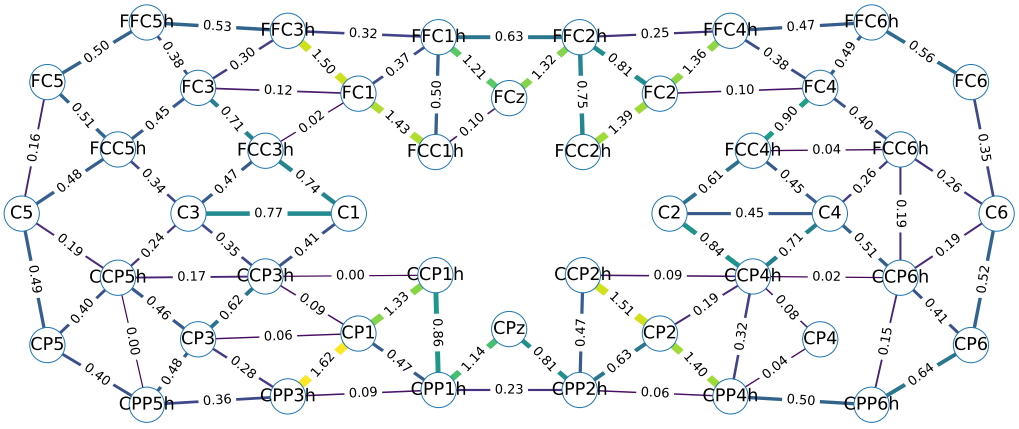



Figure 5.6 – Example of a functional connectivity graph obtained from a selected time window of Subject 5 in the MI-HG dataset, plotted using NetworkX Python library where edge weights are computed from the Laplacian; edge color and thickness reflect the learned connection strength.

8 and 16 filters respectively, with a kernel size of $3 \times 3$. The output is flattened to obtain the latent representation before being fed to a dense hidden layer of 8 neurons with a ReLU activation function followed by the output layer and a softmax activation function as shown in [Fig.5.8]. The loss function used is a categorical cross entropy between the predicted (the one with the highest probability) and the ground truth classes.

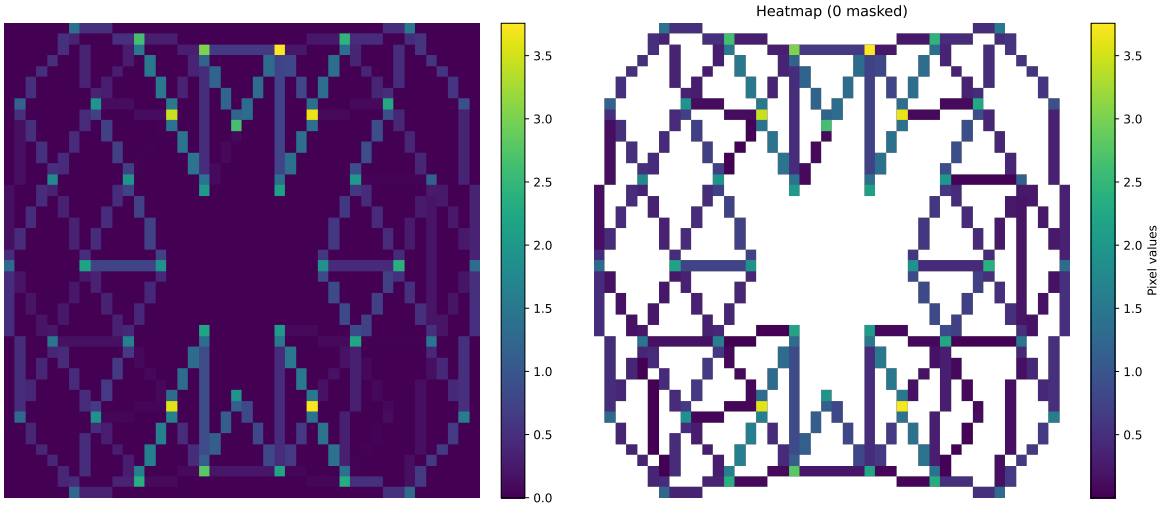



Figure 5.7 – FG2IMG representation of the functional connectivity graph for a selected time window of Subject 5 (MI-HG dataset). Left: full image including the zero-valued background; right: the background is masked to highlight edge details.

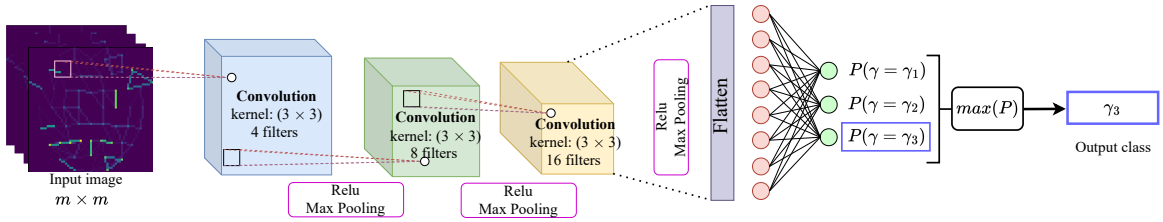



Figure 5.8 – Convolutional neural network architecture for image-based EEG graph classification. The input image is processed through three convolutional layers with ReLU activation and max pooling, followed by fully connected layers that output the predicted class.

## 5.3 Experimental setup

**COG-BCI dataset [Hinss et al. 2023].** The proposed framework is first evaluated on the three-class MW classification task using the COG-BCI dataset [Hinss et al. 2023], described in [Sec. 4.1]. The multivariate signals are segmented into overlapping windows of 2 seconds with a 1-second overlap. For each window, a graph is constructed as described in [Sec. 3.3.3] and subsequently transformed into an image as described in [Sec. 5.2.2]. To avoid overlap between training and test sets within each subject, windows are grouped by temporal order, and the last window of each group is excluded from the experiments. Classification performance is assessed through 5-fold cross-validation. In addition, different proportions of window groups are used for training and testing to analyze the impact of training set size. When using CNNs, 10% of the training groups are reserved for validation. For fair comparison, the random seed is fixed and identical folds are employed across all strategies.

**HG-MI dataset [Schirrmeister et al. 2017].** The second dataset is the HG-MI dataset [Schirrmeister et al. 2017], which was downloaded and preprocessed using the MOABB Python library. It includes four motor imagery classes recorded from 14 participants. In this case, no overlapping windows were applied since the trials are not continuous. Evaluation was performed using 5-fold cross-validation, and the results were directly compared with those reported in MOABB [Chevallier et al. 2024]. No re-implementation was carried out, and although the random seed used in MOABB was not available, a fixed seed was used for the cross-validation in our experiments.

**Implementation details.** For each multivariate EEG window, a graph Laplacian $\mathbf{L}$ is learned as described in [Sec. 5.2.2], then transformed into a 2D image representation based on the standard $(\mathsf{x}, \mathsf{y})$ electrode coordinates corresponding to the azimuthal projection of the 3D sensor layout. Electrode positions in 2D are obtained from the EEGRAPH library [Maitin et al. 2023], which provides a dictionary of standardized electrode configurations. For the COG-BCI dataset, images of size $61 \times 61$ are generated, yielding on average 297 images per class, organized into 27 groups of equal size. For the HG-MI dataset, images of size $44 \times 44$ are created, consistent with the number of electrodes in both cases, resulting in approximately 240 images per class, matching the available trials.
Regarding the CNN, the same architecture and training setup are employed across both datasets. The network is optimized using the categorical cross-entropy loss function and the Adam optimizer, with a "He normal" kernel initializer, a batch size of 1 and a learning rate of $10^{-4}$. A learning rate scheduler with a reduction factor of 0.5, a minimum learning rate of $10^{-6}$, and patience of 20 epochs (based on validation loss) is applied. Training is run for a maximum of 600 epochs with early stopping, triggered if the validation loss does not improve for 100 consecutive epochs, in order to prevent overfitting.

## 5.4 Experimental results and analysis

In this section, we evaluate the performance of the proposed approach. We investigate the impact of selecting the graph structure based on Pearson correlation versus partial correlation. For the learned Laplacian, we analyze the effect of imposing a predefined graph structure compared to learning it by promoting sparsity without structural constraints. Finally, we compare all configurations against the baseline methods.

### 5.4.1 Impact of the of the structural constraint on the Laplacian

The graph learning problem described in [Sec. 3.3.3] includes a structural constraint in the Laplacian estimation process. We first investigate the impact of this constraint on the proposed pipeline. Two alternative definitions of the connectivity matrix $\mathbf{A}$ are con-

sidered. In the first case, only sparsity is favored by constraining $\mathbf{A}$ to correspond to a fully connected adjacency matrix, where all off-diagonal entries are set to 1. In the second case, $\mathbf{A}$ is defined as a 3-NN graph, where each electrode is connected to its three nearest neighbors according to their 3D spatial coordinates.

**On the COG-BCI dataset** [Tab. 5.2], both approaches achieve comparable accuracies, with fully-connected showing a slight overall advantage.

**On the HG-MI dataset** [Tab. 5.3], the fully-connected configuration systematically outperforms 3-NN across all train/test splits, with gains of about 0.01 and 0.02 on average. These results suggest that leveraging full connectivity is more beneficial than restricting the graph to local neighborhoods, particularly in the MI classification task. On the other hand the close performance can suggest that imposing structural constraints reduces complexity and computation time by limiting the number of matrix entries to be learned, while still preserving the most relevant connections.

### 5.4.2 Pearson correlation vs partial correlation based connectivity

Next, to show the interest of learning a sparse precision matrix (*i.e.* indirectly a sparse partial correlation), a comparison is made to another strategy where the graph learning module is replaced by a thresholded Pearson correlation matrix.

$$\mathbf{W}_{ef} = \frac{\mathbf{S}_{ef}}{\sqrt{\mathbf{S}_{ee}\mathbf{S}_{ff}}} \tag{5.5}$$

$\mathbf{S}_{ij}$ being the covariance between two electrode signals e and f. The latter is one of the widely adopted *dependency* measure used in the context of EEG and brain connectivity. More precisely, the edge connecting two electrode signals is assigned a weight equal to their Pearson correlation coefficient. The graph is then usually sparsified by removing the edges with weights less then a threshold $T$. For this experiment, the threshold is empirically determined to 0.3 yielding the best performance.
When comparing Pearson correlation with the proposed FG2IMG representations, the advantage of functional connectivity becomes evident:

**On the COG-BCI dataset** [Tab. 5.2], Pearson correlation achieves a mean accuracy between 0.58 and 0.72 for S1, and between 0.57 and 0.73 for S2. In contrast, FG2IMG substantially improves performance, reaching 0.79 to 0.88 with the fully-connected variant and 0.78 to 0.88 with 3-NN.

**On HG-MI dataset** [Tab. 5.3] a similar trend is observed, where Pearson correlation remains limited to an average of 0.44 to 0.55 in terms of accuracy across splits, while FG2IMG consistently achieves a higher performance, ranging from 0.61 to 0.70.

These results demonstrate that learning a sparse partial correlation provides a more informative representation than pairwise Pearson correlation, capturing richer dependencies in EEG signals. In fact, the Laplacian-based approach estimates the connectivity between two electrodes while accounting for the influence of all other signals in the network. This is in contrast to hand-crafted measures, which compute pairwise connections independently and disregard the global network structure. As a consequence, such pairwise measures are more prone to amplifying the volume conduction problem.

Table 5.2 – COG-BCI dataset [Hinss et al. 2023] classification accuracy (mean ± std; 5-fold CV) across 29 participants for two sessions. Best result per row is **bold**.

| Session | Train/Test (%) | Pearson corr. | FG2IMG | |
|---|---|---|---|---|
| | | | Fully-connected | 3-NN |
| S1 | 20/80 | 0.58 ± 0.07 | **0.79 ± 0.12** | 0.78 ± 0.13 |
| | 40/60 | 0.65 ± 0.08 | **0.84 ± 0.10** | 0.83 ± 0.12 |
| | 60/40 | 0.70 ± 0.10 | **0.87 ± 0.10** | 0.86 ± 0.11 |
| | 80/20 | 0.72 ± 0.10 | **0.88 ± 0.10** | **0.88 ± 0.11** |
| S2 | 20/80 | 0.57 ± 0.08 | **0.78 ± 0.14** | **0.78 ± 0.13** |
| | 40/60 | 0.67 ± 0.10 | **0.84 ± 0.13** | 0.82 ± 0.13 |
| | 60/40 | 0.70 ± 0.10 | **0.88 ± 0.10** | 0.86 ± 0.11 |
| | 80/20 | 0.73 ± 0.10 | **0.90 ± 0.08** | 0.88 ± 0.09 |

Table 5.3 – HG-MI dataset [Schirrmeister et al. 2017] classification accuracy (mean ± std; 5-fold CV) across 14 participants for different approaches used with FG2IMG and classical baseline results. Only results on 80/20 % splits for baseline approaches are available. Best result per row is **bold**.

| Train/Test (%) | CSP | MDM | Pearson corr. | FG2IMG | |
|---|---|---|---|---|---|
| | | | | Fully-connected | 3-NN |
| 20/80 | - | - | 0.44 ± 0.08 | **0.61 ± 0.10** | 0.60 ± 0.10 |
| 40/60 | - | - | 0.51 ± 0.09 | **0.67 ± 0.09** | 0.65 ± 0.10 |
| 60/40 | - | - | 0.55 ± 0.09 | **0.69 ± 0.09** | 0.67 ± 0.09 |
| 80/20 | **0.73 ± 0.10** | 0.52 ± 0.10 | 0.55 ± 0.10 | 0.70 ± 0.09 | 0.69 ± 0.10 |

### 5.4.3 Comparison against baseline classification pipelines

Finally, the proposed framework, is compared with two classification pipelines commonly adopted in the literature: CSP-OVR [W. Wu et al. 2006] and R-MDM [Barachant

et al. 2012], as described in [Section 2.4.2.4].

**For the COG-BCI dataset:** [Tab. 5.4] reports the results obtained using our re-implementation. In this setup, the covariance matrix of a subset of 10 electrodes was computed in the $\theta^{EEG}$ ([4–8] Hz) and $\alpha^{EEG}$ ([8–12] Hz) frequency bands, and then fed into a R-MDM classifier. Evaluation was conducted with 5-fold cross-validation. The covariance matrices were computed for overlapping windows of 2 seconds in each of the two frequency bands, as well as from the broadband signal without frequency decomposition. For a more comprehensive comparison, the same pipeline was also applied to the covariance matrices computed from all available electrodes.
The results in [Tab. 5.4] highlights the superiority of the proposed FG2IMG framework [Tab. 5.2]. CSP-OVR achieves competitive performance, with mean accuracies in the range of 0.71 to 0.80, yet it remains consistently below FG2IMG, which reaches up to 0.90. The R-MDM pipeline, in contrast, shows notably lower performance: on average, accuracies remain around 0.50 and 0.62 when restricted to 10 electrodes in the theta or alpha bands, and only rise to values between 0.61 and 0.68 when considering all frequencies or all electrodes. Overall, these results confirm that FG2IMG not only outperforms pairwise correlation-based features, but also surpasses pipelines such as CSP-OVR and R-MDM, especially when evaluated on this dataset. The significant difference between the conventional pipelines and our framework highlights the advantage of the proposed FG2IMG method over the other approaches for mental workload classification.

**For the HG-MI dataset**, literature baselines have only reported results for the 80/20 split. In this setting, the R-MDM classifier achieves $0.52 \pm 0.10$ [Barachant et al. 2012], whereas CSP combined with LDA reaches $0.73 \pm 0.10$, see [Tab. 5.3]. In comparison, our FG2IMG framework achieves $0.70 \pm 0.09$ (fully-connected) and $0.69 \pm 0.10$ (3-NN), which represents a clear improvement over R-MDM and Pearson correlation ($0.55 \pm 0.10$), but remains slightly below CSP-based approaches.

**Comparing session 1 to session 2 in COG-BCI dataset:** Results shown in [Tab. 5.2] and [Tab. 5.4], show that for all approaches evaluated on the COG-BCI dataset, the results obtained in S2 are consistently higher than those in S1. This can be explained by the improved signal quality in the second session, as participants were already familiar with the setup and tasks after the first session this aligns with what was reported by [Roy et al. 2022].

**Comparing different train test proportions:** Finally, we note that using a 60/40 split for training and testing achieves performance very close to that of the 80/20 split, with differences not exceeding 0.02 across all approaches and datasets. This suggests

Table 5.4 – COG-BCI [Hinss et al. 2023] dataset classification accuracy (mean ± std; 5-fold CV) across 29 participants for two sessions, using Riemannian-MDM variants, CSP-OVR [W. Wu et al. 2006], and the proposed FG2IMG approach under different train/test splits. Best result per row is **bold**.

| Sess. | Train(%) | Riemannian-MDM | | | | CSP-OVR | FG2IMG |
|---|---|---|---|---|---|---|---|
| | | $\alpha^{EEG}$-10 elec | $\theta^{EEG}$-10 elec | All freq-10 elec | All freq-all elec | | |
| S1 | 20 | 0.51 ± 0.09 | 0.50 ± 0.06 | 0.61 ± 0.06 | 0.58 ± 0.06 | 0.71 ± 0.12 | **0.79 ± 0.13** |
| | 40 | 0.54 ± 0.09 | 0.53 ± 0.06 | 0.62 ± 0.07 | 0.59 ± 0.06 | 0.73 ± 0.13 | **0.84 ± 0.10** |
| | 60 | 0.55 ± 0.09 | 0.54 ± 0.06 | 0.63 ± 0.06 | 0.60 ± 0.06 | 0.73 ± 0.14 | **0.87 ± 0.10** |
| | 80 | 0.57 ± 0.09 | 0.55 ± 0.05 | 0.65 ± 0.06 | 0.61 ± 0.07 | 0.75 ± 0.14 | **0.88 ± 0.10** |
| S2 | 20 | 0.57 ± 0.13 | 0.60 ± 0.12 | 0.65 ± 0.12 | 0.59 ± 0.08 | 0.77 ± 0.12 | **0.78 ± 0.14** |
| | 40 | 0.60 ± 0.13 | 0.60 ± 0.12 | 0.66 ± 0.12 | 0.60 ± 0.08 | 0.79 ± 0.12 | **0.84 ± 0.13** |
| | 60 | 0.61 ± 0.13 | 0.60 ± 0.12 | 0.67 ± 0.11 | 0.61 ± 0.08 | 0.80 ± 0.13 | **0.88 ± 0.10** |
| | 80 | 0.62 ± 0.13 | 0.62 ± 0.11 | 0.68 ± 0.11 | 0.62 ± 0.09 | 0.80 ± 0.14 | **0.90 ± 0.08** |

that competitive results can be obtained even with less training data. For FG2IMG in particular, reliable performance is also reached with a 40/60 split, further highlighting the robustness of the proposed graph image representation.

### 5.4.4 Interpretability and explainability

An additional advantage of this method lies in its explainability: the generated visualizations offer interpretable insights into the role of electrode connectivity, which may be of practical value for clinical experts. To illustrate this, we randomly selected four subjects from different sessions of the COG-BCI dataset. The Gradient-Weighted Class Activation Mapping (Grad-CAM) [Selvaraju et al. 2017] visualizations in [Fig. 5.9] show one representative window graph for each mental-workload signal per subject and highlight how the CNN attends to different electrode regions across task difficulty levels. The Grad-CAM heatmaps [Selvaraju et al. 2017] exhibit relatively diffuse activations overall, but the strongest responses consistently align with densely connected electrode regions. Because the electrode connectivity patterns for a given subject remain largely stable across difficulty levels, we infer that the strength of connectivity plays a key role in distinguishing mental-workload classes. Furthermore, the observed activations, as well as the underlying graph structures appear to be subject-dependent, as the most salient connections do not occur between exactly the same electrodes across different individuals.

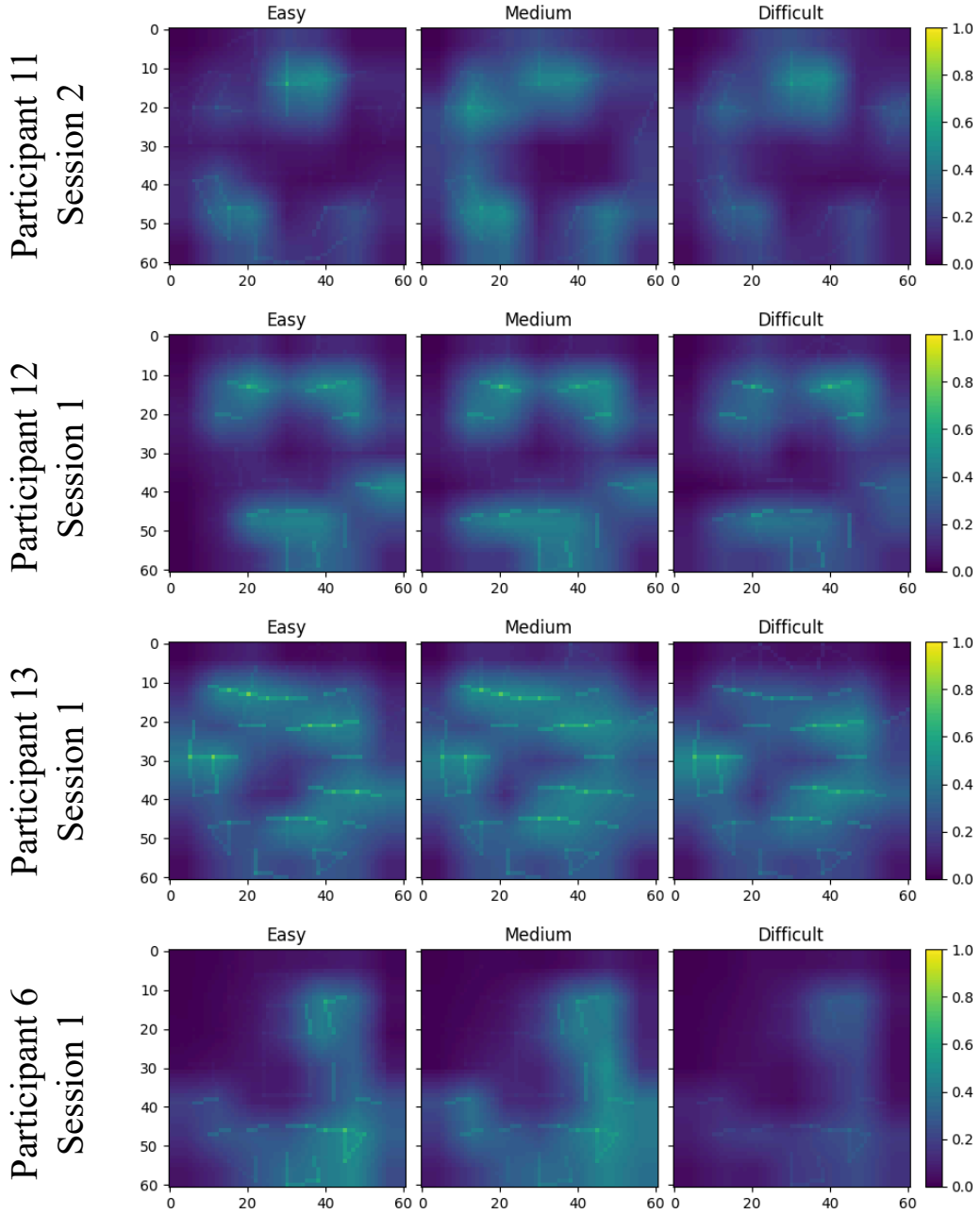



Figure 5.9 – Visualization of Grad-CAM activation maps across task difficulty levels (easy, medium, and difficult) for subjects chosen randomly from COG-BCI dataset.

## 5.5 Conclusion

In this chapter, we proposed a novel framework for the classification of EEG functional connectivity. Our approach relies on learning a generalized graph Laplacian under structural and sparsity constraints and on introducing a new functional graph to image (FG2IMG) representation which preserves the spatial arrangement of the electrodes, their importance in the graph and the connectivity patterns. These images are subsequently classified using convolutional neural networks.

The proposed framework demonstrated promising performance compared to classical pipelines such as Common Spatial Patterns and Riemannian MDM in the context of mental workload classification. On the motor imagery dataset, it outperformed R-MDM and achieved results comparable to CSP, with CSP remaining slightly superior. Furthermore, we showed that learning sparse partial correlations provides a more informative representation and leads to better classification performance than relying on pairwise Pearson-correlation connectivity.

An important advantage of the framework lies in its interpretability: it enables visualization and analysis of both the model's behavior and the underlying connectivity patterns of each subject. Our results further highlight the role of functional connectivity in class discrimination and confirm the superiority of Laplacian-based connectivity learning over simple pairwise correlation measures.

# Chapter 6

# Graph Force Learning for EEG Classification


6.1 Force-related work . . . 78
6.1.1 Interaction forces as a foundation for graph layouts . . . 78
6.1.2 Force-based models in graph clustering and embedding . . . 82
6.1.3 Discussion . . . 84
6.2 Hybrid physics data driven graph force learning framework . . . 84
6.2.1 Data-driven graph learning . . . 85
6.2.2 Physics-inspired graph force model . . . 85
6.2.2.1 2-D Interaction Forces . . . 86
6.2.2.2 3-D Interaction forces . . . 89
6.2.3 Classification based on force-related features . . . 92
6.3 Experimental results and analysis . . . 93
6.3.1 Experimental setting . . . 93
6.3.2 Classification results and analysis . . . 94
6.3.2.1 Impact of the choice of the node force-related features . . . 94
6.3.2.2 Impact of the graph structure . . . 95
6.3.2.3 Impact of the choice of classifier . . . 95
6.3.2.4 Comparison against baseline classification pipelines . . . 96
6.3.2.5 Relevance of graph-force features . . . 97
6.3.2.6 Effect of force parameters on the classification . . . 100
6.3.2.7 Visualization and explainability . . . 101
6.4 Conclusion . . . 104

While the previous chapter focused on FG2IMG, transforming connectivity graphs into 2D images to be classified with convolutional neural networks, this chapter explores a complementary perspective. Instead of converting graphs into images, we propose to build vector embeddings that directly encode the spatial arrangement and functional connectivity of EEG electrodes.
This shift from image space to vector space opens new opportunities: such embeddings integrate naturally with a wide range of machine learning models beyond CNNs while still preserving the underlying structure of the graph.
Our formulation draws inspiration from physics-based interaction models, particularly Hooke's law for springs and Coulomb's law for charged particles. Originally developed for graph visualization, force-directed models have been adapted more recently to graph embedding and clustering. Here, we extend these ideas to EEG analysis: by modeling attractive and repulsive forces between nodes, we derive force-driven feature vectors that can be directly used for classification, offering a physically grounded way to represent EEG connectivity.

## 6.1 Force-related work

We begin with a comprehensive review of force-directed approaches, tracing their evolution from their original use in graph layout visualization, where attractive and repulsive forces are employed to produce intuitive two-dimensional representations of complex networks, to their more recent adaptation for graph embedding and clustering, where similar physical models are leveraged to learn low-dimensional representations that preserve structural relationships.

### 6.1.1 Interaction forces as a foundation for graph layouts

The increasing use of graphs to represent complex data has motivated the development of methods for their visualization. A graph layout algorithm aims to assign spatial positions to vertices in such a way that the underlying structure becomes easier to interpret. Many approaches rely on the idea of forces acting on vertices: by letting attraction and repulsion drive the system towards an equilibrium, the resulting configuration often yields a clear and readable drawing even for large graphs.

**Tutte's barycentric algorithm.** A first attempt in this direction is the barycentric method proposed by [Tutte 1963]. In this approach, each vertex $e$ is attracted towards the average position of its neighbors. The force acting on $e$ can be written as $\mathbf{F}_e^a = \sum_{f \in \mathcal{N}_1(e)} \Big( \mathbf{c}_e - \mathbf{c}_f \Big)$, where $\mathbf{c}_e = (\mathsf{x}_e, \mathsf{y}_e)$ denotes the spatial coordinates of vertex $e$. In

practice, the layout is initialized by fixing a set of boundary vertices on the corners of a convex polygon. Every remaining vertex is then placed at the barycenter of its neighborhood, with coordinates $\mathsf{x}_f = \frac{1}{\deg(f)} \sum_{e \in \mathcal{N}_1(f)} \mathsf{x}_e$; $\mathsf{y}_f = \frac{1}{\deg(f)} \sum_{e \in \mathcal{N}_1(f)} \mathsf{y}_e$. Thus, $\mathbf{c}_f$ coincides with the mean position of its adjacent vertices.

**Eades' force-directed method** While the barycentric method relies only on local averaging, later approaches introduced more realistic physical analogies to better balance the layout. In particular, many models were inspired by mechanical systems of springs and electrical forces, where vertices are treated as physical objects interacting according to simple physical laws. A key principle often invoked is Hooke's law [Hook 1679h], which states that *the restoring force of a spring is proportional to its displacement from the equilibrium position*,

$$F = -k_s \, \bar{\mathsf{x}}, \tag{6.1}$$

where $k_s$ is the spring constant and $\bar{x}$ is the displacement.

One of the first algorithms in this family was introduced by [Eades 1984]. In this model, vertices are represented as steel rings and edges as springs, forming a mechanical system [Fig. 6.1]. The vertices are initially placed at random positions, and the system is then allowed to evolve under the action of the forces until it reaches equilibrium. At convergence, the edges tend to have nearly uniform length, which produces a layout with a high degree of symmetry. To achieve this, two types of forces are considered: an attractive force between adjacent vertices and a repulsive force between all pairs of non-adjacent vertices. They are defined as:

$$\|\mathbf{F}_a^{ef}\| = k_a \log\left(\frac{d_{ef}}{d_0}\right), \qquad \|\mathbf{F}_r^{ef}\| = \frac{k_r}{\sqrt{d_{ef}}}, \tag{6.2}$$

where $d_{ef}$ is the Euclidean distance between vertices $e$ and $f$, $d_0$ is the ideal edge length, and $k_a, k_r$ are constants that control the strength of the interactions. The logarithmic formulation of the attractive force ensures that, unlike the classical spring, the force does not become excessively large when vertices are far apart. This makes the algorithm more stable for sparse graphs. Originally, this method was applied to networks of moderate size, typically with up to 30 vertices.

**Fruchterman–Reingold force-directed layout** Although [Eades 1984] was motivated by the spring analogy, its formulation of the forces does not strictly adhere to Hooke's law. In light of this, the work of [Fruchterman et al. 1991] introduced a refined version of the model, where both attraction and repulsion forces are explicitly defined in accordance with the spring layout, while also taking into account the size of the drawing

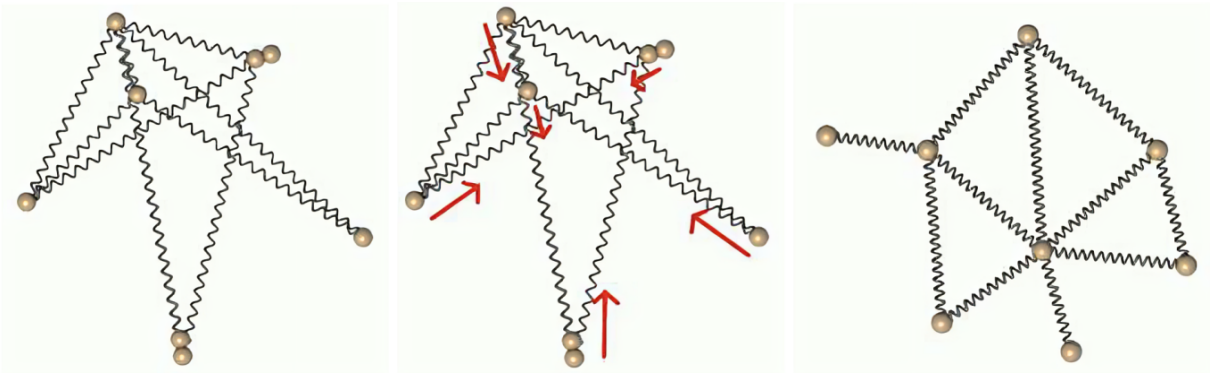


Figure 6.1 – Spring–mass system analogy. Starting from a random configuration, particles linked by springs are iteratively displaced by spring forces until the system converges to a stable, low-energy equilibrium [Kobourov 2012].

area. The forces between two vertices $e$ and $f$ separated by a Euclidean distance $d_{ef}$ are expressed as:

$$\|\mathbf{F}_a^{ef}\| = \frac{d_{ef}^2}{k_a \frac{\sqrt{area}}{n}}, \qquad \|\mathbf{F}_r^{ef}\| = -\frac{\left(k_r \frac{\sqrt{area}}{n}\right)^2}{d_{ef}}, \tag{6.3}$$

$k_a$ and $k_r$ are positive constants, $n$ the number of vertices and $area$ denotes the size of the frame in which the graph is drawn. The factors $k_a \times \frac{\sqrt{area}}{n}$, and $k_r \times \frac{\sqrt{area}}{n}$, serve as scaling factors.

**Kamada–Kawai force-directed layout** The formulation of [ Fruchterman et al. 1991] improved the stability of spring-based layouts by scaling the forces with the drawing area and the number of vertices, but still relied on Euclidean distances between vertices at each iteration to define the interactions. This can sometimes lead to distortions, when the current geometric distances in the layout do not always reflect the true graph-theoretic relationships between vertices. To overcome this limitation, the algorithm of [Kamada et al. 1989] adopted a different strategy: instead of relying on the evolving Euclidean distances, it uses the shortest path distances in the graph to determine the ideal lengths of the springs. In this model, the desired distance between two vertices $e$ and $f$ is set to $l_{e,f} = L \cdot d(e, f)$ where $d(e, f)$ is the graph-theoretic shortest path distance, and $L = L_0 / \max_{e<f} d(e, f)$ rescales this value with respect to the drawing area of side length $L_0$. The spring constant is defined as $k_{e,f} = \frac{K}{d(e,f)^2}$ so that springs are stronger between vertices that are close in the graph and weaker between distant ones where $K$ is a constant. Instead of defining attractive and repulsive forces separately, the method minimizes the following global energy function:

$$E = \sum_{e=1}^{n-1} \sum_{f=e+1}^{n} \tfrac{1}{2} k_{ef} \left(|\mathbf{c}_e - \mathbf{c}_f| - l_{e,f}\right)^2. \tag{6.4}$$

which drives the layout toward a configuration where Euclidean distances reflect the underlying graph distances as faithfully as possible.

**Noack's LinLog layout** While the [Kamada et al. 1989] model focuses on preserving graph-theoretic distances through a global energy minimization, later work sought to adapt force-directed principles to emphasize community structures rather than uniform edge lengths. In particular, the LinLog energy model introduced by [Noack 2007] was designed with graph clustering in mind. The central idea is to balance attraction and repulsion in such a way that densely connected groups of vertices are drawn close together, while sparsely connected groups are pushed apart. The attractive energy is defined between **connected vertices** and increases with their Euclidean distance $d_{ef}$:

$$\|\mathbf{F}_a^{ef}\| = d_{ef}, \tag{6.5}$$

For the repulsive forces, two variants are developed, the node repulsion and edge repulsion: while the node repulsion acts between **all pairs of vertices**, and is logarithmic in their Euclidean distance:

$$\|\mathbf{F}_r^{ef}\| = \ln(d_{ef}), \tag{6.6}$$

the edge-repulsion, further accounts for the degrees of the vertices. In this case, the repulsion between two vertices $e$ and $f$ is weighted by the product of their degrees:

$$\|\mathbf{F}_r^{ef}\| = \mathbf{deg}_e\mathbf{deg}_f \ln(d_{ef}). \tag{6.7}$$

These modifications make LinLog particularly effective at producing layouts where clusters of highly interconnected vertices are clearly visible, since intra-cluster distances are shortened by attraction while inter-cluster distances are amplified by the repulsion.

**Jacomy's ForceAtlas2 layout** Building on the principles of the LinLog model, the work of [Jacomy et al. 2014] introduced *ForceAtlas2*, the force-directed layout implemented in the Gephi software [Bastian et al. 2009]. This algorithm consolidates several ideas from earlier methods and refines both attraction and repulsion so as to better handle large, real-world networks.

For the attractive force, ForceAtlas2 retains a distance-proportional formulation while introducing additional options. In particular, it can operate in a LinLog mode [Noack 2007], incorporate scaling factors, and account for edge weights. The attraction between two vertices $e$ and $f$ is expressed as:

$$\|\mathbf{F}_a^{ef}\| \;=\; \mathbf{W}_{ef}^{\delta}\, d_{ef}, \qquad \delta \in [0,1], \tag{6.8}$$

where $\mathbf{W}_{ef}$ is the edge weight. The parameter $\delta$ controls the influence of weights. $\delta = 0$ ignores them, while $\delta = 1$ makes the attraction fully proportional to $\mathbf{W}_{ef}$.

The repulsive force is also revised in order to balance the influence of nodes with different degrees. Specifically, it reduces the repulsion between a high-degree node and a low-degree node, bringing peripheral vertices closer to hubs, while ensuring strong separation between highly connected vertices:

$$\|\mathbf{F}_r^{ef}\| \;=\; k_r \,\frac{\Big(\mathbf{deg}_e + 1\Big)\Big(\mathbf{deg}_f + 1\Big)}{d_{ef}}. \tag{6.9}$$

The addition of 1 ensures that even isolated nodes exert repulsive forces. This formulation prevents peripheral nodes from being pushed too far away, while still maintaining a clear separation of densely connected regions. ForceAtlas2 thus combines scalability, weight sensitivity, and clustering emphasis, making it a widely used choice for the visualization of complex networks. We draw inspiration from this last force representation by adding directional information between nodes and introducing hyper-parameters that modulate subject-specific force patterns. For further details on graph visualization, the reader is referred to [Cheong et al. 2020; Kobourov 2012].

### 6.1.2 Force-based models in graph clustering and embedding

Beyond graph visualization, physics-inspired approaches have more recently been applied to clustering and graph embedding. In this line, [K. Sun, J. Liu, et al. 2020] introduced the GForce model, inspired by spring systems simulating molecular interactions. In this framework, nodes exert repulsive forces on each other, while attractive forces act between neighboring nodes, with the objective of preserving the structural properties of the graph in the embedding space. The attractive force is defined as the weighted distance between the feature vectors of the nodes, whereas the repulsive force is inversely proportional to the weighted Euclidean distance.

$$\mathbf{F}_a^{ef} = -\eta_a \mathbf{W}_{ef}(\bar{\mathbf{h}}_e - \bar{\mathbf{h}}_f), \qquad \mathbf{F}_r^{ef} = \eta_r \mathbf{W}_{ef} \frac{\bar{\mathbf{h}}_e - \bar{\mathbf{h}}_f}{||\bar{\mathbf{h}}_e - \bar{\mathbf{h}}_f| + b|^2}, \tag{6.10}$$

where $\bar{\mathbf{h}}_e$ is the feature vector of node $e$ and $\eta_a, \eta_r$ constants controlling the learning rate and $b$ a bias to avoid division by 0. The algorithm is implemented in a parallel learning scheme, where each node is updated independently according to the net force it experiences. The optimization proceeds iteratively until equilibrium is reached, at which point the learning process is terminated. Experimental results in label prediction and visualization showed that GForce outperforms state-of-the-art methods such as DeepWalk and node2vec.

To mitigate the limitation of covering structural features only, this work was subsequently extended in AGForce [K. Sun, Xia, et al. 2022], where the structural embeddings are computed in the same way as in GForce, but node attributes are explicitly incorporated. Specifically, attributes are first processed through an attention mechanism, then concatenated with the structural embeddings and passed to a Conv2D layer to produce enriched feature maps for downstream tasks.

While GForce [K. Sun, J. Liu, et al. 2020] explicitly simulates attractive and repulsive interactions in a physics-inspired manner to preserve structural properties, Force2Vec [Rahman et al. 2020] reformulates the same principle into a differentiable loss function for a node as a negative log likelihood with respect to other nodes, optimized via stochastic gradient descent. The loss function is defined in terms of forces as:

$$\mathcal{L}(e) = -\sum_{v \in \mathcal{N}(e)} \log \psi(\bar{\mathbf{h}}_e, \bar{\mathbf{h}}_v) - \sum_{w \notin \mathcal{N}(e)} \log\Big(1 - \psi(\bar{\mathbf{h}}_e, \bar{\mathbf{h}}_w)\Big), \tag{6.11}$$

where $\psi$ is a similarity function. The first term corresponds to the attractive force between neighboring nodes, while the second term corresponds to the repulsive force between non-neighbors. Different similarity models were investigated, including the sigmoid function, the Student-t distribution, and others derived from graph force layout techniques such as LinLog, Fruchterman–Reingold, and ForceAtlas2. This approach achieved good performance at node classification, link prediction and clustering compared to other embedding approaches.

Table 6.1 – Overview of the Force-directed models and their force formulations.

| **Visualization models** | **Force formulation** | **Reference** |
|---|---|---|
| Tutte barycentric | $\mathbf{F}_a^{ef} = \big(\mathbf{c}_e - \mathbf{c}_f\big)$ | [Tutte 1963 ] |
| Eades | $\|\mathbf{F}_a^{ef}\| = k_a \log\Big(\frac{d_{ef}}{d_0}\Big) \quad \|\mathbf{F}_r^{ef}\| = \frac{k_r}{\sqrt{d_{ef}}}$ | [Eades 1984] |
| Fruchterman–Reingold | $\|\mathbf{F}_a^{ef}\| = \frac{d_{ef}^2}{k_a \frac{\sqrt{\text{area}}}{n}} \quad \|\mathbf{F}_r^{ef}\| = \frac{\big(k_r \frac{\sqrt{\text{area}}}{n}\big)^2}{d_{ef}}$ | [Fruchterman et al. 1991] |
| Node repulsion LinLog | $\|\mathbf{F}_a^{ef}\| = d_{ef} \quad \|\mathbf{F}_r^{ef}\| = \ln(d_{ef})$ | [Noack 2007] |
| Edge repulsion LinLog | $\|\mathbf{F}_a^{ef}\| = d_{ef} \quad \|\mathbf{F}_r^{ef}\| = \mathbf{deg}_e \mathbf{deg}_f \ln(d_{ef})$ | [Noack 2007] |
| ForceAtlas2 | $\|\mathbf{F}_a^{ef}\| = \mathbf{W}_{ef}^{\delta} d_{ef} \quad \|\mathbf{F}_r^{ef}\| = k_r \frac{\big(\mathbf{deg}_e + 1\big)\big(\mathbf{deg}_f + 1\big)}{d_{ef}}$ | [Jacomy et al. 2014] |
| Kamada–Kawai | $E = \sum_{e=1}^{n-1} \sum_{f=e+1}^{n} \frac{1}{2} k_{ef} \big(\|\mathbf{c}_e - \mathbf{c}_f\| - l_{e,f}\big)^2$ | [Kamada et al. 1989] |
| **Embedding / Clustering models** | | |
| GForce | $\mathbf{F}_a^{ef} = -\eta_a \mathbf{W}_{ef}(\bar{\mathbf{h}}_e - \bar{\mathbf{h}}_f) \quad \mathbf{F}_r^{ef} = \eta_r \mathbf{W}_{ef} \frac{\bar{\mathbf{h}}_e - \bar{\mathbf{h}}_f}{\|\bar{\mathbf{h}}_e - \bar{\mathbf{h}}_f\| + b^2}$ | [K. Sun, J. Liu, et al. 2020] |
| AGForce | GForce $\mathbf{F}_a/\mathbf{F}_r$ + attributes | [K. Sun, Xia, et al. 2022] |
| Force2Vec | $\mathcal{L}(e) = -\sum_{v \in \mathcal{N}(e)} \log \psi(\bar{\mathbf{h}}_e, \bar{\mathbf{h}}_v) - \sum_{w \notin \mathcal{N}(e)} \log\big(1 - \psi(\bar{\mathbf{h}}_e, \bar{\mathbf{h}}_w)\big)$ | [Rahman et al. 2020] |

### 6.1.3 Discussion

An overview of all the reviewed approaches is presented in [Tab. 6.1]. To the best of our knowledge, force-based models have not yet been explored in the context of EEG classification. Motivated by their success in the different tasks detailed before, we propose a novel graph embedding strategy tailored to brain connectivity classification. Specifically, we design a new feature representation that integrates the spatial arrangement of electrodes with a data-driven learned graph structure, modeling interaction forces in both two- and three-dimensional coordinate systems. The resulting force-related node features are concatenated into a feature vector suitable for classification. To provide flexibility, we introduce multiple formulations of the attractive and repulsive forces, allowing the intensity of these interactions to be tuned. Furthermore, we investigate how combining force-driven features with signal-based and structure-based features affects graph classification performance, with the goal of improving both the interpretability and the effectiveness of EEG graph representations in spatially structured domains.

## 6.2 Hybrid physics data driven graph force learning framework

Consider a set of EEG signals recorded from a network of $n$ electrodes with known spatial coordinates $\mathcal{C}$. These signals are organized into a matrix $\mathbf{X} \in \mathbb{R}^{n \times tF_s}$, where the $e^{\text{th}}$ row contains $tF_s$ samples from electrode $e$, with $F_s$ denoting the sampling frequency and $t$ the time window length. As recalled earlier, an undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ can be associated with this network, where the set of nodes $\mathcal{V}$ represents the electrodes, the edges $\mathcal{E}$ encode pairwise connections, and the weight $\mathbf{W}_{ef}$ quantifies the strength of connectivity between electrodes $e$ and $f$.
The objective of this work is to transform each graph into a vector representation that simultaneously accounts for the spatial arrangement of the electrodes and the connectivity patterns captured in $\mathbf{W}$. subsequently, these feature vectors are used as input to a classifier for supervised learning tasks.
To achieve this, we propose a Graph Force Learning framework, illustrated in [Fig. 6.2]. In this framework, a weight matrix is first derived from the EEG signals as described in Chapter 3. By exploiting the spatial positions of the electrodes, a physics-inspired model is then applied to compute interaction forces between nodes. From these interactions, a set of force-based node features is extracted and concatenated into feature vectors, which are finally processed by a supervised classification algorithm. The three main stages of this framework are detailed in the following sections.

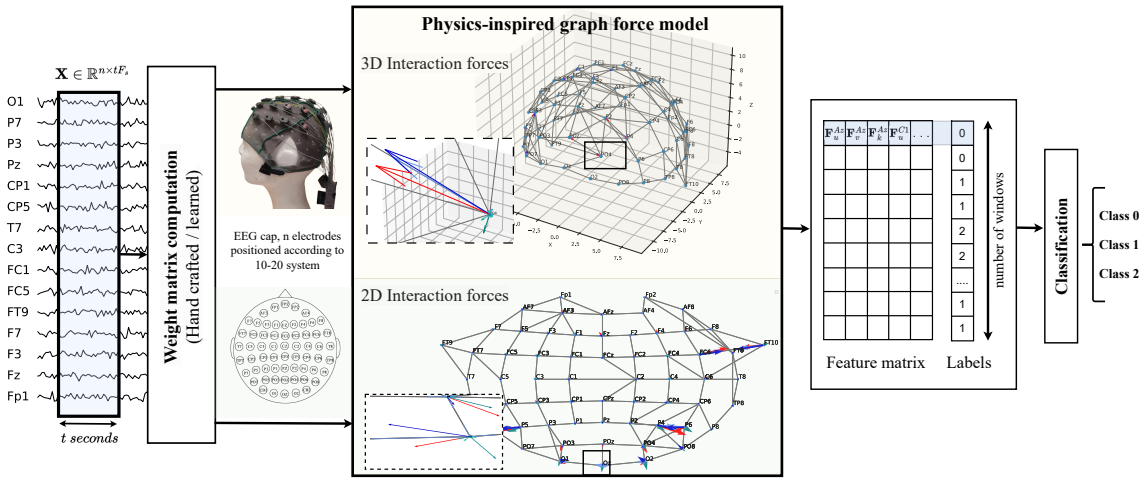



Figure 6.2 – Overview of the proposed graph force learning framework: EEG signals are pre-processed and segmented into overlapping windows of $tF_s$ samples. For each window, a graph is constructed. Using the spatial electrode positions and the graph structure, the forces exerted on each node are then calculated following a formulation in 2D or 3D. The u-, v-, and k-components of these forces are concatenated into a feature vector for each window, which are then classified using an SVM classifier.

### 6.2.1 Data-driven graph learning

Several strategies can be employed to learn the graph structure from data, as previously discussed. In this chapter, we adopt the Laplacian-based approach, which showed promising results in earlier experiments, and use it as a reference against other similarity measures such as Pearson correlation and PLV. Recall that the Laplacian corresponds to the precision matrix learned from data following the method proposed by [Egilmez et al. 2017] and detailed in [Sec. 3.3.3]. The adjacency matrix $\mathbf{W}$ is then obtained by negating the off-diagonal entries of $\mathbf{L}$, with its sparsity pattern encoding conditional independence between node signals. In contrast, the other similarity metrics are computed directly from the data and subsequently sparsified by applying a threshold in order to retain only the most significant connections.

### 6.2.2 Physics-inspired graph force model

Inspired by force directed graph layout models and the physics principle of forces [K. Sun, J. Liu, et al. 2020, Cheong et al. 2020], we propose a graph force model, which represents a weighted graph by a set of forces exerted between the nodes. In contrast to traditional approaches that update node positions based on these forces for visualization purposes, our method retains the original spatial coordinates and focuses on extracting descriptive features. These features capture the graph's structural behavior by analyzing the magnitudes and directions of the forces exerted between nodes in their true spatial configuration.

#### 6.2.2.1 2-D Interaction Forces

Given the weighted graph structure represented by the weight matrix $\mathbf{W}$ and the electrode positions in spherical coordinates $\mathcal{C} = (\mathsf{x}_e, \mathsf{y}_e, \mathsf{z}_e)_{e\in\mathcal{V}}$. First we will compute forces in 2D by considering the xy-coordinates corresponding to the azimuthal plane projection of the electrodes $\mathcal{C}^{2D} = \{(\mathsf{x}_e, \mathsf{y}_e)_{e\in\mathcal{V}}\}$, the attractive and repulsive forces exerted on each node are computed. In the following computations, forces are expressed in the frame $(\vec{\mathsf{u}}, \vec{\mathsf{v}})$ where $\vec{\mathsf{u}}$ and $\vec{\mathsf{v}}$ represent the frame vectors in a Cartesian xy-coordinate system.

**Attractive forces.** The formulation of attractive forces is inspired by Hooke's law for spring-based physical models [Hook 1679h]. [Fig. 6.1] illustrates the principle: edges are modeled as springs, and equilibrium is reached when the net energy at each element tends toward zero. In our context, edges encode connectivity measures that quantify the strength of interaction between two nodes, while the node positions are assumed to be known. Following Hooke's law, the attractive force depends on the Euclidean distance between nodes. To simultaneously preserve connectivity and incorporate the spatial arrangement of the electrodes, we weight this force by the connectivity value stored in the matrix $\mathbf{W}$, similar to the formulation adopted in ForceAtlas2 [Jacomy et al. 2014].

Consider a node $f$ in the one-hop neighborhood of a node $e$ (*i.e.*, $\mathbf{W}_{ef} \neq 0$). Node $f$ exerts an attractive force $\mathbf{F}_a^{ef}$ on node $e$. The magnitude of this force is proportional to both the connection strength $\mathbf{W}_{ef}$ and the Euclidean distance $d_{ef}$

$$d_{ef} = \sqrt{(\mathsf{x}_e - \mathsf{x}_f)^2 + (\mathsf{y}_e - \mathsf{y}_f)^2} \tag{6.12}$$

The attractive force vector exerted on node $e$ by node $f$ is therefore given by

$$\mathbf{F}_a^{ef} = \begin{bmatrix} \mathbf{F}_{a,\mathsf{u}}^{ef} \\ \mathbf{F}_{a,\mathsf{v}}^{ef} \end{bmatrix} = \mathbf{W}_{ef} \cdot d_{ef} \begin{bmatrix} \cos\beta_{ef} \\ \sin\beta_{ef} \end{bmatrix} = \mathbf{W}_{ef} \begin{bmatrix} \mathsf{x}_f - \mathsf{x}_e \\ \mathsf{y}_f - \mathsf{y}_e \end{bmatrix}, \tag{6.13}$$

where $\beta_{ef}$ denotes the angle of $\mathbf{F}_a^{ef}$, computed as

$$\beta_{ef} = \arctan\left(\frac{\mathsf{y}_f - \mathsf{y}_e}{\mathsf{x}_f - \mathsf{x}_e}\right), \tag{6.14}$$

The attractive forces as we formulate it in 2D is illustrated in [Fig. 6.6].

**Repulsive forces.** When a vertex $f$ is not directly connected to a vertex $e$ (*i.e.*, $\mathbf{W}_{ef} = 0$), it exerts on $e$ a repulsive force $\mathbf{F}_r^{ef}$. This formulation is inspired by Coulomb's law [Coulomb 1785], which states that the magnitude of the electrostatic force between two charges is proportional to the product of their charges and inversely proportional to the

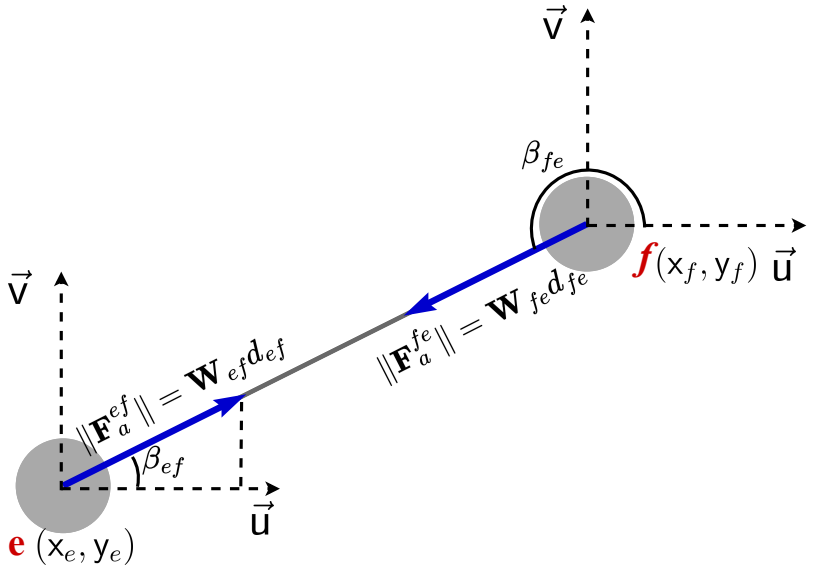



Figure 6.3 – Attractive force between two connected nodes $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_a^{ef}$. Here, $\mathbf{W}_{ef}$ is the connection weight and $d_{ef}$ the Euclidean distance between the two nodes.

square of the distance separating them. By analogy, nodes are considered as point charges whose magnitudes correspond to their degrees. Hence, if two nodes are not linked by an edge, they repel each other with a force that grows with their degrees and decreases with distance.

Formally, the repulsive force exerted by node $f$ on node $e$ is expressed as:

$$\mathbf{F}_r^{ef} = \begin{bmatrix} \mathbf{F}_{r,\mathsf{u}}^{ef} \\ \mathbf{F}_{r,\mathsf{v}}^{ef} \end{bmatrix} = -\frac{\mathbf{D}_{ee} \cdot \mathbf{D}_{ff}}{d_{ef}} \begin{bmatrix} \cos\beta_{ef} \\ \sin\beta_{ef} \end{bmatrix} = -\frac{\mathbf{D}_{ee} \cdot \mathbf{D}_{ff}}{d_{ef}^2} \begin{bmatrix} x_f - x_e \\ y_f - y_e \end{bmatrix}, \tag{6.15}$$

where $d_{ef}$ denotes the Euclidean distance between the two nodes, $\mathbf{D}_{ee}$ is the degree of node $e$ and $\beta_{ef}$ the angle associated with the attractive force direction. As illustrated in [Fig. 6.8], the orientation of the repulsive force is opposite to that of the attractive force.

This formulation is closely related to the approach adopted in ForceAtlas2 [Jacomy et al. 2014] (see Eq. 6.9). By construction, it tends to push weakly connected nodes away from one another, while allowing highly connected vertices to attract weaker ones, thereby enhancing structural separation in the embedding.

**Resultant force.** In force-directed layouts, the purpose of computing forces between nodes is typically to displace them until the system reaches equilibrium, yielding either well-separated groups or compact clusters. In our setting, the electrode positions are fixed and must be preserved. The role of force computation changes: instead of updating positions, we derive features corresponding to the forces acting on each electrode, thereby

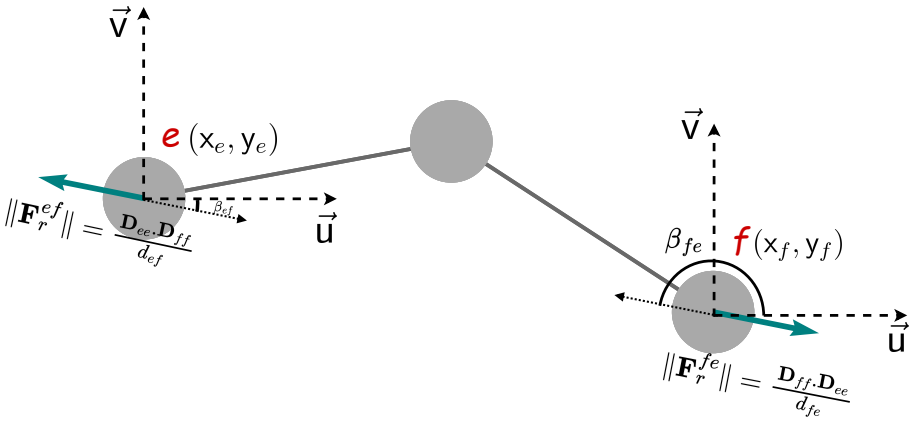



Figure 6.4 – Repulsive force between two unconnected nodes $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_r^{ef}$. Here, $\mathbf{D}_{ee}$ and $\mathbf{D}_{ff}$ are the node degrees, and $d_{ef}$ is the Euclidean distance between the two nodes.

capturing their strength and orientation.

These features are expressed through the *resultant force*, defined for each node as the sum of attractive forces from its neighbors and repulsive forces from all other nodes.

Formally, the resultant force $\mathbf{F}^e$ acting on node $e$ is given by

$$\mathbf{F}^e = \begin{bmatrix} \mathbf{F}^e_{\mathsf{u}} \\ \mathbf{F}^e_{\mathsf{v}} \end{bmatrix} = \begin{bmatrix} \sum_{f \in \mathcal{N}_1(e)} \mathbf{F}^{ef}_{a,\mathsf{u}} + \sum_{f \notin \mathcal{N}_1(e)} \mathbf{F}^{ef}_{r,\mathsf{u}} \\ \sum_{f \in \mathcal{N}_1(e)} \mathbf{F}^{ef}_{a,\mathsf{v}} + \sum_{f \notin \mathcal{N}_1(e)} \mathbf{F}^{ef}_{r,\mathsf{v}} \end{bmatrix}, \tag{6.16}$$

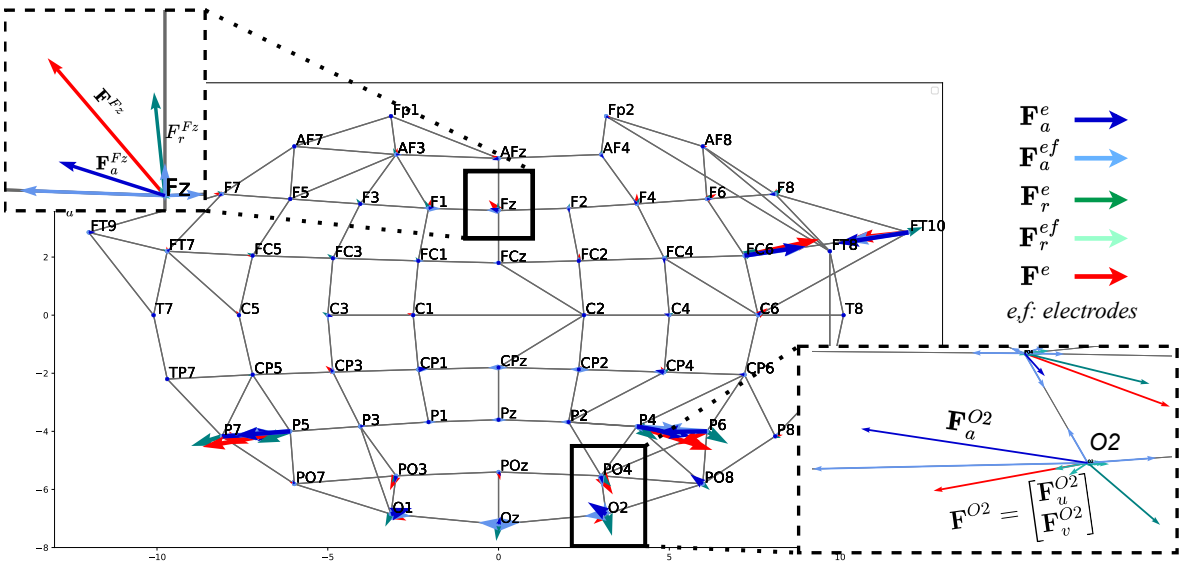



Figure 6.5 – Example of the resultant 2D force acting on each electrode, expressed as the sum of all attractive and repulsive forces applied to the node.

#### 6.2.2.2 3-D Interaction forces

Since the brain is inherently three-dimensional, it is natural to extend the force formulation into 3D using the electrode coordinates $\mathcal{C} = \{(\mathsf{x}_e, \mathsf{y}_e, \mathsf{z}_e)\}_{e\in\mathcal{V}}$.
In this setting, forces are computed in the reference frame $(\vec{\mathsf{u}}, \vec{\mathsf{v}}, \vec{\mathsf{k}})$. The principle remains the same as in 2D; the main modifications are the introduction of the $z$-component and two parameters for generalization: a scaling factor and a power parameter as detailed in the sequel.

**Attractive forces.** A vertex $f$ is considered a neighbor of vertex $e$ if $\mathbf{W}_{ef} \neq 0$, in which case it exerts an attractive force $\mathbf{F}_a^{ef}$ on $e$.
In the 3D formulation, the force magnitude is determined by the Euclidean distance $d_{ef}$, scaled by three factors: the connection strength $\mathbf{W}_{ef}$, which encodes how strongly $e$ and $f$ are related, a global scaling coefficient $k_a$, which adjusts the overall intensity of attraction in the resultant force, and a power parameter $\lambda_a$, which modulates the dependence on distance. With the Euclidean distance given by

$$d_{ef} = \sqrt{(\mathsf{x}_e - \mathsf{x}_f)^2 + (\mathsf{y}_e - \mathsf{y}_f)^2 + (\mathsf{z}_e - \mathsf{z}_f)^2}, \tag{6.17}$$

the attractive force vector can first be expressed in spherical coordinates as

$$\mathbf{F}_a^{ef} = \begin{bmatrix} \mathbf{F}_{a,\mathsf{u}}^{ef} \\ \mathbf{F}_{a,\mathsf{v}}^{ef} \\ \mathbf{F}_{a,\mathsf{k}}^{ef} \end{bmatrix} = k_a \cdot \mathbf{W}_{ef} \cdot d_{ef}^{\lambda_a} \begin{bmatrix} \cos\bar{\theta}_{ef}\ \sin\bar{\phi}_{ef} \\ \sin\bar{\theta}_{ef}\ \sin\bar{\phi}_{ef} \\ \cos\bar{\phi}_{ef} \end{bmatrix}, \tag{6.18}$$

where the direction of the force is given in spherical coordinates by the azimuthal angle $\bar{\theta}_{ef}$ (measured in the $(\vec{\mathsf{u}}, \vec{\mathsf{v}})$-plane) and the polar angle $\bar{\phi}_{ef}$ (measured from the vertical axis $\vec{\mathsf{k}}$) (see [Fig. 6.6]).

Equivalently, by substituting the spherical coordinates with Cartesian components, the attractive force can be rewritten as

$$\mathbf{F}_a^{ef} = \begin{bmatrix} \mathbf{F}_{a,\mathsf{u}}^{ef} \\ \mathbf{F}_{a,\mathsf{v}}^{ef} \\ \mathbf{F}_{a,\mathsf{k}}^{ef} \end{bmatrix} = k_a \cdot \mathbf{W}_{ef} \cdot d_{ef}^{(\lambda_a - 1)} \begin{bmatrix} \mathsf{x}_f - \mathsf{x}_e \\ \mathsf{y}_f - \mathsf{y}_e \\ \mathsf{z}_f - \mathsf{z}_e \end{bmatrix}. \tag{6.19}$$

**Repulsive forces.** When two vertices are not directly connected ($\mathbf{W}_{ef} = 0$), vertex $f$ exerts a repulsive force $\mathbf{F}_r^{ef}$ on vertex $e$. Its magnitude depends on the product of their degrees $\mathbf{D}_{ee}$ and $\mathbf{D}_{ff}$, scaled by a repulsive factor $k_r$, and decays with distance according to an exponent $\lambda_r$. This allows the formulation to reproduce the inverse-square inspired by Coulomb law when $\lambda_r = 2$, while other values yield slower or faster decay. As illustrated

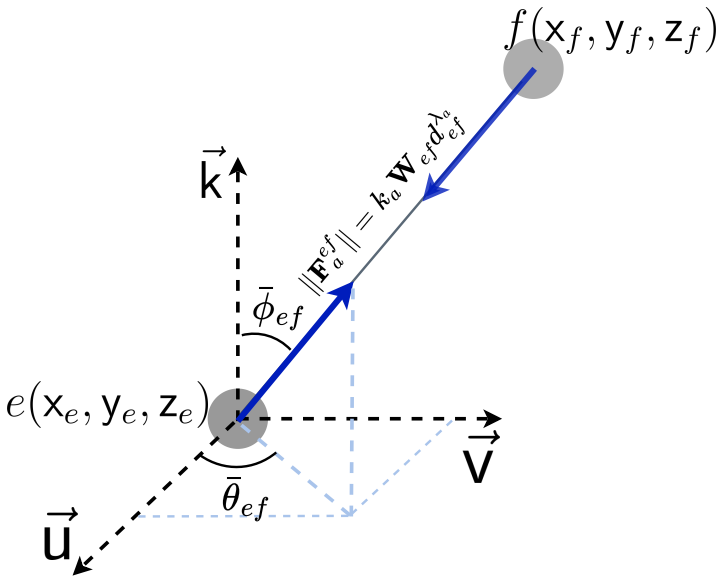



Figure 6.6 – Attractive force between two connected vertices $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_a^{ef}$. Here, $\mathbf{W}_{ef}$ is the connection weight, $d_{ef}$ the Euclidean distance between the vertices, and $k_a$ a scaling factor.

in [Fig. 6.8], the repulsive force is defined in spherical coordinates as:

$$\mathbf{F}_r^{ef} = \begin{bmatrix} \mathbf{F}_{r,\mathsf{u}}^{ef} \\ \mathbf{F}_{r,\mathsf{v}}^{ef} \\ \mathbf{F}_{r,\mathsf{k}}^{ef} \end{bmatrix} = -k_r \cdot \frac{\mathbf{D}_{ee} \cdot \mathbf{D}_{ff}}{d_{ef}^{\lambda_r}} \begin{bmatrix} \cos\bar{\theta}_{ef} \sin\bar{\phi}_{ef} \\ \sin\bar{\theta}_{ef} \sin\bar{\phi}_{ef} \\ \cos\bar{\phi}_{ef} \end{bmatrix}. \tag{6.20}$$

and equivalently in Cartesian coordinates as:

$$\mathbf{F}_r^{ef} = \begin{bmatrix} \mathbf{F}_{r,\mathsf{u}}^{ef} \\ \mathbf{F}_{r,\mathsf{v}}^{ef} \\ \mathbf{F}_{r,\mathsf{k}}^{ef} \end{bmatrix} = -k_r \cdot \frac{\mathbf{D}_{ee} \cdot \mathbf{D}_{ff}}{d_{ef}^{(\lambda_r+1)}} \begin{bmatrix} \mathsf{x}_f - \mathsf{x}_e \\ \mathsf{y}_f - \mathsf{y}_e \\ \mathsf{z}_f - \mathsf{z}_e \end{bmatrix}. \tag{6.21}$$

**Resultant forces.** The resultant force acting on a vertex $e$ is obtained by combining the attractive contributions from its neighbors in $\mathcal{N}_1(e)$ and the repulsive contributions from all non-neighboring vertices. The total force vector expressed in the $(\vec{\mathsf{u}}, \vec{\mathsf{v}}, \vec{\mathsf{k}})$ reference frame is expressed as:

$$\mathbf{F}^e = \begin{bmatrix} \mathbf{F}_{\mathsf{u}}^e \\ \mathbf{F}_{\mathsf{v}}^e \\ \mathbf{F}_{\mathsf{k}}^e \end{bmatrix} = \begin{bmatrix} \sum_{f \in \mathcal{N}_1(e)} \mathbf{F}_{a,\mathsf{u}}^{ef} + \sum_{f \notin \mathcal{N}_1(e)} \mathbf{F}_{r,\mathsf{u}}^{ef} \\ \sum_{f \in \mathcal{N}_1(e)} \mathbf{F}_{a,\mathsf{v}}^{ef} + \sum_{f \notin \mathcal{N}_1(e)} \mathbf{F}_{r,\mathsf{v}}^{ef} \\ \sum_{f \in \mathcal{N}_1(e)} \mathbf{F}_{a,\mathsf{k}}^{ef} + \sum_{f \notin \mathcal{N}_1(e)} \mathbf{F}_{r,\mathsf{k}}^{ef} \end{bmatrix}. \tag{6.22}$$

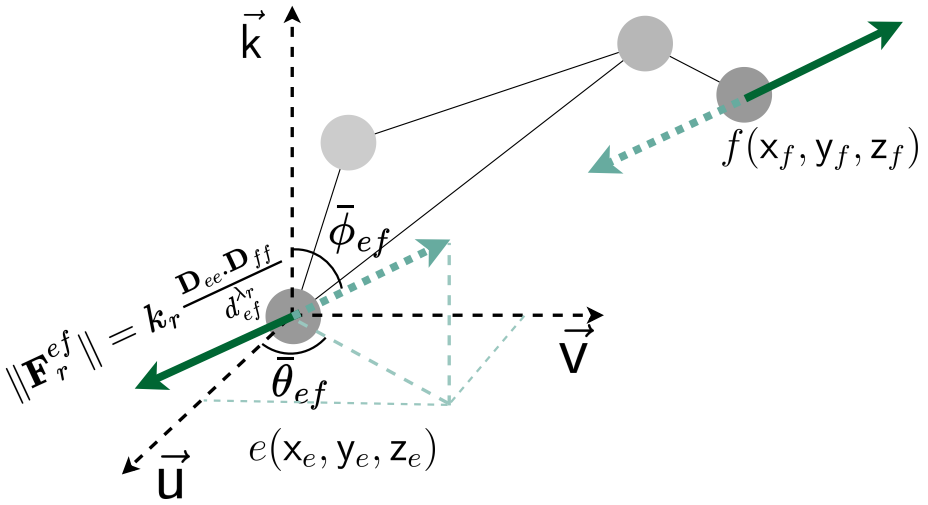



Figure 6.7 – Repulsive force between two unconnected vertices $e$ and $f$. The force exerted by $f$ on $e$ is denoted by $\mathbf{F}_r^{ef}$. Here, $\mathbf{D}_{ee}$ and $\mathbf{D}_{ff}$ are the node degrees, $d_{ef}$ the Euclidean distance, and $k_r$ a scaling factor.

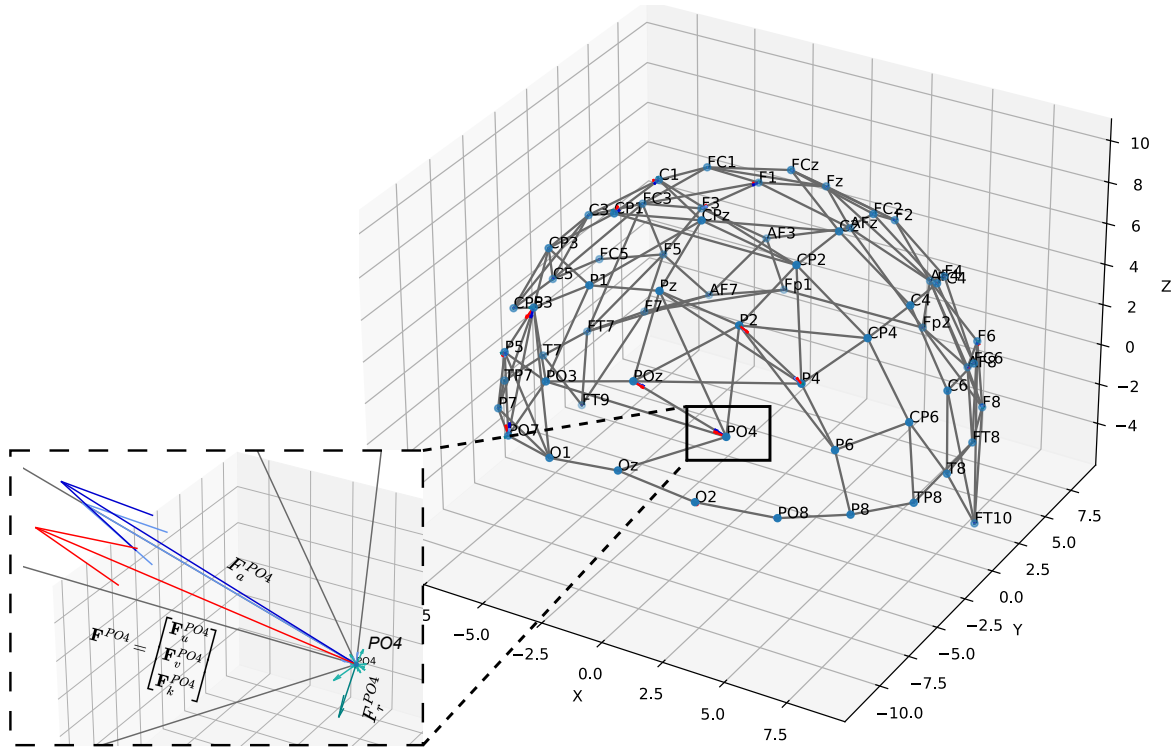



Figure 6.8 – Example of the resultant 3D force acting on electrodes expressing the sum of attractive and repulsive forces.

### 6.2.3 Classification based on force-related features

Force-related features for each vertex $e$ are defined from the components of its resultant force vector, denoted

$$\mathbf{F}^e = \begin{cases} \left(\mathbf{F}^e_{\mathsf{u}}, \mathbf{F}^e_{\mathsf{v}}\right), & \text{in 2-D layouts,} \\ \left(\mathbf{F}^e_{\mathsf{u}}, \mathbf{F}^e_{\mathsf{v}}, \mathbf{F}^e_{\mathsf{k}}\right), & \text{in 3-D layouts.} \end{cases}$$

These quantities describe, for each electrode, the effect of the attractive and repulsive forces exerted by all other vertices in the network. Intuitively, they capture both how strongly a node is influenced by the surrounding graph structure and in which spatial direction this influence is oriented. To exploit these local force features into machine learning representations, several strategies can be employed:

1. **Direct component encoding.** The force components $\mathbf{F}^e_{\mathsf{u}}$, $\mathbf{F}^e_{\mathsf{v}}$ (and $\mathbf{F}^e_{\mathsf{k}}$ in 3-D) can be directly used as node features. Each vertex is thus described by the signed strength of the forces acting along each spatial axis.

2. **Magnitude–phase representation.** Alternatively, in the case of 2-D layout, the resultant vector can be expressed in terms of its magnitude and its direction (phase angles).

At the graph level, a feature vector representing one time window is built by concatenating the force-derived features of all electrodes into a single vector:

$$\mathbf{F}_{\text{graph}} = \left[\mathbf{F}^1 \parallel \mathbf{F}^2 \parallel \cdots \parallel \mathbf{F}^e\right], \quad e \in \mathcal{V} \tag{6.23}$$

where $\mathbf{F}^e$ contains the chosen force representation for vertex $e$. This representation can be further enriched by incorporating additional node-specific attributes derived from the EEG signals, such as power spectral densities computed across different frequency bands. Moreover, graph-structural features (e.g., node degree, betweenness centrality, clustering coefficients) can be added to embed topological information alongside the force-based features.

The resulting feature matrix is obtained by stacking feature vectors of each window signals into a matrix and can serve as input to a wide range of learning algorithms. It can be used in unsupervised or in supervised classification tasks where each window is associated with a class. In this work, we focus on the supervised paradigm and evaluate several commonly used classifiers Support Vector Machines, Random Forests, and XGBoost while keeping in mind that the framework is general and could easily accommodate other classifiers.

## 6.3 Experimental results and analysis

In this section, we conduct an ablation study to assess the impact of the different design choices. Specifically, we examine: (i) the effect of the underlying graph structure, (ii) the influence of computing the forces in the 2D plane versus in 3D space, (iii) the role of the classifier on performance, and (iv) the comparison of our approach against baseline methods. We also investigate the contribution of the proposed force-based features by studying their discriminative power, exploring their fusion with additional features, and analyzing the effect of the proposed hyperparameters. Finally, we highlight the advantage of this approach for visualization. The details of each component are presented in the following subsections.

### 6.3.1 Experimental setting

The proposed force-based graph embedding is evaluated on two EEG classification tasks: MW using the COG-BCI dataset [Hinss et al. 2023], and MI using the HG-MI dataset [Schirrmeister et al. 2017]. Both datasets, together with their preprocessing and windowing strategies, were previously described in Chapter 4. To summarize the aspects relevant to the present analysis:

— **COG-BCI dataset**: We consider 29 participants and focus on MATB-II recordings from sessions 1 and 2. Each participant performed three independent five-minute trials at low, medium, and high difficulty levels. After preprocessing [Hinss et al. 2023], 61 electrodes sampled at 250 Hz are retained. EEG signals are segmented into overlapping windows of 2 s with a 1 s overlap.

— **HG-MI dataset**: contains four motor imagery classes recorded from 14 participants. Trials are non-continuous, and therefore no overlapping windows are applied.

The proposed framework is applied separately to each subject and session.

**2D and 3D force representations.** For the 2D experiments, electrode positions are obtained from the EEGRAPH library [Maitin et al. 2023], which provides a dictionary of standardized electrode configurations. These positions correspond to an azimuthal equidistant projection onto the 2D plane, which preserves inter-electrode distances. This choice ensures consistency with the FG2IMG approach introduced in the previous chapter. The 3D forces are computed as described in [Sec. 6.2.2.2], using the true coordinates provided in the dataset with parameters fixed to $\lambda_a = \lambda_r = 1$ and $k_a = k_r = 1$.

**Classification.** For the analysis of force-based and node-related features, we evaluate the classification performance using a SVM classifier, whose hyperparameters are optimized through a 5-fold cross-validation procedure. Hyperparameters are optimized by grid

search, considering: the regularization strength $\in \{0.1, 1, 10, 100\}$; the kernel type (linear or Radial Basis Function kernel (RBF)); and, in the case of the RBF kernel, the $\gamma_{RBF}$ parameter, which controls the influence of individual training samples (a small $\gamma_{RBF}$ value implies a wider influence region, whereas a large $\gamma_{RBF}$ restricts the influence to nearby points).
For each subject and session, the best model configuration is selected via grid search, and the mean accuracy over the 5 folds is retained. Final results are reported as the average accuracy ($\pm$ standard deviation) across all participants and sessions.

### 6.3.2 Classification results and analysis

We now present the quantitative results of the different strategies in order to highlight the impact of each component in the proposed pipeline. We first analyze the influence of the graph construction strategy, followed by the choice of force-based node features. The corresponding results are reported in [Tab. 6.3].

#### 6.3.2.1 Impact of the choice of the node force-related features

We first investigate the impact of force-related node features. To this end, we compare the two proposed encoding strategies in 2D and in 3D for the different graph structures, using an SVM classifier. For the 2D approach, we consider either the direct encoding of the force components $[\mathbf{F}_u, \mathbf{F}_v]$ or their transformation into magnitude and orientation angle $[\|\mathbf{F}\|, \beta_F]$, with the learned Laplacian method used to define the graph structure. The corresponding results are reported in [Tab. 6.2].
For the COG-BCI dataset, the results obtained with force components in 2D are comparable, with a slight advantage around 2% in average accuracy for the component-based representation over the magnitude–angle representation. When extending the analysis to 3D forces, we observe an improvement in mean classification accuracy compared to 2D features. A similar trend is observed for the HG-MI dataset, where the use of 3D force features consistently enhances performance. In all configurations, PLV achieves the lowest scores, while Pearson correlation performs better but still falls behind the learned Laplacian.

Table 6.2 – SVM classification accuracy (mean $\pm$ std) for different node feature representations, graph structure is computed using the Learned Laplacian. Best results per dataset are in **bold**.

| **Dataset** | $[\|\mathbf{F}\|, \beta_F]$ | $[\mathbf{F}_u, \mathbf{F}_v]$ | $[\mathbf{F}_u, \mathbf{F}_v, \mathbf{F}_k]$ |
|---|---|---|---|
| COG-BCI | $0.88 \pm 0.08$ | $\mathbf{0.90 \pm 0.08}$ | $\mathbf{0.90 \pm 0.08}$ |
| HG-MI | $0.71 \pm 0.10$ | $0.72 \pm 0.11$ | $\mathbf{0.74 \pm 0.11}$ |

#### 6.3.2.2 Impact of the graph structure

The next analysis concerns the choice of graph structure by comparing our learned Laplacian approach with two hand-crafted functional connectivity measures: the Pearson correlation, already considered in the previous chapter, and the Phase Locking Value [Tab. 3.2], which is widely used in the EEG literature [Klepl, F. He, et al. 2022]. Both measures estimate edge weights by computing pairwise connections between electrodes, resulting in fully connected graphs. To obtain sparser and more interpretable structures, negative connections are discarded and a threshold $T$ is set to be the median of all pairwise values computed across the training set graphs retaining only the strongest positive connections.

Results are reported in [Tab. 6.3], learning a sparse graph Laplacian from EEG signals consistently yields the best performance across both datasets, while PLV achieves the lowest accuracy. For the COG-BCI dataset, Pearson correlation performs about 10% better than PLV but remains approximately 12% lower than the learned Laplacian. A similar trend is observed on the HG-MI dataset, although in this case the performances of Pearson correlation and PLV are closer to each other.

Unlike Pearson correlation and PLV, which capture only pairwise relationships and ignore the broader network dependencies, the precision matrix underlying the learned Laplacian effectively incorporates conditional dependencies between electrodes. Furthermore, while Pearson correlation and PLV require explicit thresholding to sparsify the connectivity matrix, graph learning directly enforces sparsity through the optimization process, thereby avoiding the need for an arbitrary threshold.

Table 6.3 – SVM classification accuracy (mean ± std) for different graph construction strategies and node feature representations across two datasets. Best results per column for each dataset is **bold**

| **Dataset** | **Graph Structure** | $[\|\|\mathbf{F}\|\|, \beta_F]$ | $[\mathbf{F}_u, \mathbf{F}_v]$ | $[\mathbf{F}_u, \mathbf{F}_v, \mathbf{F}_k]$ |
|---|---|---|---|---|
| COG-BCI | Pearson Correlation | $0.75 \pm 0.08$ | $0.77 \pm 0.08$ | $0.78 \pm 0.08$ |
| | Phase Locking Value | $0.63 \pm 0.07$ | $0.66 \pm 0.07$ | $0.68 \pm 0.07$ |
| | Learned Laplacian | $\mathbf{0.88 \pm 0.08}$ | $\mathbf{0.90 \pm 0.08}$ | $\mathbf{0.90 \pm 0.08}$ |
| HG-MI | Pearson Correlation | $0.57 \pm 0.09$ | $0.59 \pm 0.09$ | $0.59 \pm 0.09$ |
| | Phase Locking Value | $0.54 \pm 0.08$ | $0.56 \pm 0.09$ | $0.58 \pm 0.10$ |
| | Learned Laplacian | $\mathbf{0.71 \pm 0.10}$ | $\mathbf{0.72 \pm 0.11}$ | $\mathbf{0.74 \pm 0.11}$ |

#### 6.3.2.3 Impact of the choice of classifier

We select the 3D components as features for the rest of this analysis, now we compare two other algorithms for classification: Random Forest (RF) and Extreme Gradient

Boosting (XGBoost). A grid search with cross-validation was employed to optimize the hyperparameters of both classifiers as it was done for the SVM.
For the RF classifier, the following parameters were explored: the number of trees in the forest $\{50, 100, 200\}$, the maximum depth of the trees $\{\text{None}, 10, 20\}$, the minimum number of samples required to split an internal node $\{2, 5, 10\}$, the minimum number of samples required to be at a leaf node $\{1, 2, 4\}$, and whether bootstrap sampling was used for training $\{\text{True}, \text{False}\}$. For the XGBoost classifier, the search space included the learning rate $\{0.001, 0.01, 0.1\}$, the maximum depth of the trees $\{3, 5, 7\}$, and the number of estimators $\{50, 100, 150\}$. The best configuration for each classifier was selected based on cross-validation accuracy, ensuring a fair comparison across graph construction strategies and classifiers.
The corresponding results are reported in [Tab. 6.4]. As expected, the learned Laplacian consistently yields the best performance across both datasets, confirming the trend already observed with the SVM classifier. When comparing classifiers, XGBoost slightly outperforms RF for most graph structures, although the difference remains marginal. Both methods, however, achieve lower accuracy than the SVM, with a gap of approximately 4% on average for the COG-BCI dataset for PC and PLV. A similar pattern is observed for the HG-MI dataset, where RF and XGBoost perform comparably, but a substantial improvement is obtained when using the SVM with a gain around 5%.

Table 6.4 – Classification accuracy (mean ± std) of different algorithms across graph construction strategies. Best results per row and column are **bold**

| Dataset | Graph Structure | RF | XGBoost | SVM |
|---|---|---|---|---|
| COG-BCI | Pearson Correlation | $0.70 \pm 0.07$ | $0.73 \pm 0.08$ | $\mathbf{0.78 \pm 0.08}$ |
| | Phase Locking Value | $0.64 \pm 0.07$ | $0.66 \pm 0.07$ | $\mathbf{0.68 \pm 0.07}$ |
| | Learned Laplacian | $\mathbf{0.89 \pm 0.08}$ | $\mathbf{0.90 \pm 0.08}$ | $\mathbf{0.90 \pm 0.08}$ |
| HG-MI | Pearson Correlation | $0.56 \pm 0.07$ | $0.56 \pm 0.08$ | $\mathbf{0.59 \pm 0.09}$ |
| | Phase Locking Value | $0.56 \pm 0.09$ | $0.56 \pm 0.08$ | $\mathbf{0.58 \pm 0.10}$ |
| | Learned Laplacian | $\mathbf{0.69 \pm 0.10}$ | $\mathbf{0.69 \pm 0.11}$ | $\mathbf{0.74 \pm 0.11}$ |

#### 6.3.2.4 Comparison against baseline classification pipelines

Finally, the proposed framework is compared against two classification pipelines that serve as baselines in this manuscript and were also considered in the previous chapter ([Tab. 5.4]): the R-MDM and CSP-OVR methods. For fairness, CSP-OVR is evaluated using an SVM classifier, since our approach also relies on SVM.
The results in [Tab. 6.5] show that our framework, which learns the Laplacian from EEG data and represents it through force-based features in 2D and 3D, clearly outperforms

conventional spatial-domain methods for mental workload classification on the COG-BCI dataset [Hinss et al. 2023], achieving gains of approximately 8% over CSP and 28% over R-MDM. For motor imagery classification using the HG-MI dataset [Schirrmeister et al. 2017], CSP-OVR combined with SVM achieves slightly better accuracy (about 2% higher) than our method, whereas R-MDM remains significantly weaker, with our framework showing an advantage of roughly 22%.
These results suggest that force-based graph representations provide a more informative and discriminative feature space for EEG classification. They also highlight that the relative performance of different methods depends strongly on the task and inter-subject variability.

Table 6.5 – Comparison against baseline approaches for EEG classification. Best results in column is **bold**.

| **Features** | **Classifier** | **COG-BCI** | **HG-MI** |
|---|---|---|---|
| Covariance Matrix | R-MDM | $0.62 \pm 0.08$ | $0.52 \pm 0.10$ |
| CSP-OVR | LDA | $0.78 \pm 0.14$ | $0.73 \pm 0.10$ |
| CSP-OVR | SVM | $0.82 \pm 0.12$ | $\mathbf{0.76 \pm 0.10}$ |
| Laplacian $[\mathbf{F}_u, \mathbf{F}_v]$ | SVM | $\mathbf{0.90 \pm 0.08}$ | $0.72 \pm 0.11$ |
| Laplacian $[\mathbf{F}_u, \mathbf{F}_v, \mathbf{F}_k]$ | SVM | $\mathbf{0.90 \pm 0.08}$ | $0.74 \pm 0.11$ |

#### 6.3.2.5 Relevance of graph-force features

To assess the importance and relevance of force-based features compared to other feature types, we consider two additional sets of node-level features : **Spectral features**, namely the relative power spectral density computed for each electrode across six frequency bands and **Structural features** that capture the role of each node within the graph including degree, clustering coefficient, betweenness centrality, closeness centrality, and eigenvector centrality as detailed in [Sec. 3.1].
We first train the classifier using each feature type independently to evaluate its discriminative power. As shown in [Tab. 6.6], for both datasets, force-based features outperform both spectral and structural features when used alone, although the latter still achieve reasonable accuracy.
Motivated by this observation, we then consider their combination. The three feature sets force-based, structural, and spectral are computed at the node level, concatenated, and subsequently flattened across all electrodes to form a single graph-level feature vector. We evaluate the classification performance of this combined representation and further investigate the effect of applying a feature selection step, where the number of selected features is treated as a tunable hyperparameter.
The results in [Tab. 6.6] show that combining all feature types improves performance

compared to using any single set, and that incorporating a feature selection step further boosts accuracy on both datasets achieving performance that slightly surpasses the baseline CSP+SVM pipeline for the HG-MI task.

Table 6.6 – Classification accuracy (mean ± std) of different signal-related feature sets on the two datasets.

| **Feature Set** | **Feature Selection** | **COG-BCI** | **HG-MI** |
|---|---|---|---|
| Relative PSD | ✗ | $0.83 \pm 0.06$ | $0.68 \pm 0.10$ |
| Structural features | ✗ | $0.87 \pm 0.08$ | $0.69 \pm 0.10$ |
| Force features | ✗ | $0.90 \pm 0.07$ | $0.74 \pm 0.11$ |
| Force \|\| Relative PSD \|\| Structural | ✗ | $0.90 \pm 0.06$ | $0.76 \pm 0.10$ |
| Force \|\| Relative PSD \|\| Structural | ✓ | $\mathbf{0.93 \pm 0.06}$ | $\mathbf{0.77 \pm 0.10}$ |

We next analyze the contribution of force-based features when combined with other feature types, in order to better understand their role in the classification process. To this end, we examine the outcome of our feature selection procedure, where the SelectKBest method with an ANOVA $F$-value scoring function is employed [Pedregosa et al. 2011] This function evaluates each feature independently by measuring the ratio of between-class variance to within-class variance. We vary the total number of selected features and, for each setting, analyze the composition of the resulting subsets to determine which feature types are prioritized by the model.

[Fig. 6.9] reports the results for the COG-BCI dataset containing 61 electrodes. When using the entire feature set (854 features: $61 \times (3 + 5 + 6)$), the distribution is unbalanced, with force-related features representing the smallest proportion. However, as the number of selected features decreases, the proportion of retained force features increases, highlighting their higher relevance in classification. A similar analysis is performed on the HG-MI dataset, which includes 44 electrodes and therefore a smaller overall feature set. As shown in [Fig. 6.10], the same trend is observed: for subsets of 50, 100, and 300 features, force-related features dominate the selection. Moreover, the highest classification accuracy is achieved when using 300 features, further supporting the discriminative power of force-based features. In both datasets, PSD features are consistently under-represented in the selected subsets compared to their proportion in the original feature space. From these experiments, we conclude that force-related and graph structure features capture essential information that is critical for distinguishing between classes. Furthermore, combining them with structural and spectral features may lead to enhanced classification performance.

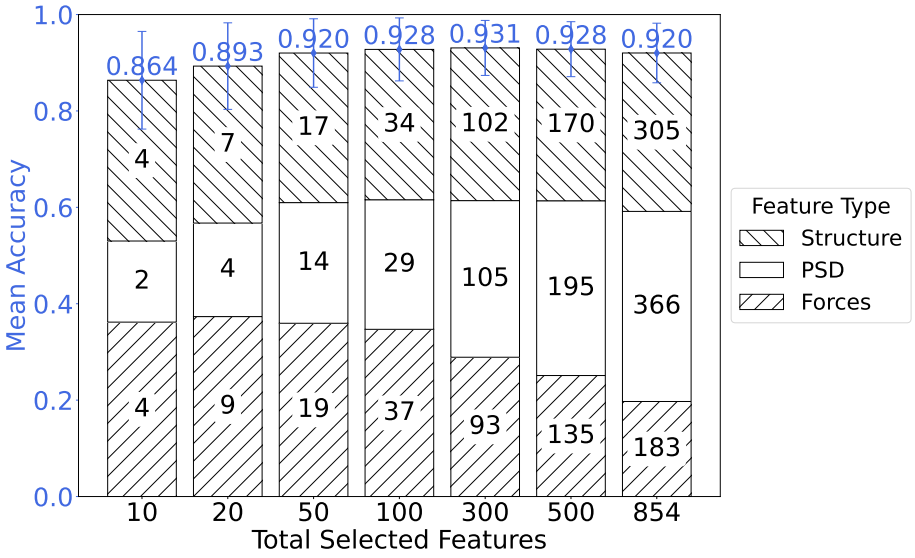



Figure 6.9 – Proportion of selected force, PSD, and structural features for varying total feature counts for COG-BCI dataset. As fewer features are selected, the proportion of force features increases, suggesting their higher discriminative power. Mean classification accuracy with standard deviation is also reported.

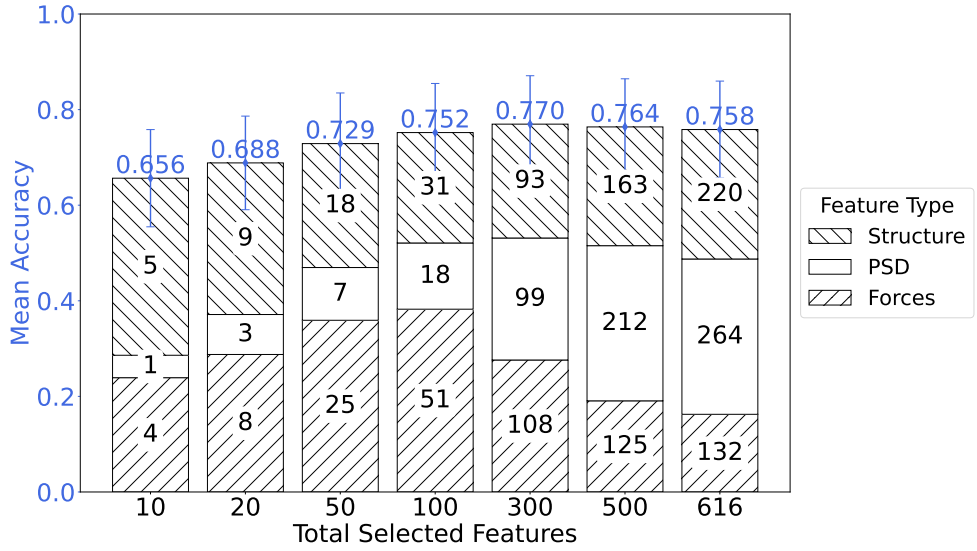



Figure 6.10 – Proportion of selected force, PSD, and structural features for varying total feature counts for HG-MI dataset. As fewer features are selected, the proportion of force features increases, suggesting their higher discriminative power. Mean classification accuracy with standard deviation is also reported.

#### 6.3.2.6 Effect of force parameters on the classification

To evaluate the influence of the parameters involved in force computation, we selected two representative participants, one from each dataset, and tested different parameter configurations.
First, the scaling factors were fixed to $k_a = k_r = 1$, and the power parameters $\lambda_a$ and $\lambda_r$ were varied in the range $[-2, 2]$ with a step of 0.25. For the COG-BCI subject with the lowest classification performance (Subject 12, Session 2), [Fig. 6.11](left) shows that the best accuracy (0.73) was obtained when $\lambda_a = \lambda_r = 0.25$, compared to a baseline accuracy of 0.65.

For the HG-MI dataset, we selected Subject 7, whose performance was close to the average of the group. As shown in [Fig. 6.12](left), the best performance (0.70) was reached when $\lambda_a = 1$ and $\lambda_r = 1.25$, compared to a baseline of 0.68.

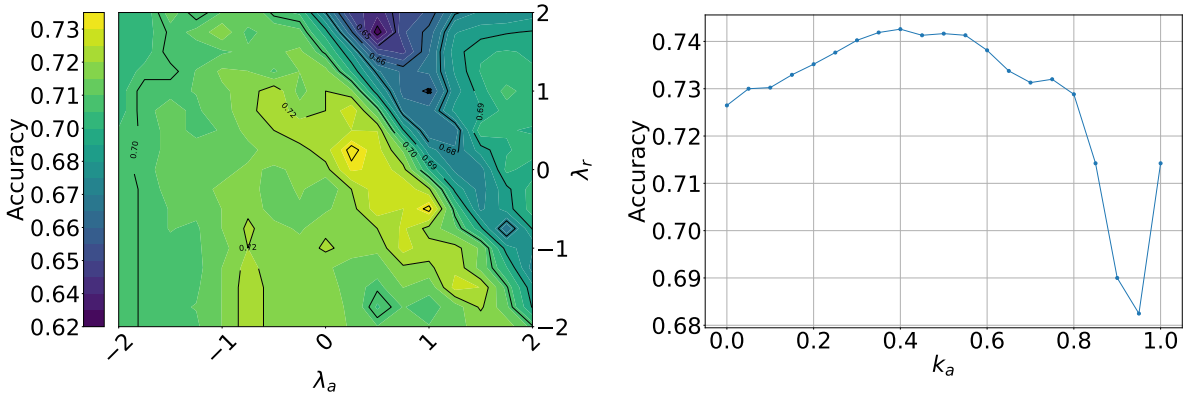



Figure 6.11 – Impact of attractive and repulsive force computation parameters on classification accuracy for subject 12 S2 of COG-BCI dataset. (left) Power parameters $(\lambda_a, \lambda_r)$, (right) scaling parameters $(k_a, k_r = 1 - k_a)$.

Next, to investigate the impact of the scaling parameters, we fixed $\lambda_a = \lambda_r = 0.25$ for the COG-BCI subject and $\lambda_a = 1$, $\lambda_r = 1.25$ for the HG-MI subject. We then varied $k_a$ and $k_r$ under the constraint $k_a + k_r = 1$. The results, shown in [Fig. 6.11] and [Fig. 6.12](right), report the mean classification accuracy across multiple seeds for each value of $k_a$.
For COG-BCI, the accuracy decreases when $k_a = 1$ or $k_a = 0$, *i.e.*, when only one type of force (attractive or repulsive) is considered, highlighting the importance of balancing both forces. In contrast, for HG-MI, performance increases as $k_a$ approaches 1, while the lowest accuracy is observed when $k_a = 0$. This suggests that, in this case, attractive forces contribute more strongly to classification performance.

Overall, these experiments demonstrate that carefully tuning the force parameters on a per-subject basis can lead to noticeable improvements in classification accuracy, and that the relative contribution of attractive and repulsive forces to the classification

performance may vary depending on the task.

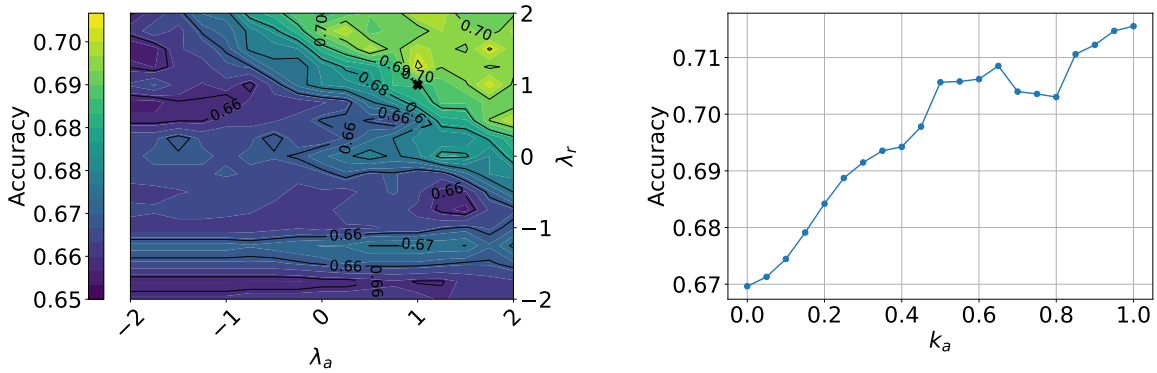



Figure 6.12 – Impact of attractive and repulsive force computation parameters on classification accuracy subject 7 of HG-MI dataset. (left) Power parameters $(\lambda_a, \lambda_r)$, (right) scaling parameters $(k_a, k_r = 1 - k_a)$.

#### 6.3.2.7 Visualization and explainability

In this section, we will analyze the topographies of the force-based features and those of the relative PSD in the theta and alpha bands of the EEG signals, two bands known to be related to mental workload. For visualization, we computed the mean per electrode across all windows of the $\mathsf{u}, \mathsf{v}, \mathsf{k}$ components of the forces with both scaling and power parameters fixed to 1, and then derived the resultant magnitude $\|\mathbf{F}\|$. [Fig. 6.13] shows the corresponding force maps for the first subject, first session of the COG-BCI dataset, a case where classification performance was high. In the same way, we display the mean per electrode of the relative PSD in the theta and alpha frequency bands.

We can observe high magnitudes of the forces, in the right central and temporal regions as well as the left parietal area in the *easy* condition. In the *medium* condition, the magnitude decreases in the left parietal region but increases in the right parietal, extending toward the occipital and right temporal regions, with additional activity between central and frontal sites. In the *difficult* condition, central and frontal activity further increases, while right central and temporal regions show reduced activity.

For the Theta band, activity is present broadly across the scalp, except for the occipital region. In the *easy* condition, it is particularly pronounced over left frontal areas. In the *medium* and *difficult* conditions, activity becomes stronger over frontal and parietal sites, with some extension toward the temporo-occipital regions.

For the Alpha band, the neural activity is more localized in the parietal and occipital lobes, with stronger patterns observed in the frontal area for *easy*, in parietal for *medium* and in occipital for *difficult* conditions.

Each of the topographic maps, depicted above, highlight one specific feature, used for classification. It's important to note that the force topography in a region does not corre-

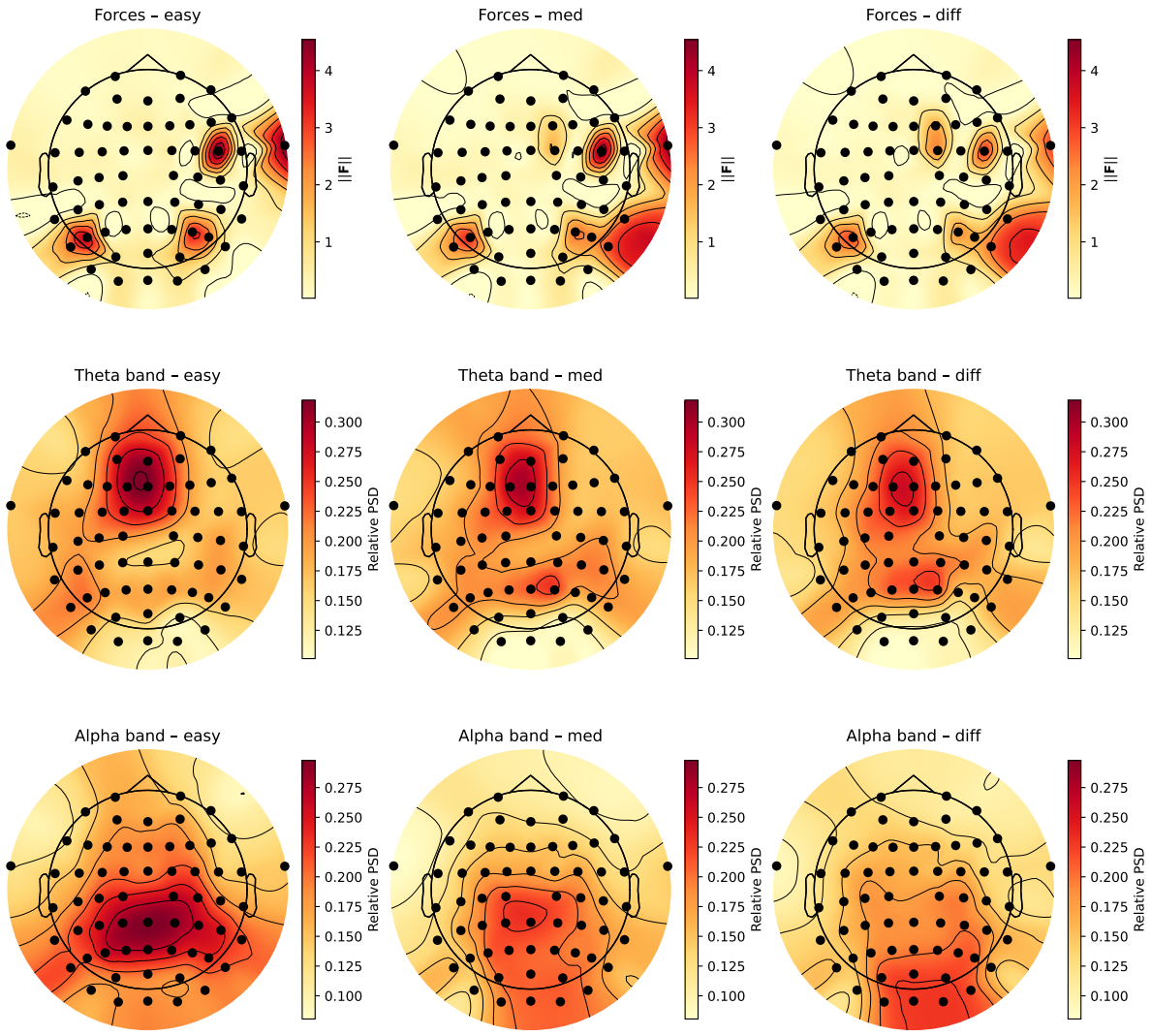



Figure 6.13 – Topographic maps of EEG-derived features for Participant 1, Session 1, across three workload conditions (easy, medium, difficult). The first row shows the spatial distribution of resultant force magnitudes obtained from the force-based graph representation. The second and third rows display the mean spectral energies in the theta and alpha bands, respectively.

spond or directly relate to brain activity in those regions. Instead, the complementarity between the two sources of information, force- and spectral-related, offers valuable insight about the mental workload in this example.

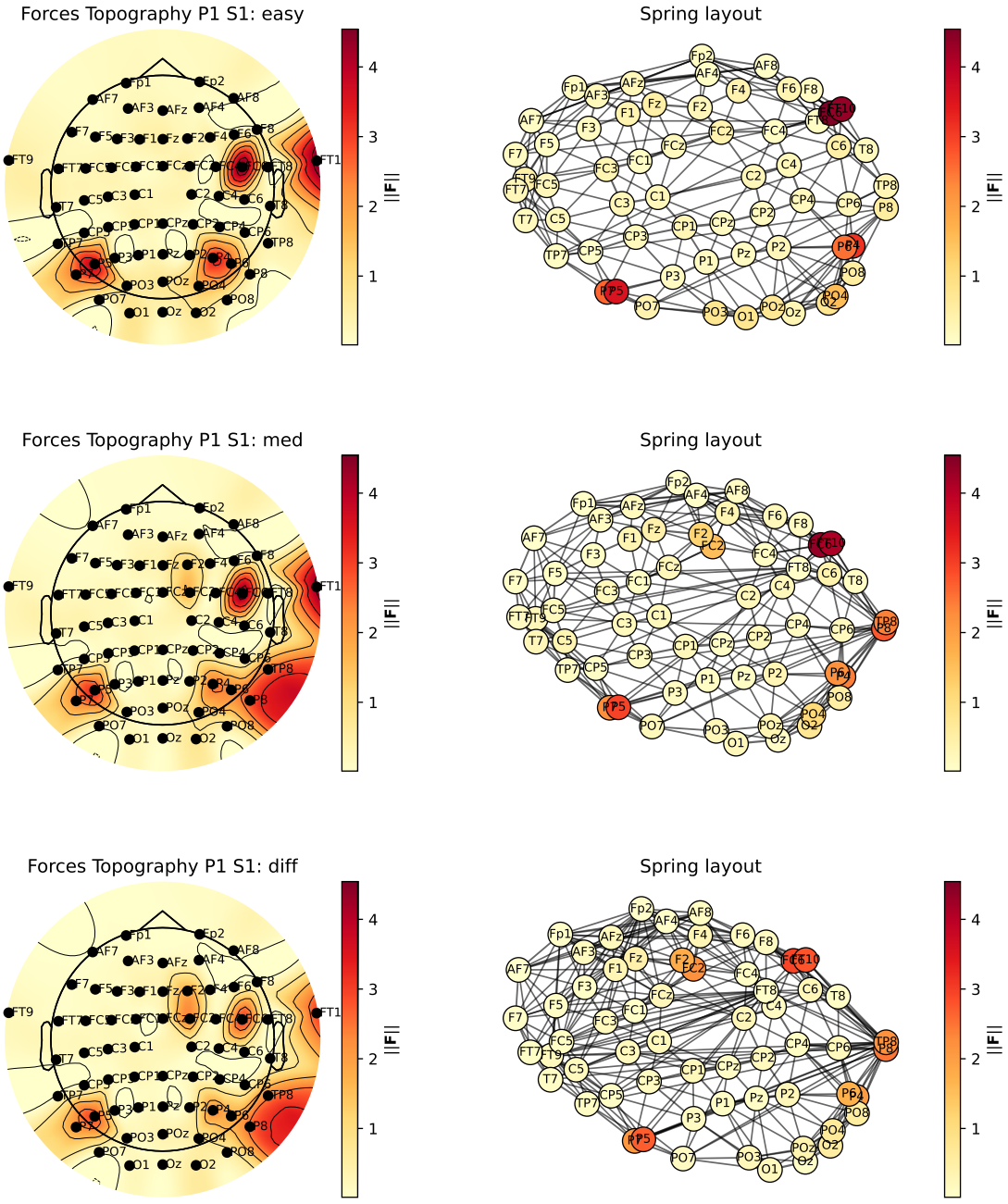



Figure 6.14 – Visualization of force-based features for Participant 1, Session 1, across three workload levels (easy, medium, difficult). (Left) Force topography maps showing the spatial distribution of the force magnitudes $||\mathbf{F}||$ over the scalp. (Right) Corresponding spring layout graph representations, where node color encodes the force magnitude.

[Fig. 6.14] provides a visual comparison between our force-based representation and a conventional force-layout approach. On the left, the force topographies are shown for Participant 1, Session 1 of the COG-BCI dataset, across the three workload levels (easy,

medium, difficult).
On the right, we display the corresponding spring layouts generated using the NetworkX implementation of the method introduced in [Fruchterman et al. 1991], their formulation accounts for edge weights. In contrast to our formulation, where electrode positions remain fixed and only the forces vary, the spring layout adjusts node positions until an equilibrium is reached.

This figure illustrates the analogy between the two approaches: regions where strong connectivity is visible in the force topographies (left) correspond to groups of electrodes that are pulled closer together in the spring layout (right). For example, clusters of higher forces observed in central and parietal regions of the topographies are mirrored in the layout by spatially condensed electrode groups.

## 6.4 Conclusion

In this chapter, we proposed a novel EEG graph force learning framework that integrates both the spatial arrangement of electrodes and a data-driven connectivity matrix. The representation is inspired by physical analogies: attractive forces modeled according to Hooke's law (springs) and repulsive forces modeled according to Coulomb's law (charged particles). We introduced a generalized formulation of forces, in both 2D and 3D settings, and showed that careful tuning of parameters such as distance decay and force scaling can lead to improved classification performance.
We demonstrated the feasibility of force-related node features, when coupled with a classifier such as SVM, to discriminate between different mental states. The proposed approach yielded promising results, outperforming conventional spatial-domain methods for mental workload classification and achieving comparable performance for motor imagery classification. Furthermore, we explored the fusion of these features with EEG signal features and graph-topological measures. While force-based features proved particularly important, combining them with complementary feature types further enhanced classification accuracy.
An ablation study was also conducted to assess the impact of graph structure and classifier choice. The results highlighted the advantage of learning the graph Laplacian from data over relying on handcrafted connectivity measures. Beyond classification, this framework provides visually interpretable and explainable representations, which are valuable for domain experts.
Finally, the study underscored the persistent challenge of generalization in EEG: the effectiveness of an approach strongly depends on the task and exhibits high inter-subject variability.

# Chapter 7

# Brain Connectivity Classification using Graph Neural Networks


7.1 Graph neural networks in EEG analysis . . . 108
7.1.1 Node features in EEG graphs . . . 109
7.1.2 Underlying EEG graph structure . . . 109
7.1.2.1 Connectivity-based graphs. . . . 109
7.1.2.2 Distance-based graphs. . . . 110
7.1.2.3 Hybrid graphs . . . 111
7.1.3 GNN architectures . . . 111
7.1.4 Discussion . . . 112
7.2 Proposed EEG-GNN framework . . . 114
7.2.1 Graph construction strategies . . . 115
7.2.1.1 Connectivity-driven graphs . . . 115
7.2.1.2 Connectivity-aware distance-based graphs . . . 116
7.2.2 Sets of node-specific features . . . 122
7.2.3 Classification using graph neural networks . . . 123
7.3 Experimental results and analysis . . . 124
7.3.1 Experimental setting . . . 125
7.3.2 Connectivity-driven graphs . . . 125
7.3.3 Connectivity-aware distance-based graphs . . . 126
7.3.3.1 2D vs 3D Cartesian distances. . . . 126
7.3.3.2 Electrode projections and geodesic distances. . . . 127
7.3.4 Ablation study on nodes and edge attributes . . . 130
7.3.5 Comparison against baseline pipelines . . . 133
7.4 Conclusion . . . 135

Recently, graph neural networks have emerged as powerful models for EEG classification. At their core, spatial GNNs operate through message passing between k-hop neighbors, taking as input a graph with predefined edges and node feature vectors. Across successive layers, node embeddings are updated through aggregation and message passing. Pooling operations are further applied to derive a compact vector representation of the entire graph.

In this chapter, we build upon our earlier work by comparing alternative graph input representations within the GNN framework for the same classification tasks addressed previously. In particular, we investigate how spatial information can be integrated either as node features (via electrode coordinates) or as edge features (via inter-electrode distances). We also explore different strategies to encode these distances and combine them with signal-derived measures: topological (connectivity-based) or energy-based features. Through this comparative analysis, our goal is to identify graph construction approaches that best leverage the capabilities of GNNs and enable a more comprehensive exploitation of EEG data with GNN. In the following, we will first review the state-of-the-art approaches for graph neural networks applied to EEG signals, then present the proposed framework, our experimental setup and results.

## 7.1 Graph neural networks in EEG analysis

Graph neural networks constitute a powerful class of deep learning models specifically designed to process data structured as graphs, where entities are represented as nodes and their interactions as edges. By leveraging message-passing mechanisms to aggregate information from neighboring nodes, GNNs have demonstrated remarkable performance across diverse domains such as biology [Li et al. 2021], chemistry [Reiser et al. 2022], drug discovery [M. Sun et al. 2020], traffic forecasting [W. Jiang et al. 2022], and others.

In the context of EEG analysis, GNNs have recently gained considerable attention for applications such as emotion recognition, epilepsy detection, and motor imagery classification. Their capacity to jointly exploit node-level features and edge attributes makes them particularly suitable for EEG analysis, where signals can be naturally represented as graphs, with nodes corresponding to electrodes and edges encoding their functional or spatial relationships. In the sequel, we review state-of-the-art approaches for EEG graph classification with GNNs, focusing on three main aspects: (i) the definition of node features (ii) the construction of the underlying graph structure, and (iii) the design of the GNN architecture.

### 7.1.1 Node features in EEG graphs

Several strategies have been proposed to define node features in EEG graphs. The most direct approach is to use the raw EEG signals as node attributes [Shan et al. 2022; Zhdanov et al. 2022].

A more common strategy relies on feature engineering from the recorded signals. Spectral features, such as the PSD computed over different frequency bands, are widely employed [Andac et al. 2022; J. Jia et al. 2022; Klepl, F. He, et al. 2022]. For instance, in [Jang et al. 2018], node features were defined as power and entropy values across eight frequency bands, while [M. Jia et al. 2022] combined diverse features including band energy, Hjorth parameters, higher-order crossings, differential entropy, and fractal dimension. Time-domain and time–frequency features have also been explored. In [Raeisi et al. 2022], node attributes were defined using a rich set of features such as the mean, standard deviation, skewness, and regularity of the smoothed nonlinear energy operator; spike-related measures (number, duration, intervals, and peak values); statistical features of spike correlation; as well as spectral features including harmonic power, relative band power, and the maximum value of the short-time Fourier transform.

Beyond signal-derived features, graph-theoretic measures can also serve as node features highlighting the node importance in the graph. These include centrality-based attributes such as degree, eccentricity, betweenness centrality, and mean connection weight, which quantify the structural importance of each node within the graph [Duong et al. 2019; H. Liu et al. 2022].

Finally, node attributes can also be automatically extracted using neural networks, which learn discriminative representations directly from the data [Y. Liu et al. 2023].

### 7.1.2 Underlying EEG graph structure

The graph structure in EEG-based GNN models is typically defined either from functional connectivity between signals or from physical distances between electrodes. Some studies also propose hybrid approaches combining both.

#### 7.1.2.1 Connectivity-based graphs.

Functional connectivity measures have been widely used to capture statistical dependencies between EEG channels. In [Raeisi et al. 2022], graph structures based on PLV and coherence were compared with Euclidean distance graphs; results showed that functional connectivity yielded superior performance. In [H. Liu et al. 2022], PLI was combined with a minimum spanning tree, ensuring a sparse but connected structure. Similarly, [S. Tang et al. 2021] compared distance-based graphs with correlation-based weighted directed graphs, reporting that the latter more accurately localized focal seizures. Correlation-based graphs have also been coupled with graph attention networks (GAT) in [J. He et al.

2022] in order to automatically learn the spatial dependencies of the graph.

Other works extended the connectivity paradigm by testing multiple measures. For instance, [Shan et al. 2022] compared metrics, including Pearson correlation, coherence, wavelet coherence, PLV, and PLI.

In order to incorporate spectral information into the graph representation, in [Jang et al. 2018], one graph per frequency band was constructed using correlations, with edge weights set to the absolute values. These intra-band graphs were either concatenated or merged via inter-band connections across corresponding electrodes.

More complex feature sets have also been investigated. In [M. Jia et al. 2022], geodesic distances and spectral correlations defined the edges. Their representation was shown to be efficient for wearable biomedical devices while enhancing interpretability of discharge localization.

#### 7.1.2.2 Distance-based graphs.

An important advantage of distance-based graph construction is its independence from the recorded signals, which makes it inherently robust to noise and inter-trial variability. Another benefit of this formulation is that, when combined with spectral GNNs, it enhances interpretability: since the graph structure remains fixed across signals, the Laplacian eigenvectors are also preserved. Furthermore, in [ Salvador et al. 2005], it has been shown, based on the functional MRI, that the strength of connections between brain regions decays approximately with the inverse square of their spatial distance. Building on this principle, several works have adapted only distance-based strategies to EEG graph modeling. In earlier CNN-based approaches [Bashivan et al. 2015, Nogales et al. 2023] (see [Sec.5.1]), electrodes were projected onto a 2D azimuthal map and brain connectivity was defined as the inverse of inter-electrode distance. Extending this strategy to GNNs, the graph topology can be directly derived from electrode positions.

A common strategy is to define edge weights as an exponential decay of Euclidean distance, as in [Tian et al. 2025], where distance-based graphs outperformed other graph structure approaches including correlation, directed transfer function and fully connected for seizure detection. Similarly, [Zhong et al. 2020] encoded edges from 3D electrode distances as inverse of distance squared, with additional global links added for efficiency. In [Xu et al. 2023], distance-based graphs were combined with graph attention Networks to select the most discriminative electrodes, followed by a domain-adversarial strategy to reduce inter-subject variability in sleep stage classification.

Other works have explored unweighted distance-based strategies. In [Demir et al. 2021], fully connected and k-nearest-neighbor graphs were compared, showing that sparse distance-based graphs maintained performance while improving computational efficiency without affecting the classification performance. This work was later extended in [Andac et al. 2022] by learning edge weights with a multi-head attention mechanism from

frequency-based node features. Finally, [Ding, Robinson, et al. 2023] grouped electrodes into spatially local subgraphs assumed to reflect functional brain regions.
Overall, distance-based graphs provide robustness and interpretability by relying on fixed spatial structures, but they may overlook dynamic functional interactions.

#### 7.1.2.3 Hybrid graphs

Hybrid approaches integrating both spatial distances and functional connectivity have been recently explored and offer complementary benefits. For example, [J. Jia et al. 2022] defined the adjacency as the union of inverse squared distances and correlations between eigenvectors of node signals. In [Z. Jia et al. 2021], both spatial and functional graphs were built using the inverse distances and correlation-based similarities, respectively, and integrated within a spatio-temporal GNN for sleep stage classification. [Lin et al. 2021] fused Cartesian distances (after 2D projection) with mutual information, where edges were defined as the union of nearest spatial neighbors and the lowest mutual information values, then the edge weights were set to the differences of coordinates of the connected nodes in the cartesian space.

### 7.1.3 GNN architectures

The core component of any GNN model is the GNN layer, which defines the message-passing process. Broadly, two categories of GNN layers exist: spatial and spectral [Z. Wu et al. 2020]. Spatial GNNs operate directly on the graph structure through the adjacency matrix. Given a set of nodes with associated features, they perform neighborhood aggregation to generate node embeddings, a process commonly referred to as message passing. The underlying principle is that connected nodes should exhibit similar embeddings, reflecting local similarity. In practice, message passing is achieved by updating each node representation with aggregated information from its neighbors. The aggregation layers used in this manuscript are detailed in [Sec. 7.2.3]. Spectral GNNs, on the other hand, are based on spectral graph theory [Bo et al. 2023]. They define convolution operations in the Fourier domain of the graph, where the eigenvectors of the Laplacian act as the Fourier basis. Node features are projected into this spectral domain, filtered through learnable parameters, and subsequently mapped back to the original graph space.

In the context of EEG classification, no single architecture has been universally adopted for all EEG classification tasks. Various studies have explored different architectural choices based on both categories. For example, [Klepl, F. He, et al. 2022] proposed a model with two Graph Convolutional Network (GCN) layers, using max aggregation, followed by a max-pooling operation to obtain a graph-level embedding. This embedding is processed by a dropout layer and two linear layers for classification. Similarly, [M. Jia et al. 2022] employed a deeper model with three GCN layers, one global average pooling

layer, and three fully connected layers. The study of [Demir et al. 2021] compared several spatial GNN layers, including GCN, GraphSAGE (where aggregation is performed via mean pooling of neighboring nodes) and the Graph Isomorphism Network (GIN). Their architecture consisted of two graph convolution layers, followed by a readout layer (sum pooling) and a MLP classifier.

Beyond convolution-based layers, some works rely on (GATs), which extend the message-passing paradigm by learning attention coefficients on edges. For example, [J. He et al. 2022] proposed an architecture with two stacked GAT layers to extract spatial features from the connectivity matrix, followed by a LSTM network to model temporal dependencies prior to classification. In the spectral domain, [Jang et al. 2018] explored Chebyshev polynomial filters, designing a model with four graph convolution layers, two pooling layers, and a fully connected layer. They further investigated the effect of varying the number of filters, polynomial orders, pooling reduction factors, and output dimensions. Furthermore, in [Raeisi et al. 2022], the architecture is based on two graph spectral convolution layers followed with an average pooling along the spatial dimension, followed with three fully connected layers and a softmax.

### 7.1.4 Discussion

In this section, we reviewed several works addressing the construction of EEG graphs at the node level, the edge level, and through different GNN architectures a summary is provided in [Tab. 7.1]. Our contribution focuses primarily on the first two aspects: rather than designing novel GNN architectures, we adopt a relatively simple and fixed architecture, and concentrate instead on modifying the definitions of nodes and edges in order to evaluate their impact on the classification performance.

The central objective of our work is to integrate spatial information from the electrode layout into the constructed graphs and investigate how this information should be encoded, whether at the node level or the edge level. Encoding spatial information on the edges implies computing distances between electrode pairs and using these distances as edge weights. This naturally raises the question of how such distances should be defined. In the literature, when distances are considered, most studies adopt a straightforward strategy: they compute Euclidean distances between electrode coordinates and then apply a decay function, often based on the inverse of distance, to define edge weights. Some comparisons have been made between connectivity-based metrics and distance-based edges, or between hybrid approaches combining connectivity and distance. However, to the best of our knowledge, no prior work has systematically compared different formulations of inter-electrode distances themselves.

Our contribution seeks to address this gap. We propose to investigate multiple ways of computing distances between electrodes, and applying various decay functions to trans-

form these distances into edge weights. This allows us to assess how distance modeling choices influence EEG graph classification's performance, and examining different sets of node features - derived from spectral, spatial, and network domains - and compare their effectiveness.

Table 7.1 – Summary of the reviewed methods for EEG graph construction and GNN modeling.

| Category | Subcategory | Methods |
|---|---|---|
| **Node features** | Time-domain | Mean, standard deviation, skewness, regularity, spike count, spike duration and intervals, Hjorth parameters, higher-order crossings, differential entropy |
| | Spectral-domain | PSD, band energy and entropy, fractal dimension, nonlinear energy operator features |
| | Time–frequency | Maximum short-time Fourier transform value combined with temporal statistics |
| | Graph-structure | Degree, eccentricity, betweenness centrality, mean connection weight, clustering coefficient |
| | Learned | End-to-end learned node embeddings via neural network layers |
| **Graph structure** | Connectivity-based | PC, coherence, wavelet coherence, PLV, PLI with minimum spanning tree, directed transfer function, mutual information, multi-band correlations |
| | Distance-based | Euclidean or 2D azimuthal projection, inverse distance, inverse squared distance, exponential decay of distance, k-NN or fully connected distance graphs, spatial subgraph grouping, attention-learned weights on distances, distance graphs combined with GAT selection |
| | Hybrid | Union of distance and correlation, distance combined with mutual information, spatio-temporal fusion of spatial and functional graphs |
| **GNN architecture** | Spatial | GCN with pooling, GraphSAGE, GIN, GCN with domain adversarial adaptation, GAT with LSTM |
| | Spectral | Chebyshev polynomial filters with pooling, spectral graph convolution with average pooling and MLP classifier |
| | Spatio-temporal | Spatio-temporal GNNs combining spatial and functional graphs, attention-based edge learning with frequency features |

## 7.2 Proposed EEG-GNN framework

Consider a set of EEG signals acquired using a network of $n$ electrodes with known spatial positions $\mathcal{C}$. The goal is to predict a class label from each $t$-second window of these signals. We formulate this as a graph classification problem, where each multivariate EEG window is represented as a graph and assigned a class label. More precisely, the proposed framework consists of the processing steps illustrated in [Fig. 7.1]. An undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ associated with this network is defined where nodes correspond to electrodes, and the weight $\mathbf{W}_{ef}$ on the edge between two nodes $e$ and $f$, following different construction strategies. Node-specific features, emanating from different domains, are extracted to enrich the EEG graph representation. The constructed graph is then processed by a graph neural network, whose output is the predicted class. The details of each of the graph construction strategies, the sets of node features, and the GNN architecture, are detailed below.

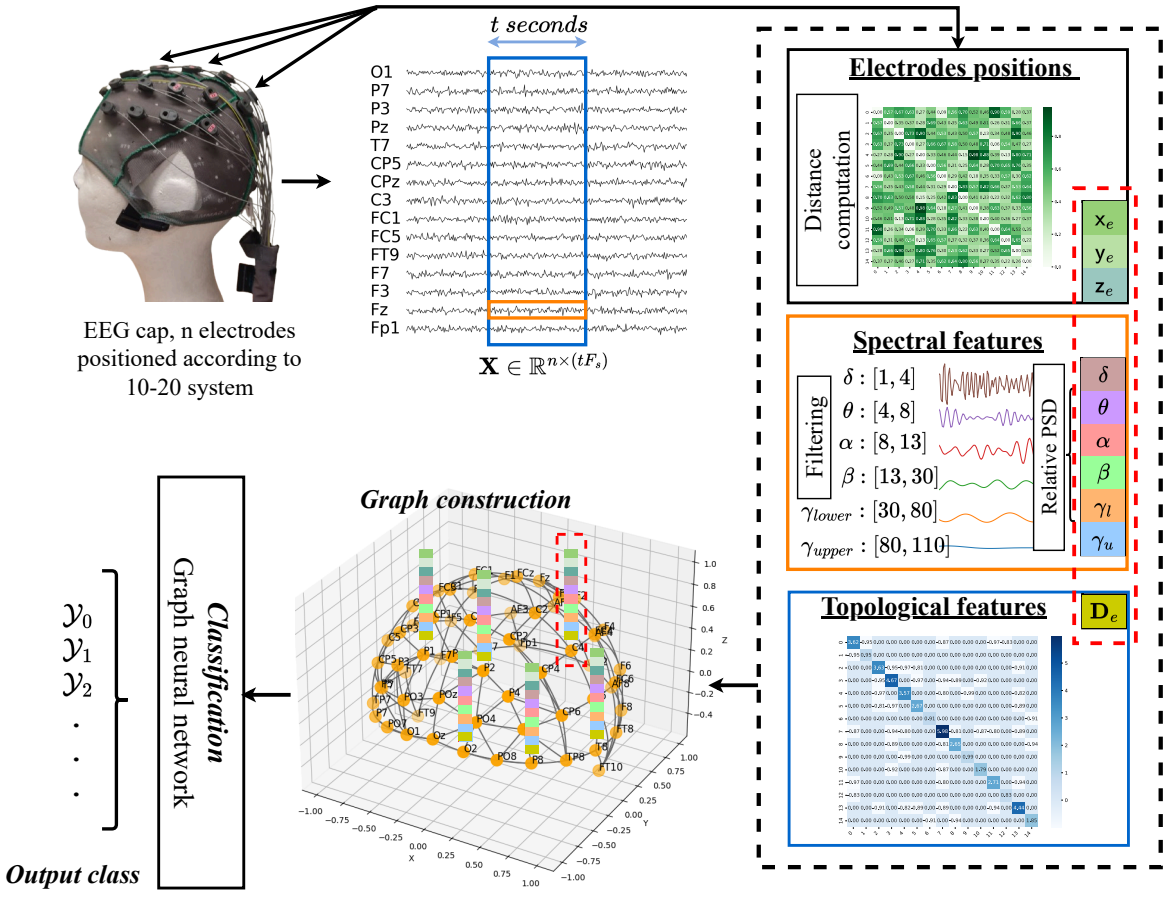



Figure 7.1 – From each EEG window, a connectivity graph is learned. Frequency-based features, possibly combined with spatial positions and a topological feature, are used to define the node attributes. The edges reflect either functional connectivity or the inverse of the geodesic distance. The resulting graphs are then processed by a graph neural network (GNN) for classification.

### 7.2.1 Graph construction strategies

The first step consists of defining the underlying graph structure : the edge weights. The latter are determined using two strategies: one based on connectivity, which can be either handcrafted or learned from data, and another based on the spatial distance between electrodes following different projections.

#### 7.2.1.1 Connectivity-driven graphs

Computing the graph from connectivity metrics can be approached using handcrafted techniques, as described in [Sec. 3.3.1]. These methods estimate pairwise connectivity between electrodes $e$ and $f$ based on their recorded signals. Alternatively, the graph structure can be inferred directly from data, as discussed in [Sec. 3.3.3]. In our experiments, we consider two handcrafted measures that are widely adopted in the literature, PC and PLV, as well as a learned graph construction method under Laplacian constraints. These methods are detailed in the following.

**a) Pearson correlation (PC)** this weighting scheme quantifies the correlation between the signals of two electrodes and is computed as follows:

$$\mathbf{W}_{ef} = \frac{\mathbf{S}_{ef}}{\sqrt{\mathbf{S}_{ee}\mathbf{S}_{ff}}} \tag{7.1}$$

where $\mathbf{S}_{ef}$ denotes the covariance between the EEG signals recorded by electrodes $e$ and $f$. This formulation yields a fully connected graph, which may include negative edge weights, an outcome we aim to avoid. To mitigate this, a thresholding step is applied. While determining an optimal threshold remains an open research problem, we adopt a simple strategy: setting the threshold to the median of all pairwise values computed across the training set graphs.

**b) Phase Locking Value:** this weighting quantifies the coherence of the phase difference between two signals, it is defined by:

$$\mathbf{W}_{ef} = \left| \frac{1}{tF_s} \sum_{t=1}^{tF_s} e^{-\mathrm{i}(\phi_{\mathbf{x}_e(t)} - \phi_{\mathbf{x}_f(t)})} \right| \tag{7.2}$$

where $tF_s$ is the number of samples, and $\mathbf{x}_e(t)$, $\mathbf{x}_f(t)$ are the EEG signals recorded at electrodes $e$ and $f$, respectively. The phase argument $\phi$ is computed via the Hilbert transform [Briels et al. 2020].

**c) Generalized graph Laplacian:** the weight matrix $\mathbf{W}$, in this case, is learned as proposed in [Egilmez et al. 2017] and detailed in [Section 3.3.3]. This approach allows

estimating the underlying structure of the graph, modeled by a precision matrix reflecting the conditional probability of a connection between electrodes $e$ and $f$, from multivariate EEG signals, under Laplacian and connectivity constraints. The computed matrix is a graph Laplacian $\mathbf{L}$. The weight matrix can be deduced from [eq. 3.9] as:

$$\mathbf{W} = \mathrm{Diag}(\mathbf{L}) - \mathbf{L} \tag{7.3}$$

where $\mathrm{Diag}(\mathbf{L})$ denotes a diagonal matrix containing the diagonal elements of $\mathbf{L}$, with all other elements equal to zero.

#### 7.2.1.2 Connectivity-aware distance-based graphs

The second approach for defining the graph structure relies on the spatial arrangement of the electrodes. Our objective is to emphasize connections between spatially proximal electrodes while enforcing sparsity. However, spatial proximity does not necessarily imply functional connectivity. To address this, we retain only the edges for which the connectivity matrix entry $\mathbf{W}_{ef}$ is positive, and assign edge weights which decay with spatial distance. Specifically, the weight between two connected nodes is determined by a decay function $\mathcal{F}(d_{ef})$, which decreases as the distance $d_{ef}$ between nodes $e$ and $f$ increases. The resulting graph structure is encoded in the matrix $\mathbf{R}$, defined as:

$$\mathbf{R}_{ef} = \begin{cases} \mathcal{F}(d_{ef}) & \text{if } \mathbf{W}_{ef} > 0 \text{ and } e \neq f, \\ 0 & \text{otherwise.} \end{cases} \tag{7.4}$$

We investigate three different forms of the decay function $\mathcal{F}(\cdot)$:

— **Inverse decay:** $\mathcal{F}(d_{ef}) = \frac{1}{d_{ef}}$,

— **Exponential decay:** $\mathcal{F}(d_{ef}) = \exp(-d_{ef})$,

— **Gaussian decay:** $\mathcal{F}(d_{ef}) = \exp\left(-\frac{d_{ef}^2}{2\sigma^2}\right)$,

where $\sigma$ denotes the standard deviation of the distance distribution.

**Defining the distance matrix** Consider a set of electrodes and their Cartesian coordinates $\mathcal{C} = \begin{bmatrix} \mathsf{x}, & \mathsf{y}, & \mathsf{z} \end{bmatrix} \in \mathbb{R}^{n\times 3}$, where $\mathsf{x} = [\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n]^\top \in \mathbb{R}^n$. In the graph analysis and EEG literature, there exists no universal way to define the distance matrix between them. A first naive way is to compute Cartesian distances between pairwise electrodes as :

$$d_{ef} = \|\mathbf{c}_e - \mathbf{c}_f\|_2 = \sqrt{(\mathsf{x}_e - \mathsf{x}_f)^2 + (\mathsf{y}_e - \mathsf{y}_f)^2 + (\mathsf{z}_e - \mathsf{z}_f)^2}, \quad \forall e, f \in \{1, \ldots, n\} \tag{7.5}$$

[Fig. 7.2] depicts a sparse graph from the COG-BCI dataset, in which connections are drawn as lines corresponding to the Cartesian distances between electrodes. This visualization shows that by doing so, the true distance on the surface is under-seen. In order to compute the true distance on the surface of the scalp, two alternative approaches are proposed: (1) projecting the electrodes onto the plane and computing a Cartesian distance. (2) projecting the electrodes onto a sphere or an ellipsoid and computing the geodesic distance on the surface.

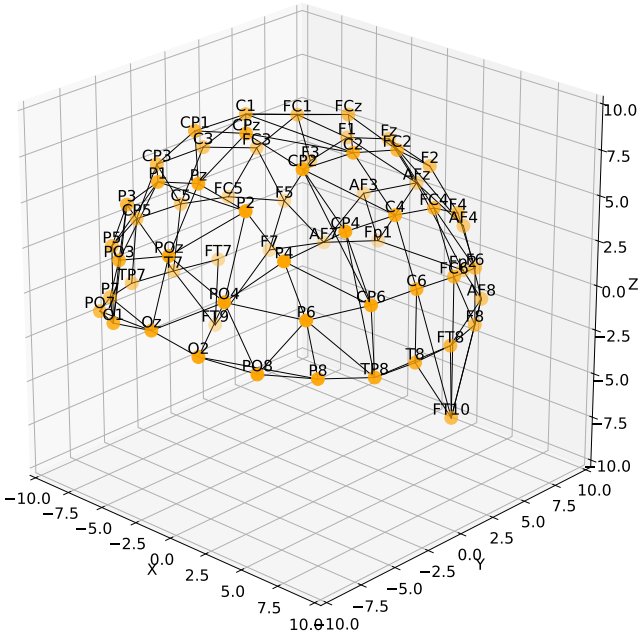



Figure 7.2 – Example of a graph with the original coordinates of the electrodes; connections are drawn as lines corresponding to the Cartesian distances between electrodes

**(1) Projection on a 2D plane, Cartesian distance:** To project the coordinates in $\mathcal{C}$ into the plane, we choose the azimuthal equidistant projection: for each $\mathbf{c}_e \in \mathcal{C}$, the radius $r_e = \|\mathbf{c}_e\|_2$, the azimuth $\bar{\theta}_e = \arctan(\frac{\mathsf{y}_e}{\mathsf{x}_e})$ (angle in the $\mathsf{xy}$, plane); the colatitude (angle from $\mathsf{z}$ downward) is $\bar{\phi}_e = \arccos(\frac{\mathsf{z}_e}{r_e}) \in [0, \pi]$ see [Fig. 7.4]. We project each point on the plane centered at $(0, 0, r_e)$, and compute the radial distance in the 2D projection as:

$$\rho_e = \frac{2r_e\bar{\phi}_e}{\pi} \in [0, 2r_e] \tag{7.6}$$

where $\frac{2r_e}{\pi}$ is a scaling factor to preserve the true distance between electrodes based on their coordinates. The projected coordinates in the plane thus becomes

$$\mathsf{x}_e^{2D} = \rho_e \cos(\bar{\theta}_e) \qquad \mathsf{y}_e^{2D} = \rho_e \sin(\bar{\theta}_e) \tag{7.7}$$

This projection is applied to each point, in order to get the 2D coordinates. This correspond to the projection used in MNE library [Larson et al. 2025, Gramfort et al. 2013] for visualization, the version implemented in the library is not scaled in the same way as in our case. A comparison of the obtained projected electrodes is shown in [Fig.7.3]: right is our projected coordinates; left is the one implemented in MNE library. As we can observe, the electrodes are arranged in a same way.

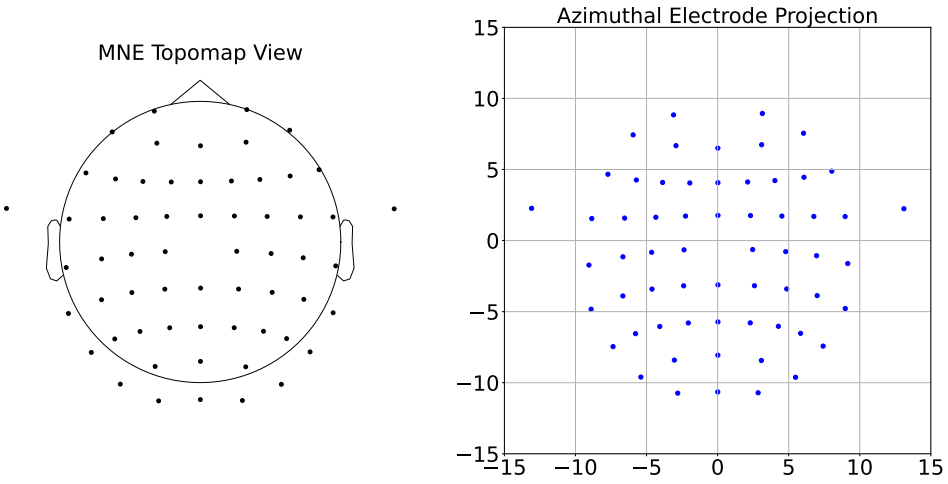



Figure 7.3 – Illustration of the azimuthal equidistant mapping of the 61 electrodes from the COG-BCI dataset. left: conventional MNE-like projection, right: projection generated by our approach.

After projecting the electrodes, the distance is computed as the Cartesian distance between the projected coordinates $\in \mathcal{C}^{2D}$.

**(2) Projection on a sphere, geodesic distance:** An alternative methodology consists in the direct computation of geodesic distances between electrodes within the electrode space. This procedure presupposes that the electrodes are positioned on a well-defined surface. The most tractable configuration corresponds to a spherical surface; accordingly, we initiate the process by projecting the electrode positions onto a sphere of radius $r = \frac{1}{|\mathcal{V}|}\sum_{e\in\mathcal{V}} \|\mathbf{c}_e\|_2$. The coordinates of a projected electrode $e$ become:

$$\bar{\mathbf{c}}_e = r \cdot \frac{\mathbf{c}_e}{\|\mathbf{c}_e\|_2} \tag{7.8}$$

The geodesic distance $d_{ef}$ between two electrodes $e$ and $f$ is defined as the arc length connecting their projections on the sphere, and is computed via:

$$d_{ef} = r \cdot \arccos\left(\frac{\langle \bar{\mathbf{c}}_e, \bar{\mathbf{c}}_f \rangle}{r^2}\right). \tag{7.9}$$

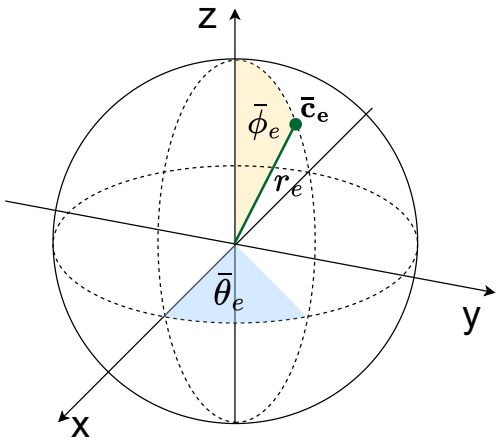



Figure 7.4 – A point on a sphere has coordinates $\bar{\mathbf{c}}_e$. The sphere radius, the azimuth and colatitude are denoted as $r_e$, $\bar{\theta}_e$ and $\bar{\phi}_e$, respectively.

An example of the spherical projection is shown in [Fig. 7.5]. The figure illustrates that the sphere's surface slightly alters the spacing of the electrodes compared to their actual positions, as shown in [Fig. 7.2]. In addition, [Fig. 7.6], which displays the electrode modulo, reveals a non-uniform distribution. This non-uniformity indicates that the electrodes are not equally distant from the center, suggesting that their true positions do not lie exactly on a sphere.

**(3) Projection onto an ellipsoid, geodesic distance:** Given the general shape of the head can be approximated by an ellipsoid [Makris et al. 2008], we investigated the impact of using the geodesic ellipsoidal distances between electrodes. For simplicity and ease of implementation, we chose to model the electrode space as an ellipsoid of revolution, a special case of an ellipsoid where two semi-axes are equal ($\mathsf{a} = \mathsf{b}$). This choice is also supported by previous work on EEG channel interpolation [Courellis et al. 2016], the objective was to recover missing EEG data from close electrodes using interpolation. In their paper, they studied the effect of combining electrodes signals from neighbors based on distance and compared euclidean, spherical and ellipsoid geodesic length and showed that the latter yielded to superior interpolation results.
An **ellipsoid of revolution** [Fig. 7.7], is characterized by two principal axis: the semi-major axis $\mathsf{a}$, corresponding to the equatorial radius, and the semi-minor axis $\mathsf{b}$, corresponding to the polar radius. The *flattening* of the ellipsoid is defined as $\xi = \frac{\mathsf{a}-\mathsf{b}}{\mathsf{a}}$, and the *first eccentricity* is given by $\epsilon = \sqrt{1 - \left(\frac{\mathsf{b}}{\mathsf{a}}\right)^2}$. These parameters describe the ellipsoid's deviation from a perfect sphere. In geodesy, such ellipsoids serve as mathematical models for Earth's shape, enabling the definition of geodetic coordinates: latitude $\varphi$ (*i.e.* the

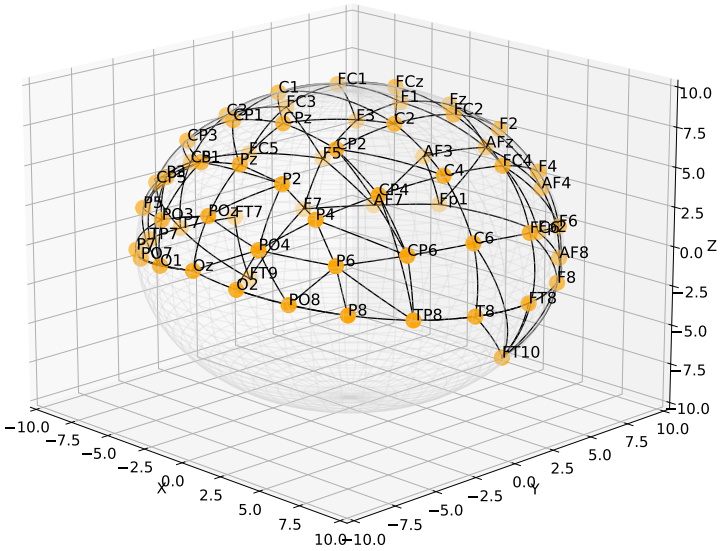



Figure 7.5 – Example of a graph where the electrodes ,*i.e.* the nodes, are projected onto a sphere with a radius $r$.

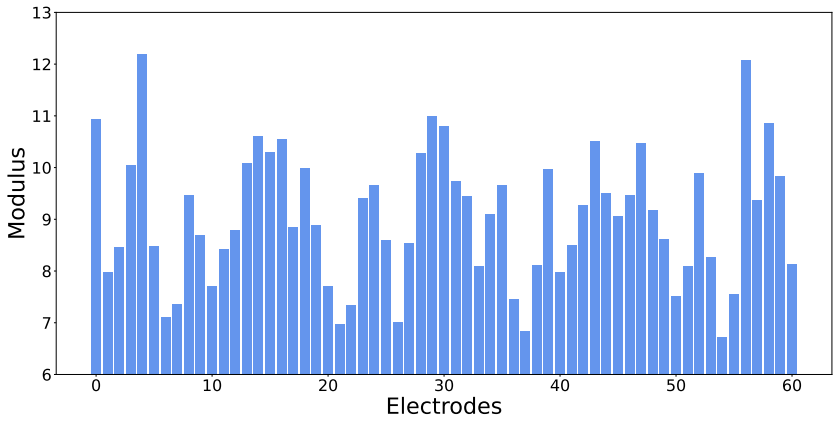



Figure 7.6 – Plot of the Euclidean norms of the electrode positions after centering. The variation in distances suggests that a spherical geometry may not be optimal for representing the head.

angle between the normal vector to the ellipsoid at a point and the equatorial plane), longitude $\Lambda$ the angle between the prime meridian and the meridional plane passing through the point), and height $\mathsf{h}$ (*i.e.* the vertical distance of a point above (or below) a reference surface) relative to the ellipsoid surface.

We fit a ellipse of revolution to the set of electrodes coordinates $\mathcal{C}$ by solving the non linear least squares approximation to find $(\mathsf{x}_0, \mathsf{y}_0, \mathsf{z}_0)$ the coordinates of the center of the

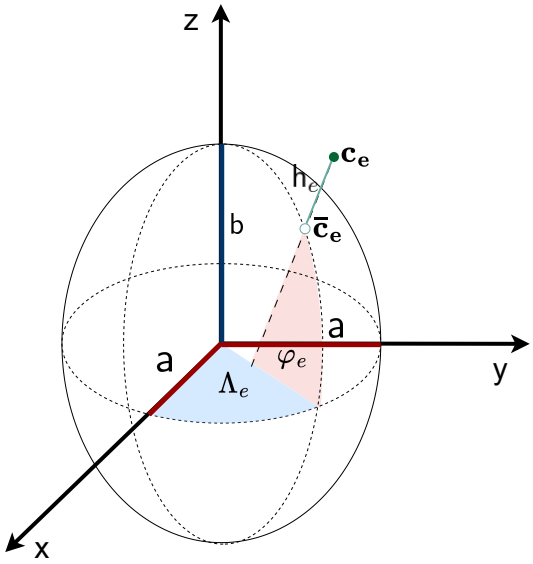



Figure 7.7 – Ellipsoid of revolution showing the semi axis a and b as well as the conversion between the Cartesian coordinates and the ellipsoidal coordinates.

ellipsoid, as well as $a$ and $b$.

$$\min_{x_0,y_0,z_0,a,b} \sum_{i=1}^{n} \left[ \left( \frac{x_i - x_0}{a} \right)^2 + \left( \frac{y_i - y_0}{a} \right)^2 + \left( \frac{z_i - z_0}{b} \right)^2 - 1 \right]^2 \tag{7.10}$$

Once the ellipsoid is found, we approximate the longitude, latitude and height of the closest points on the ellipsoid for each corresponding Cartesian coordinates using the analytic method proposed by [Vermeille 2002] in which :

$$\Lambda = 2 \arctan \left( \frac{y}{x + \sqrt{x^2 + y^2}} \right) \tag{7.11}$$

$$\varphi = 2 \arctan \left( \frac{z}{q + \sqrt{q^2 + z^2}} \right) \tag{7.12}$$

$$h = \left( \frac{\kappa + \epsilon^2 - 1}{\kappa} \right) \sqrt{q^2 + z^2} \tag{7.13}$$

where $q = \frac{\kappa \sqrt{x^2+y^2}}{\kappa+\epsilon^2}$; $\kappa$ is a parameter, and $\epsilon$ is the eccentricity. For a detailed description of this method, the reader can refer to the paper [Vermeille 2002]. Finally, the latitudes and longitudes are used to approximate the the geodesic distances on the ellipsoid between the points using the method of Karney's algorithm [Karney 2013]. The latter has been first conceived for the earth and geodesic distances between locations, and could be applied to any ellipsoid of revolution. Karney solves the geodesic distance problem between two

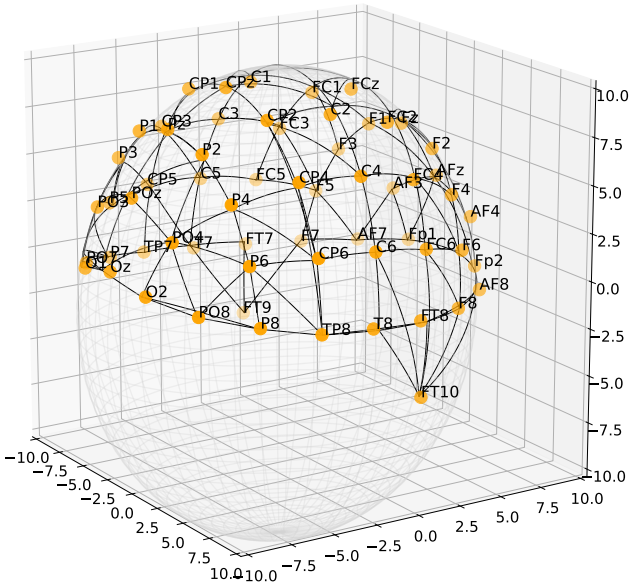



Figure 7.8 – Example of a graph with electrode positions projected onto an ellipsoid fitted to the points using mean square error approach.

points knowing their latitudes and longitudes using a reduction to the auxiliary sphere followed by a series expansion to compute the final distance. For more details on this approximation, the reader is referred to [Karney 2013].
We visualize in [Fig. 7.8] the projection of the electrodes on a fitted ellipsoid of revolution using the least square method from the COG-BCI dataset. Compared to [Fig. 7.2 and 7.5] we can see that the shape of the electrode space is more accurate using the ellipsoid of revolution projection.

### 7.2.2 Sets of node-specific features

With the connections and weights now established, we further enhance this graph representation by assigning feature vectors to all the vertices as follows. Each node of the graph is described by a feature vector that can contain various types of information. In our comparative study and analysis, we consider the three feature types used in chapter 6: the spatial positions of the electrodes, topological features derived from the functional connectivity, and spectral features extracted directly from the signals.

1. **Topological features** To quantify the relative importance of a node in the graph, we use its weighted degree as a topological measure, which corresponds to the sum of the weights of the connections linking it to other nodes. This value, simple yet informative, reflects the structural connectivity of each electrode and its importance

within the network. Note that in contrast to chapter 6 here we use only the degree as topological feature.

2. **Spectral features** The relative PSD is used due to its proven effectiveness in EEG classification tasks [ Borghini et al. 2014; Klepl, F. He, et al. 2022]. For each time window, the PSD is estimated for each electrode using Welch's method, across the following frequency bands: $\delta^{EEG}$ [1–4 Hz], $\theta^{EEG}$ [4–8 Hz], $\alpha^{EEG}$ [8–13 Hz], $\beta^{EEG}$ [13–30 Hz], $\gamma^{EEG}_{\text{low}}$ [30–80 Hz], and $\gamma^{EEG}_{\text{high}}$ [80–110 Hz]. The power in these bands is then normalized by the total power [1–110 Hz], allowing the relative contribution of each frequency band to be expressed.
3. **Electrode positions** Beyond the brain activity itself, the spatial location of this activity provides complementary and relevant information. Therefore, the three-dimensional Cartesian coordinates $(\mathsf{x}, \mathsf{y}, \mathsf{z})$ of each electrode can be included as positional features at the node level.

At the end of this step, $\bar{\mathbf{H}}^{(0)} \in \mathbb{R}^{n \times k}$ represents the feature matrix of the $n$ electrodes. Each row of $\bar{\mathbf{H}}^{(0)}$, denoted by $\bar{\mathbf{h}}_e^{(0)}$, corresponds to the initial feature vector of a node, with dimension $k$. After constructing the graph representation, incorporating both node features and edge weights, the data is input into a GNN model to predict the output class.

### 7.2.3 Classification using graph neural networks

Before introducing the graph neural network architecture adopted in our comparative study, we start by defining a simple spatial GNN layer based on message passing *GraphConv*.

**Graph Convolution layer (*GraphConv*)** proposed by [Morris et al. 2019], it performs two key steps:

1. It first aggregates messages from the 1-hop neighbors of a node $e$, $\mathcal{N}_1(e)$, where each neighbor's feature vector is weighted by the corresponding edge weight before being summed.
2. It then adds a separately transformed version of the node's own features, allowing the model to treat the node's message differently from the incoming messages of its neighbors.

This separation is achieved by learning two distinct transformation matrices, $\mathbf{\Theta}_1$ for the node's own features and $\mathbf{\Theta}_2$ for its neighbors. Formally, the feature vector $\bar{\mathbf{h}}_e$ of node $e$ is updated between layers $(l-1)$ and $(l)$ as:

$$\bar{\mathbf{h}}_e^{(l)} = \mathbf{\Theta}_1 \bar{\mathbf{h}}_e^{(l-1)} + \mathbf{\Theta}_2 \Psi_{f \in \mathcal{N}_1(e)}(\mathbf{W}_{ef} \cdot \bar{\mathbf{h}}_f^{(l-1)}), \tag{7.14}$$

where $\mathbf{W}_{ef}$ denotes the edge weight between node $e$ and its neighbor $f$, and $\bar{\mathbf{h}}_f$ is the feature vector of neighbor $f$ .
The architecture of the GNN, shown in [Fig. 7.9], is fixed in all the experiments of this chapter. It consists of two Graph Convolution layers (*GraphConv*) layers, allowing it to incorporate information from two-hop neighborhood. To obtain a global representation of the graph, the node embeddings are aggregated through a readout layer by taking the maximum, for each feature dimension, across all nodes:

$$\bar{\mathbf{h}}_{\mathcal{G}}(k) = \max_{e} \bar{\mathbf{h}}_e(k), \quad k = 1, \ldots, d. \tag{7.15}$$

A *dropout* operation is then applied to the graph embedding, with a given probability, in order to randomly deactivate certain elements and improve the model's generalization ability. Finally, a fully connected linear layer is used to perform the classification task based on the graph embedding.

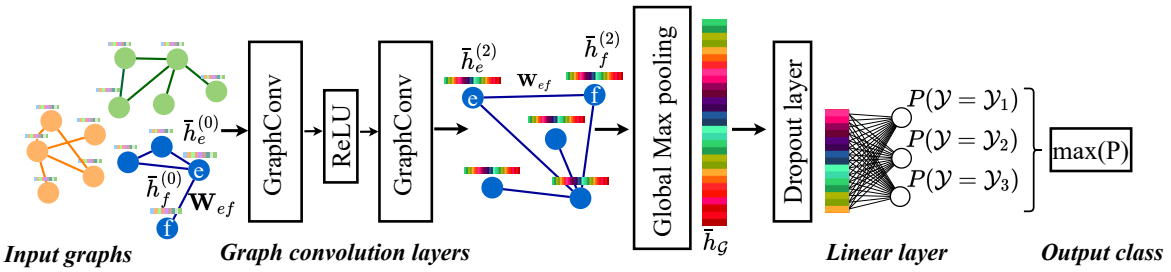



Figure 7.9 – GNN model architecture consisting of two GraphConv layers, a global max-pooling layer, a dropout and linear layer.

The GNN model is trained in a supervised manner by minimizing the standard categorical cross-entropy loss between the predicted class probabilities and the ground-truth labels. Optimization is performed using the Adam optimizer.

## 7.3 Experimental results and analysis

After introducing the methods used to construct the graphs, both at the node level and the edge level, where edges encode either functional connectivity or spatial distances, and nodes contains spectral, structural and topological features, we now present the experimental results and their comparative analysis. We begin by comparing the graph-construction strategies based on functional connectivity. Next, we examine the distance-based graph structures, evaluating the different decay functions and electrode projection schemes introduced earlier. Finally, we focus on the node-level features and assess the various possible combinations in this context. The detailed results and analysis are provided in the following sections.

### 7.3.1 Experimental setting

All experiments are conducted in an *intra-session classification* setting. Since the aim of this study is to compare the impact of different node and edge feature configurations in graph-based inputs, we use the same set of GNN hyperparameters across all tests. These hyperparameters were determined through a grid search on a randomly selected subject from the dataset. To ensure generalizability, additional combinations were tested on multiple subjects before finalizing the configuration.
The selected hyperparameters are chosen as follows: a learning rate of $10^{-3}$ for mental workload classification and $10^{-2}$ for motor imagery classification, a batch size of 30, a dropout probability of 0.5, and a hidden layer size of 65 channels. Model training is governed by an early stopping criterion with a maximum of 2000 epochs and a patience of 100 epochs.
For the COG-BCI dataset, EEG signals are segmented into 2-second windows with 1-second overlap, grouped into window sets as described in [Sec. 4.1]. For the HG-MI dataset, windows are 4 seconds long without overlap, as detailed in [Sec. 4.2].
The performance of the models is evaluated separately for each session and each subject using *5-fold cross-validation*, and the results reported correspond to the average accuracy across all sessions and subjects. Train-test splits are kept identical across all the tests for each dataset to ensure a fair comparison between the different approaches with 80% of the windows are using for training, among which 10% are used for validation and the rest for the test.

### 7.3.2 Connectivity-driven graphs

We compare different strategies for constructing graph structures from connectivity measures derived from EEG signals, focusing on three approaches: the generalized graph Laplacian, Phase Locking Value(PLV), and Pearson correlation(PC). Each resulting graph is then enriched with node features formed by the concatenation of positional (Cartesian coordinates), topological (node degrees), and spectral (relative power spectral density across six frequency bands) attributes, as detailed in [Sec. 7.2.2].

The results presented in [Tab. 7.2] indicate that learning a sparse graph from EEG signals using the generalized Graph Laplacian (GGL) yields the best performance across both datasets. On the COG-BCI dataset, it achieves an accuracy of $0.84 \pm 0.10$, clearly outperforming both Pearson correlation $0.76 \pm 0.08$ and Phase Locking Value $0.73 \pm 0.07$. A similar trend is observed on the HG-MI dataset, where the generalized Laplacian reaches $0.68 \pm 0.08$ while PLV and Pearson correlation, perform equally at $0.61 \pm 0.09$.

When comparing PLV and Pearson correlation directly, PC shows a slight advantage on

the COG-BCI dataset, suggesting that linear dependencies may be more informative than phase-based synchronization in this context. However, for the HG-MI dataset, both measures perform equally, indicating no clear advantage of one over the other in capturing task-relevant brain dynamics.

Importantly, both PLV and Pearson correlation are limited to modeling pairwise relationships and require arbitrary thresholding to induce sparsity in the resulting graph. In contrast, the generalized graph Laplacian based on the precision matrix, not only captures conditional dependencies among nodes but also enforces sparsity intrinsically, without the need for predefined thresholds. Consequently, the Laplacian-based graph construction method will be adopted for the remainder of this chapter.

Table 7.2 – Comparison of connectivity-driven graph construction strategies (mean ± std).

| **Dataset** | **Node features** | **Graph structure** | **Accuracy** |
|---|---|---|---|
| COG-BCI | Topological \|\| spectral \|\| positional | PLV | $0.73 \pm 0.07$ |
| | | PC | $0.76 \pm 0.08$ |
| | | GGL | $\mathbf{0.84 \pm 0.10}$ |
| HG-MI | Topological \|\| spectral \|\| positional | PLV | $0.61 \pm 0.09$ |
| | | PC | $0.61 \pm 0.09$ |
| | | GGL | $\mathbf{0.68 \pm 0.08}$ |

### 7.3.3 Connectivity-aware distance-based graphs

Having evaluated the effect of incorporating electrode positions as node features, we now investigate the impact of encoding this spatial information directly into the graph structure through distance-based edge weighting. In this setting, node features are limited to the concatenation of topological attributes (*i.e* weighted node degrees derived from the connectivity matrix) and spectral features (*i.e* relative power spectral density across the six frequency bands). The graph structure, on the other hand, is constructed using the distance-based methods described in [Section 7.2.1.2], for the ellipsoid, we use the classes *EcefKarney*, *Ellipsoid* and *Geodesic* of the *pygeodesy* python library [Brouwers 2016].

#### 7.3.3.1 2D vs 3D Cartesian distances.

We begin by comparing the classification performance obtained using distance matrices computed from 3D Cartesian coordinates [Fig. 7.2] versus those derived from 2D projections of the electrodes onto the azimuthal plane, following the convention used in the MNE library [Fig. 7.3]. We also analyzed the three decay functions detailed in

[Sec. 7.2.1.2]. Results are summarized in [Tab. 7.3].

**For COG-BCI dataset**, which involves a three-class classification task, both 2D and 3D spatial representations yield competitive results. The highest accuracy is obtained using the 2D projection combined with either the inverse decay function or the Gaussian decay function, both achieving $0.78 \pm 0.08$. This outcome is expected, as direct Cartesian distances in 3D do not reflect the true geodesic paths along the scalp surface, potentially leading to distorted spatial relationships between electrodes. Interestingly, the Gaussian decay consistently outperforms the exponential decay in both 2D and 3D settings. This suggests that sharper decay profiles may better emphasize local interactions that are relevant for mental workload classification.

**For the HG-MI dataset**, which involves a four-class classification task, the overall performance is lower across all configurations. In addition, both the inverse and the Gaussian decay functions again lead to the best results, with accuracies up to $0.62 \pm 0.09/0.10$. These findings further support the use of stronger decay functions, which more effectively penalize distant connections compared to the exponential decay. This effect can also be observed in [Fig. 7.10, Fig. 7.11] where we show the histograms of inter-electrode distances for both 2D and 3D cases, along with the corresponding decay functions. As shown, the distribution of distances differs between the two representations. The exponential decay function drops sharply for distances greater than approximately 5 cm, while the distance distribution is centered around 15 cm. This mismatch may explain the lower performance of the exponential decay, as it excessively suppresses a large portion of meaningful connections.

#### 7.3.3.2 Electrode projections and geodesic distances.

Going beyond Euclidean distances, we now compare strategies for expressing distances based on the geodesics on the scalp surface. This approach accounts for the true curvature of the scalp rather than assuming straight-line (Euclidean) connections between electrodes. To this end, we project the electrode space onto two known geometric surfaces: a sphere of radius $r$, and a revolution ellipsoid. Results are summarized in [Tab. 7.4].

**For the COG-BCI dataset,** both geodesic projections (sphere and ellipsoid) yield comparable performance. The inverse decay function results in slightly better accuracy on the sphere than on the ellipsoid, with a difference of approximately 0.01.

**For the HG-MI dataset,** in contrast, the projection on a sphere leads to a lower accuracy when used with the exponential and inverse decay functions, and only matches the ellipsoid's performance under the Gaussian decay function.

Table 7.3 – Classification accuracy (mean ± std) using different distance decay functions and spatial representations (2D vs. 3D Cartesian). Node features are fixed to topological and spectral features.

| **Dataset** | **Node features** | **Distance method** | **Decay function** | **Accuracy** |
|---|---|---|---|---|
| COG-BCI | Topological \|\| spectral | 3D Cartesian | $\exp(-d)$ | $0.69 \pm 0.08$ |
| | | | $1/d$ | $0.75 \pm 0.09$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.77 \pm 0.08}$ |
| | | 2D Cartesian | $\exp(-d)$ | $0.73 \pm 0.09$ |
| | | | $1/d$ | $\mathbf{0.78 \pm 0.08}$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.78 \pm 0.08}$ |
| HG-MI | Topological \|\| spectral | 3D Cartesian | $\exp(-d)$ | $0.55 \pm 0.07$ |
| | | | $1/d$ | $0.60 \pm 0.09$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.62 \pm 0.09}$ |
| | | 2D Cartesian | $\exp(-d)$ | $0.60 \pm 0.09$ |
| | | | $1/d$ | $\mathbf{0.62 \pm 0.09}$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.62 \pm 0.09}$ |

To evaluate the added value of geodesic over Cartesian distances, we project the electrodes into the azimuthal plane of the sphere, ensuring comparability by fixing the projection radius $r$ to match that of the sphere in [Eq. 7.6]. Results show that for both datasets, this 2D Cartesian representation (*Sphere+2D*) outperforms the purely geodesic sphere projections, suggesting it may be a more effective way to encode distances.

Across all projection types and datasets, the exponential decay function consistently produces the lowest classification accuracy. In contrast, the inverse decay function frequently leads to the best results, particularly for the COG-BCI dataset. The Gaussian decay function demonstrates stable and robust performance across all settings. Overall, projecting electrodes into the azimuthal plane (*Sphere+2D*) appears to be the most effective strategy, despite the relatively close performance across projection types.

**Discussion:** One limitation of the projection schemes arises, when the considered electrodes cover a limited region of the scalp. In the case of the HG-MI dataset, only 44 electrodes are used, primarily covering the cortical region. This limited spatial support underconstrains surface fitting, particularly for the ellipsoid of revolution model, so the fitted surface may deviate from the true head geometry. To mitigate this bias, we leverage

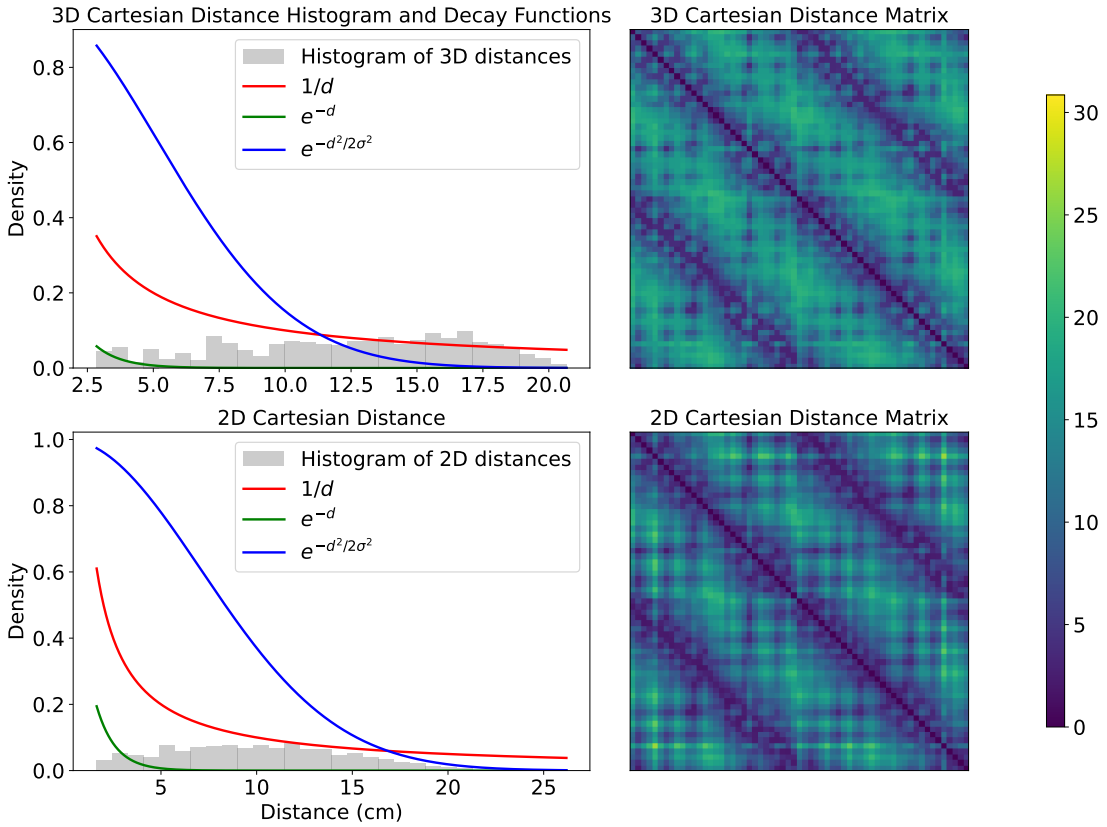



Figure 7.10 – Pairwise Euclidean distances in 3D vs. 2D Cartesian coordinates on the COG-BCI dataset. Left panels: histograms of distances with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right panels: corresponding distance matrices displayed with a shared color scale; the colorbar indicates $d$ in cm

the fact that the original recordings were acquired with a 128-channel cap: we first fit the ellipsoid using all available electrodes and then compute pairwise distances only on the selected subset of 44 channel. This can be observed in [Fig. 7.14] where we depict the results of the different projection schemes used to compute the distances described above.

This limitation is less prominent in the COG-BCI dataset, which includes 61 electrodes distributed across the entire scalp, allowing for more reliable surface fitting. The difference between the distance matrices of the two datasets is reflected in [Fig. 7.12, 7.13] as well as the decay functions in each case. In particular, the matrices in [Fig. 7.13] exhibit block-like structures that reflect spatial subregions captured by the 44-electrode subset, whereas [Fig. 7.12] shows a more uniformly sampled geometry.

Interestingly, the classification results based on pairwise distances appear to be relatively close across the different projection types and across both datasets. This suggests that the model may be primarily leveraging relative spatial relationships between electrodes rather than absolute distance magnitudes. Performance differences across settings are likely

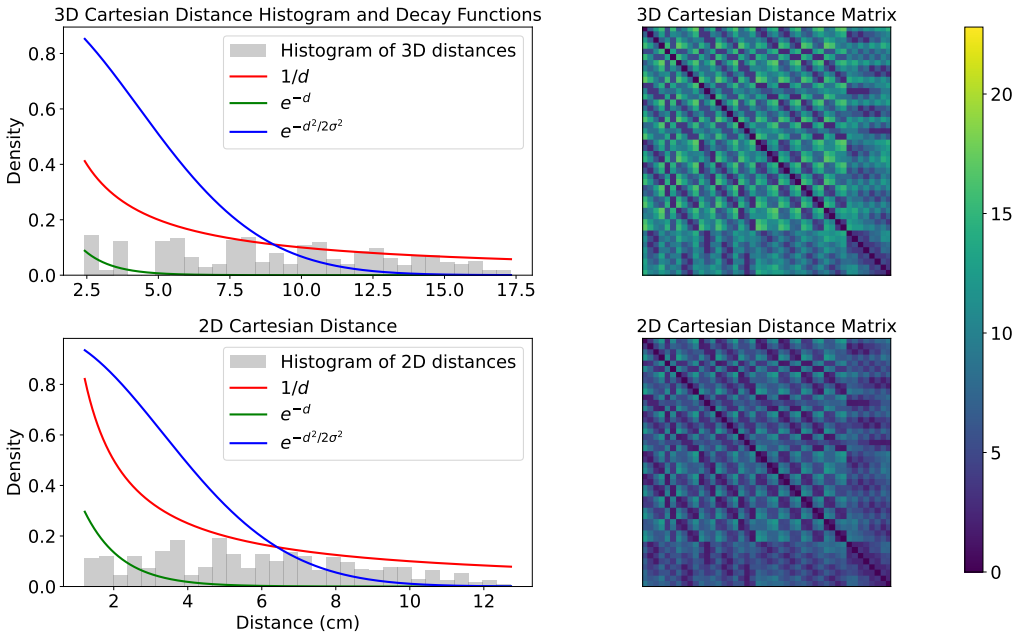



Figure 7.11 – Pairwise Euclidean distances in 3D vs. 2D Cartesian coordinates on the HG-MI dataset. Left panels: histograms of distances with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right panels: corresponding distance matrices displayed with a shared color scale; the colorbar indicates $d$ in cm

influenced by the sparsity patterns imposed by the decay functions and the corresponding connectivity weight matrices. Notably, when comparing distance-based edge encoding [Tab. 7.3, 7.4] with an alternative strategy that uses electrode positions as node attributes and functional connectivity as edge attributes [Tab. 7.2], the latter approach yields higher classification accuracy. This further indicates that encoding richer spatial and functional information rather than relying solely on geometric distances can enhance classification performance.

### 7.3.4 Ablation study on nodes and edge attributes

To assess the individual contribution of each feature type and the impact of different spatial and functional encoding strategies, we compare various graph configurations by varying both the node and edge attributes. For distance-based graphs (denoted as **R**), we fix the projection and decay function to the best-performing one identified earlier, that is the Gaussian decay applied to Cartesian distances after azimuthal projection. In this setting, we evaluate each node feature type independently, excluding the 3D spatial coordinates, as spatial information is already embedded in the graph structure through the distance-based weights.

For connectivity-based graphs where edge features are defined based on functional con-

Table 7.4 – Classification accuracy (mean ± std) using different distance decay functions applied to geodesic distances under various projection types. Node features are fixed to topological and spectral features.

| Dataset | Node features | Projection | Decay function | Accuracy |
|---|---|---|---|---|
| COG-BCI | Topological \|\| spectral | Sphere | $\exp(-d)$ | $0.72 \pm 0.09$ |
| | | | $1/d$ | $0.76 \pm 0.09$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.77 \pm 0.08}$ |
| | | Sphere+2D | $\exp(-d)$ | $0.73 \pm 0.09$ |
| | | | $1/d$ | $\mathbf{0.78 \pm 0.09}$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $0.77 \pm 0.08$ |
| | | Ellipsoid | $\exp(-d)$ | $0.71 \pm 0.09$ |
| | | | $1/d$ | $0.76 \pm 0.09$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.77 \pm 0.08}$ |
| HG-MI | Topological \|\| spectral | Sphere | $\exp(-d)$ | $0.55 \pm 0.07$ |
| | | | $1/d$ | $0.59 \pm 0.08$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.62 \pm 0.09}$ |
| | | Sphere+2D | $\exp(-d)$ | $0.58 \pm 0.08$ |
| | | | $1/d$ | $0.61 \pm 0.09$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.62 \pm 0.09}$ |
| | | Ellipsoid | $\exp(-d)$ | $0.57 \pm 0.09$ |
| | | | $1/d$ | $0.61 \pm 0.09$ |
| | | | $\exp\left(-\frac{d^2}{2\sigma^2}\right)$ | $\mathbf{0.62 \pm 0.09}$ |

nectivity (denoted as $\mathbf{W}$), we explore different combinations of node attributes, including the spatial coordinates. However, we do not consider spatial coordinates *alone* as node features in this case, since they remain constant across epochs, sessions, and classes, and thus provide no discriminative value on their own. Instead, they are combined with other feature types (e.g., topological or spectral). The results of the experiments, presented in [Tab. 7.5], reveal several key findings:

— Encoding functional connectivity as edge weights provides more informative graph structures than encoding based solely on distances.

— Spectral features (i.e PSD) alone contribute little to classification performance.

— Topological features (i.e node degree), either used alone or combined with spatial coordinates, lead to higher classification accuracy, even outperforming combinations that include spectral features.

— Including PSD in the full set of features does not improve, and may slightly de-

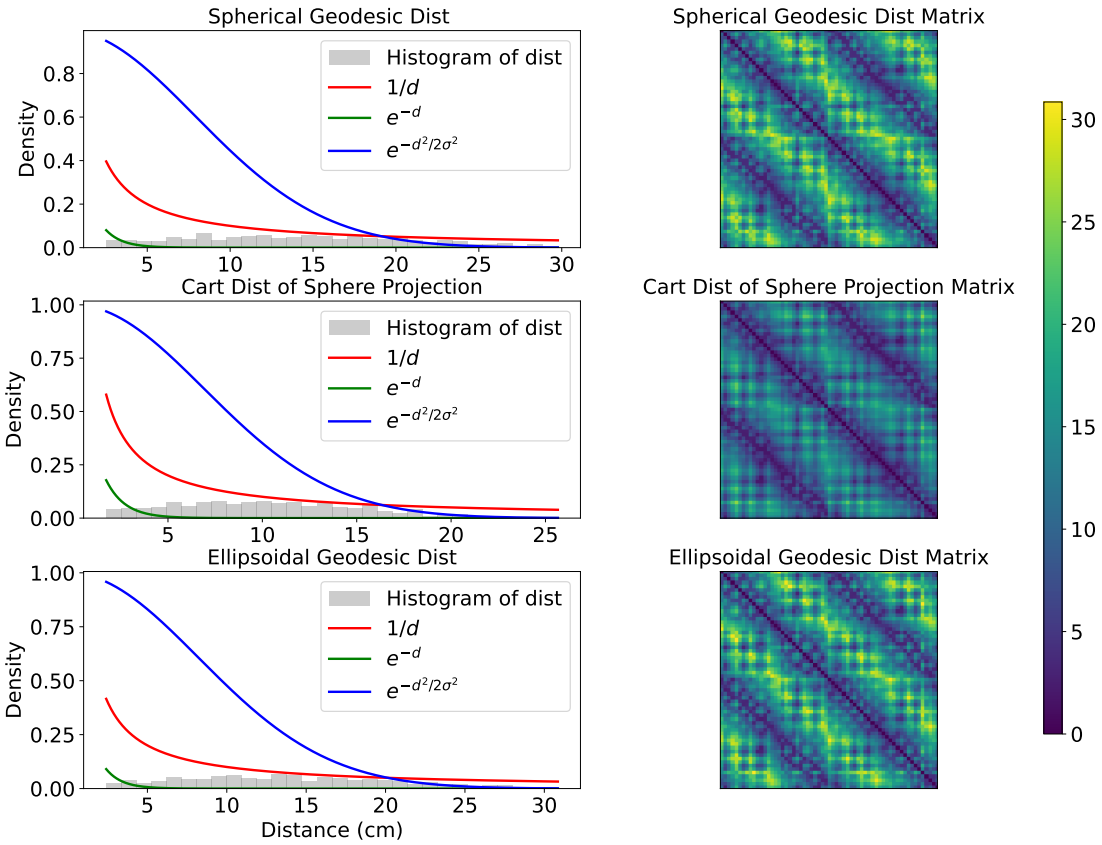



Figure 7.12 – Spherical/ellipsoidal geodesic distances and sphere-projected Cartesian distances on the COG-BCI dataset. Left column: histograms of pairwise distances $d$ with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right column: corresponding distance matrices displayed with a shared color scale across rows; the colorbar indicates $d$ in cm.

grade performance in case of mental workload, suggesting that spectral power is not a relevant feature for this classification task but is useful for motor imagery classification.

Table 7.5 – Comparison of node and edge attribute configurations (mean ± std).

| **Edge features** | **Node features** | **COG-BCI** | **HG-MI** |
|---|---|---|---|
| **R** | Spectral | $0.71 \pm 0.09$ | $0.50 \pm 0.06$ |
| | Topological | $0.74 \pm 0.13$ | $0.55 \pm 0.08$ |
| | Spectral \|\| Topological | $0.78 \pm 0.08$ | $0.62 \pm 0.09$ |
| **W** | Spectral | $0.69 \pm 0.08$ | $0.56 \pm 0.06$ |
| | Position \|\| Topological | $\mathbf{0.88 \pm 0.09}$ | $0.65 \pm 0.08$ |
| | Position \|\| Spectral | $0.84 \pm 0.10$ | $0.66 \pm 0.08$ |
| | Position \|\| Topological \|\| Spectral | $0.84 \pm 0.10$ | $\mathbf{0.68 \pm 0.08}$ |

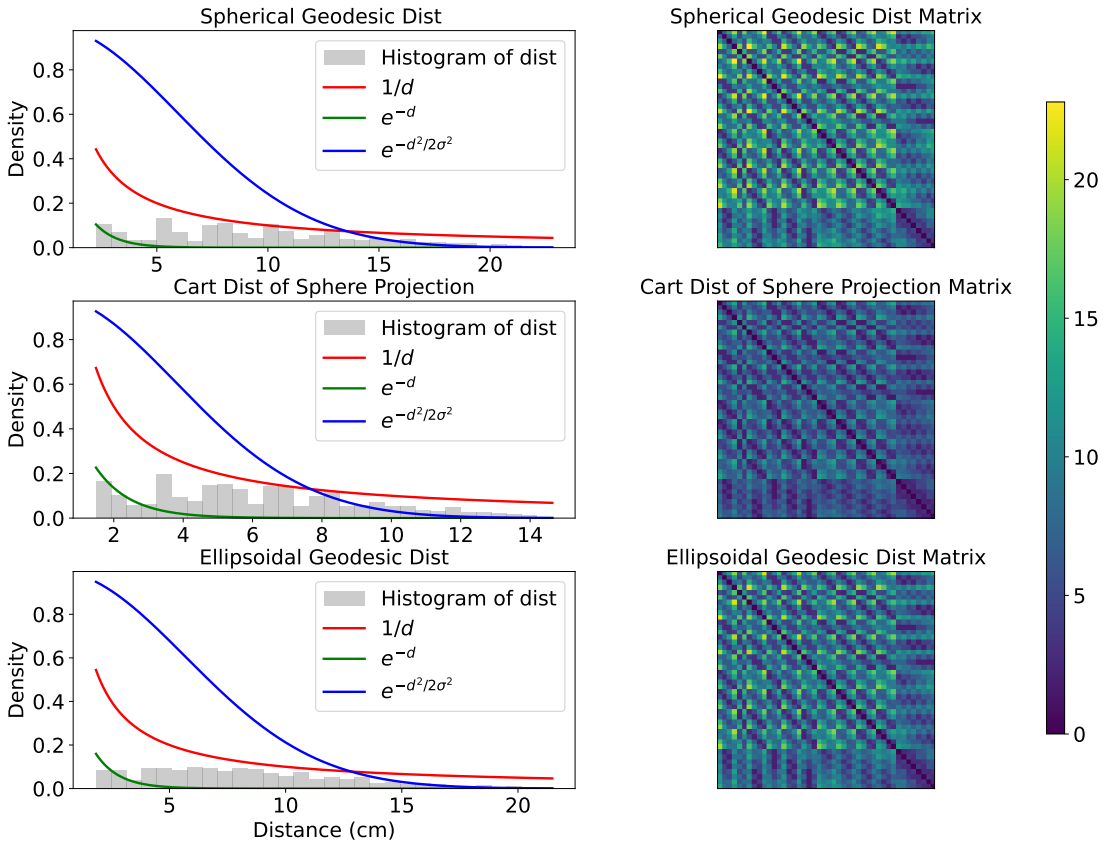
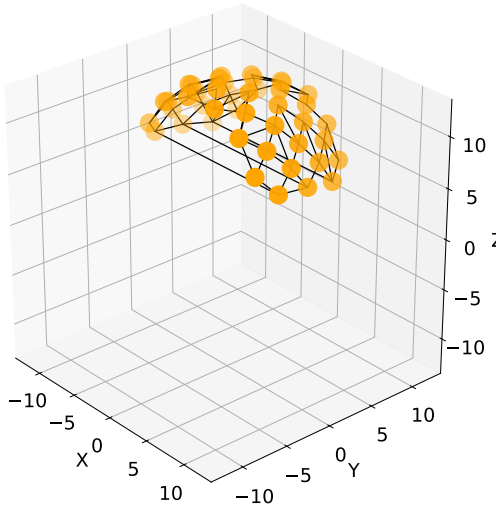



Figure 7.13 – Spherical/ellipsoidal geodesic distances and sphere-projected Cartesian distances on the HG-MI dataset. Left column: histograms of pairwise distances $d$ with decay functions $1/d$, $e^{-d}$, and $e^{-d^2/(2\sigma^2)}$ overlaid. Right column: corresponding distance matrices displayed with a shared color scale across rows; the colorbar indicates $d$ in cm.

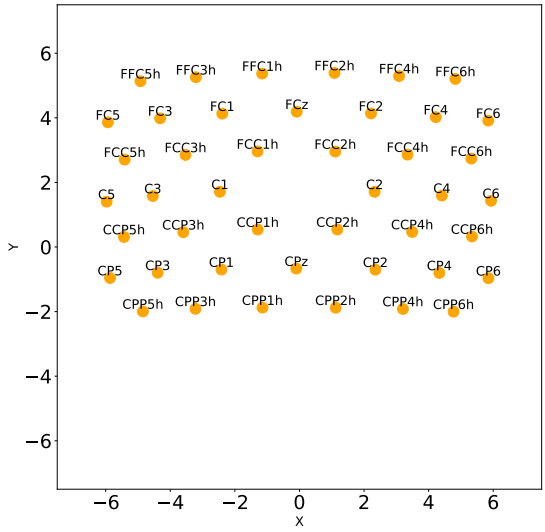
### 7.3.5 Comparison against baseline pipelines

Finally, we compare the proposed GNN-based approach to the same baseline methods leveraging EEG spatial features used in the previous chapters: CSP-OVR [ W. Wu et al. 2006] and R-MDM [Hinss et al. 2023]. As a recap, for the COG-BCI dataset, we re-implemented the CSP-OVR method using an LDA classifier, as well as the Riemannian MDM approach. Both methods were evaluated using 5-fold group-wise cross-validation with the same splits used for our GNN-based models, ensuring a fair comparison. For the HG-MI dataset, we report baseline results as provided by the MOABB benchmark and compare them with our best-performing GNN configurations, one using connectivity-based edge features (denoted as **W**), and the other using distance-based edge features (denoted as **R**). The results, presented in [Tab. 7.6], highlight the following key observations:

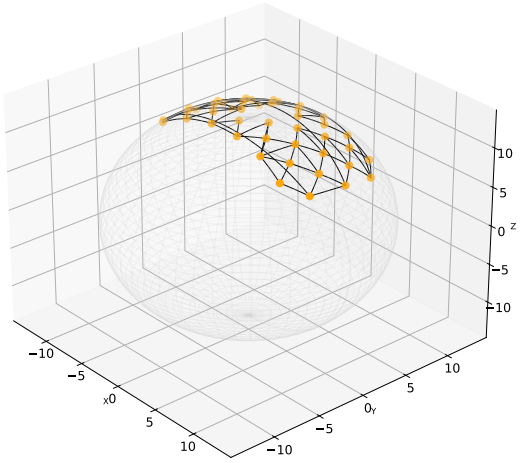
— For COG-BCI, the proposed GNN-based model significantly outperforms classical baselines, achieving up to 10% higher accuracy than CSP. This underscores the benefit of modeling brain connectivity as graphs enriched with spatial and spectral features, providing a more comprehensive representation of EEG signals in the



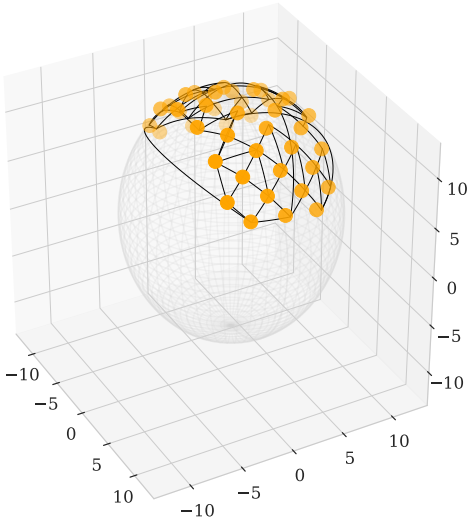

(a) Original 3D electrode positions (no projection).




(b) Azimuthal projection (MNE-like) onto the 2D plane.



(c) Projection onto a spherical surface of radius $r$.



(d) Projection onto a revolution ellipsoid.

Figure 7.14 – Visual comparison of different electrode projection schemes for the HG-MI dataset (44 channels covering the cortex zone): (a) original 3D positions; (b) MNE-style azimuthal projection onto the 2D plane; (c) spherical projection; and (d) ellipsoidal projection.

context of mental workload classification. Notably, the GNN model with distance-based edges achieves a comparable mean accuracy compared to CSP, but with a lower standard deviation, indicating more stable predictions.

— For HG-MI, CSP remains the top-performing approach. However, both of our GNN configurations consistently outperform R-MDM, especially when multiple node features are incorporated. This suggests that graph-based modeling still offers a perfor-

mance advantage even when traditional methods like CSP are strong competitors.

Table 7.6 – Comparison with state-of-the-art methods using spatial features (mean ± std).

| **Features** | **Classifier** | **COG-BCI** | **HG-MI** |
|---|---|---|---|
| Covariance | R-MDM | $0.62 \pm 0.08$ | $0.52 \pm 0.10$ |
| CSP | LDA | $0.78 \pm 0.14$ | $\mathbf{0.73 \pm 0.10}$ |
| **GNN-based approaches** | | | |
| **Node features** | **Edge features** | **COG-BCI** | **HG-MI** |
| Spectral, Topological | **R** | $0.78 \pm 0.08$ | $0.62 \pm 0.09$ |
| Positional, Topological | **W** | $\mathbf{0.88 \pm 0.09}$ | $0.65 \pm 0.08$ |
| Positional, Topological, Spectral | **W** | $0.84 \pm 0.10$ | $0.68 \pm 0.08$ |

## 7.4 Conclusion

In this chapter, we proposed an EEG signal classification approach based on graph representations. We demonstrated the benefit of incorporating electrode positions as node features, alongside spectral and topological features. We also highlighted the importance of leveraging the multivariate nature of EEG signals by using a generalized Laplacian to define the graph's connectivity. This representation, when combined with Graph Neural Networks (GNNs), led to improved classification performance compared to traditional baseline methods based on spatial EEG features.

We further investigated different strategies for representing distances between electrodes, comparing geodesic distances (on spherical and ellipsoidal surfaces) with Cartesian distances projected into a 2D azimuthal plane. These distance measures were combined with three types of decay functions to modulate the edge weights. Experimental results on two datasets, COG-BCI for mental workload classification and HG-MI for motor imagery, showed that the relevance of node features depends on the classification task. Specifically, Cartesian distances in the azimuthal plane consistently outperformed geodesic distances, and certain combinations of features (particularly positional and topological) were proven more discriminative than spectral.

In conclusion, modeling EEG signals as graphs that integrate spatial structure and spectral/topological features not only improves classification performance but also enables a more comprehensive understanding of the spatial and functional relationships underlying brain activity.

Part III

# Conclusions and perspectives

# Conclusions and Perspectives

This thesis has tackled the challenge of designing graph-based representations that explicitly integrate the spatial configuration of EEG electrodes for classification of brain functional connectivity. In fact, most conventional EEG classification pipelines rely either on handcrafted features or on pairwise functional connectivity measures. More recent works have explored deep learning architectures capable of automatically extracting features from raw signals, while other contributions have employed graph neural networks to process EEG data. However, in all these approaches, the spatial arrangement of electrodes has rarely been explicitly incorporated. When distance-based strategies were used, they often relied on simple Euclidean or geodesic measures without questioning how these spatial distances should be defined or integrated into the connectivity model.

## Conclusions

Dealing with the previous limitations, we began by proposing a new way to define the graph structure from EEG signals. Instead of relying on handcrafted similarity measures that estimate pairwise statistical dependencies between electrodes, we proposed to learn the precision matrix directly from the EEG signals under structural and Laplacian constraints [Egilmez et al. 2017]. This formulation guarantees that the resulting graph is a valid Laplacian, with positive edge weights that model the interaction between each pair of electrodes while accounting for the influence of all other nodes in the network. Once the functional connectivity graph was computed, we introduced three distinct pipelines to exploit this graph while explicitly incorporating the spatial positions of the electrodes.

In the first approach, we proposed a **functional graph to image representation**, which converts the learned connectivity graph into an image while preserving both the electrode arrangement and the underlying graph topology. This transformation enabled the use of convolutional neural networks for classification.

The second pipeline introduced a **force-based embedding paradigm**, in which electrodes are modeled as interacting particles subject to attractive and repulsive forces derived from their spatial and functional relationships. This concept was proposed in 2D and 3D formulations, and the resulting force-based features were further enriched with spectral and graph-topological features. These representations were then classified using

conventional machine learning methods.

Finally, the third pipeline explored the use of **graph neural networks operating directly on the learned or distance-driven graphs** and combined to spectral and topological features. We compared multiple strategies for encoding spatial information within edge weights and node features. In addition, when used as graph structure, we proposed different techniques to compute the underlying distances with adaptive decay functions.

We validated the proposed pipelines on two public EEG datasets: the first designed for three-class mental workload classification [Hinss et al. 2023] and the second for four-class motor imagery classification [Schirrmeister et al. 2017]. Both tasks are highly relevant for potential applications in BCI-based neuro-rehabilitation systems, where reliable decoding of mental workload and motor imagery is essential to adapt training difficulty and provide effective feedback.

The obtained results showed that learning the graph Laplacian consistently outperformed pairwise handcrafted structures, which rely on thresholding to remove irrelevant connections. This superiority was observed on both datasets and across all proposed pipelines, highlighting the effectiveness of this approach for computing functional connectivity. The best results were achieved by force-based approaches, especially when combined with graph-derived features and relative energy measures. On both datasets, our proposed pipelines outperformed the baseline Riemannian Minimum Distance to Mean classifier. For the mental workload classification task, our approaches also surpassed the Common Spatial Patterns with Linear Discriminant Analysis, while for motor imagery the best force-based method reached performance comparable to CSP.

These results show that classification performance strongly depends on the underlying task, since each paradigm elicits different neural activation patterns. Furthermore, a subject-wise analysis of performance highlights the inherent variability of EEG signals across subjects and recording sessions [Appendix. B]. We observed that the quality of the underlying EEG signals plays a crucial role: if the recordings are noisy or if a subject is unable to generate clear and discriminative brain patterns, even highly sophisticated processing pipelines result in poor classification performance.

# Perspectives

We identified many ways to extend this work further in order to explore in depth the proposed approaches. In the following, we present a few of them, specific to each proposed pipeline, and then we give general perspectives related to graph learning and classification in the context of EEG analysis.

**On the functional graph to image representation.** In the pipeline that we proposed, we generated square images of size $m \times m$, implicitly assuming equal spacing in both directions ($\Delta \mathsf{x} = \Delta \mathsf{y}$). A future perspective could be to investigate the impact of the grid aspect ratio and consider non-square or adaptively scaled image layouts to better preserve spatial geometry while retaining the benefits of an image-based representation. Beyond 2D layouts, an interesting extension would be to move toward 3D volumetric representations, allowing electrode positions to be modeled more faithfully in three-dimensional space and potentially capturing spatial relationships with greater accuracy.

**On the force based representation.** In this study, we showed that careful per-subject tuning of force parameters can enhance classification performance. Future work can include an optimized way to learn those parameters making the framework more robust.

**On the GNN-based approach.** The pipeline based on graph neural networks can be extended in many ways:

- **Subject-adaptive graph and feature design.** Inter-subject variability in EEG suggests that both the graph structure and the node features should be personalized. Future work could include (i) adapting the GNN architecture and its hyperparameters per subject, (ii) optimizing the sparsity of the distance-based graph and introducing scalable decay functions to better capture individual spatial/functional relationships, and (iii) selecting the most discriminative node features (spectral, topological, positional, etc.) on a subject-wise basis.
- **Advanced graph construction using force features.** Force-based features has demonstrated their effectiveness, therefore, it would be interesting to integrate these features to the GNN framework either as node features, or retained directly as a graph embedding step before classification.
- **Exploration of spectral and end-to-end GNN frameworks.** Since the proposed graph-learning strategy yields a valid Laplacian suitable for spectral filtering, it would be promising to investigate spectral GNNs tailored to EEG graphs. A fully end-to-end pipeline could also be developed, where graph learning from the signal and task-specific GNN training are jointly optimized rather than treated as separate stages.

Broader research directions could further extend the contributions of this work, including:

**Multiscale extension.** While the present work focused on exploiting the spatial organization of the electrodes by integrating their positions into the graph construction, future extensions could aim to enrich this representation by incorporating both frequency and temporal information. Regarding frequency, instead of only encoding spectral energy

at the node level, the EEG signal could first be filtered into distinct frequency bands before learning the graph structure. This would enable the creation of band-specific graphs that could be integrated into all our proposed pipelines and would naturally serve as a data augmentation strategy by increasing the number of graph instances per time window. Beyond this, frequency content could be modeled through *multilayer graphs*, [Kivelä et al. 2014] where each layer captures the connectivity pattern within a single band and inter-layer connections describe cross-frequency interactions. Similarly, temporal dynamics could be better exploited by constructing multilayer graphs across successive time segments or temporal scales, with inter-layer edges modeling dependencies across time. Combining the spatial foundation already leveraged in this work with frequency- and time-aware multilayer representations could provide a richer and more physiologically grounded description of EEG activity for graph-based learning.

**Multimodal extension.** A natural extension of this work is to enrich the EEG-based graphs with information from other neuroimaging modalities [Qu et al. 2025]. Structural connectivity derived from diffusion MRI (DTI) or anatomical constraints from structural MRI could be used to guide or regularize the graph learning process, ensuring that the estimated functional connections remain physiologically plausible. Such multimodal integration could produce graphs that better reflect both the functional dynamics captured by EEG and the underlying anatomical pathways.

In parallel, incorporating subject-level metadata, such as fatigue, attention level, sleep quality, or other cognitive and clinical factors, could help address the variability and sometimes low quality of EEG recordings. In many BCI protocols, participants complete questionnaires prior to each session; embedding this information into the graph representation could help model population-level patterns that explain why some subjects produce cleaner or more discriminative signals than others. One promising strategy is the use of *graph-in-graph* [Ding and Guan 2023] representations, where each node corresponds to a subject and contains an internal EEG graph enriched with clinical, demographic, or behavioral attributes. Such representations could not only improve personalization and decoding performance but also offer insights into the factors influencing signal quality and inter-subject variability. Preliminary work in this direction has shown encouraging results and merits larger-scale evaluation.

**Transfer learning and online adaptation.** Another important challenge in EEG-based BCIs is the development of models that generalize across subjects and sessions. Transfer learning [K. Zhang et al. 2020] aims to pretrain models on large and heterogeneous datasets and then adapt them rapidly to new participants, thereby reducing calibration time and improving robustness in real-time BCI applications. However, the strong variability of EEG signals across individuals and recording conditions makes this

problem particularly challenging. A promising future direction would be to investigate the feasibility of more generalizable graph learning techniques capable of capturing invariant functional patterns while remaining flexible enough to adapt to new subjects or tasks through fine-tuning or online learning.

Finally, in the present work, we focused on analyzing the feasibility and potential of the proposed graph-based approaches without imposing strict constraints on computational time or resource usage. While this choice allowed us to explore the full expressive power of these methods, it is important to note that, in the context of BCIs, particularly for online applications, low-latency and computationally efficient models are essential. Future work could therefore extend this study by investigating optimized implementations of the proposed frameworks and developing strategies to reduce computation time and memory requirements, thus moving closer to practical real-time BCI deployment.

Part IV

# Supplementary material

# Appendix A

# Common Spatial Patterns

Let us consider a multivariate EEG signal, recorded from $n$ channels at a sampling frequency $F_s$ and then epoched into trials of $t$ seconds corresponding to two different classes of cognitive state a and b. We denote this signal by $\mathbf{X} \in \mathbb{R}^{n \times tF_s}$. For each trial, the normalized spatial covariance matrice is computed as:

$$\mathbf{S} = \frac{\mathbf{X}\mathbf{X}^\top}{\text{trace}(\mathbf{X}\mathbf{X}^\top)}.$$

Averaging across all trials of each class yields the mean covariance matrices $\mathbf{S}_a$ and $\mathbf{S}_b$. The eigen-decompositon of the sum is given by:

$$\mathbf{S}_a + \mathbf{S}_b = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top. \tag{A.1}$$

This allows to find the whitening transformation:

$$\mathbf{P} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top.$$

Now, applying $\mathbf{P}$ to $\mathbf{S}_a$ and $\mathbf{S}_b$ yields:

$$\tilde{\mathbf{S}}_a = \mathbf{P}\mathbf{S}_a\mathbf{P}^\top, \quad \tilde{\mathbf{S}}_b = \mathbf{P}\mathbf{S}_b\mathbf{P}^\top,$$

where $\tilde{\mathbf{S}}_a + \tilde{\mathbf{S}}_b = \mathbf{I}$. Both matrices are diagonal and share the same eigenvectors, with eigenvalues summing to 1 for each component. If $\tilde{\mathbf{S}}_a = \mathbf{V}\mathbf{\Lambda}_a\mathbf{V}^\top$, then $\tilde{\mathbf{S}}_b = \mathbf{V}\mathbf{\Lambda}_b\mathbf{V}^\top$ with $\mathbf{\Lambda}_a + \mathbf{\Lambda}_b = \mathbf{I}$. When an eigenvalue in $\mathbf{\Lambda}_a$ is maximal, the corresponding eigenvalue in $\mathbf{\Lambda}_b$ is minimal, and vice versa. The projection matrix is given by $\tilde{\mathbf{W}} = \mathbf{V}^\top\mathbf{P}$. Projecting the original signals as:

$$\mathbf{Z} = \tilde{\mathbf{W}}\mathbf{X}, \tag{A.2}$$

yields spatially filtered signals $\mathbf{Z}$ whose components maximize the variance for one class while simultaneously minimizing it for the other, thus enhancing the discriminative power of the extracted features for subsequent classification.

# Appendix B

# Per-subject results using all methods

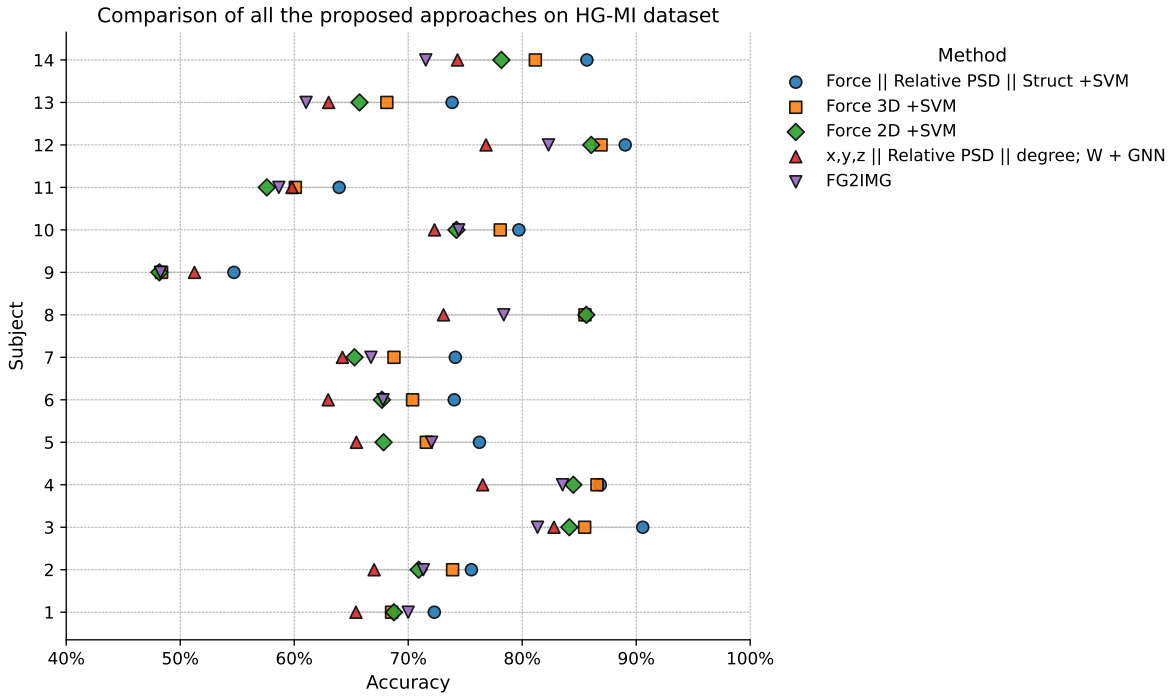



Figure B.1 – Results per subject of all the proposed approaches on HG-MI dataset

[Fig. B.1] presents the subject-wise classification accuracies obtained for the 14 participants of the MI-HG dataset across the main proposed approaches. For clarity, only a subset of methods achieving the best average performance is shown. From Chapter 5, we report the results of the FG2IMG framework, which employs a graph structure learned as a Laplacian. From chapter 6 we choose the force-based node features in 3D, in 2D classified with SVM, and their concatenation with structural and relative PSD features, followed by feature selection and SVM classification. In addition, we include the GNN-based approach, where each node is described by the concatenation of its spatial coordinates $(\mathsf{x}, \mathsf{y}, \mathsf{z})$, relative PSD features, and weighted degree, while edge weights are learned from the Laplacian. Baseline pipelines (e.g., CSP and R-MDM) are omitted here since the results per subject are not provided.

The results show that force-based approaches generally achieve superior accuracy for most subjects, while deep learning models (CNN and GNN) yield overall lower performance on this dataset. Furthermore, the performance curves across methods tend to follow similar subject-wise trends: accuracies rise and fall together, likely reflecting differences in EEG signal quality and the varying number of available windows per subject.

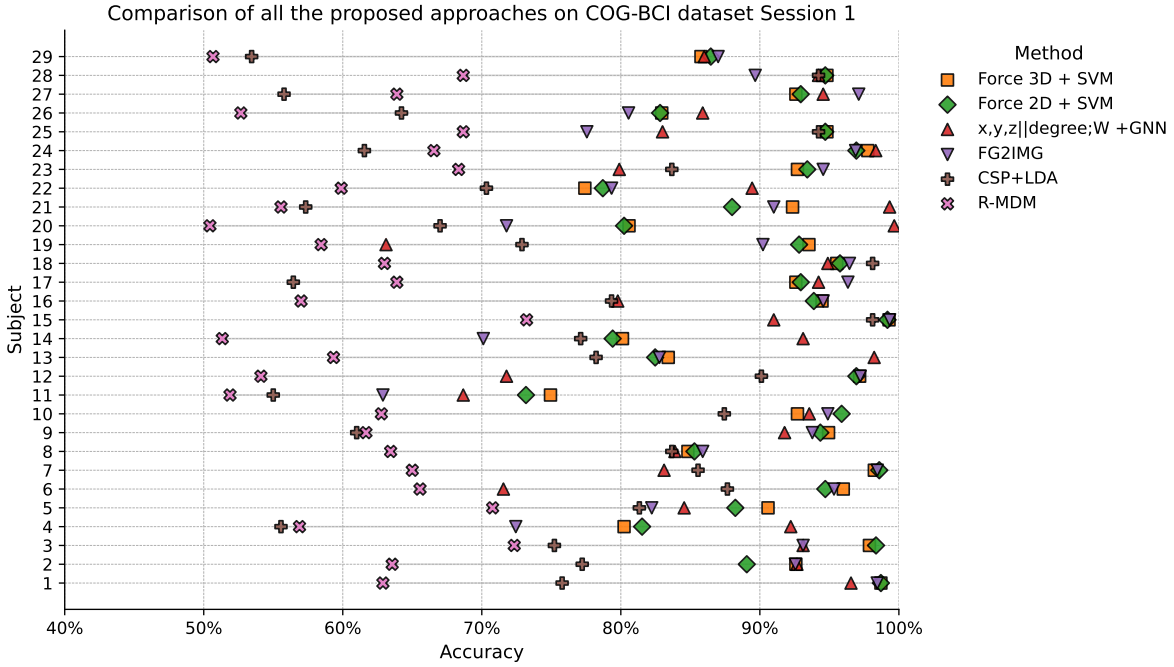



Figure B.2 – Results per subject of all the proposed approaches on COG-BCI S1

For the COG-BCI dataset, which includes 29 participants, the results are grouped by Session 1 [Fig. B.2] and Session 2 [Fig. B.2] for better visual clarity. In this analysis, we also include the traditional baselines (CSP+LDA and R-MDM), whose results were computed by us for comparison. For the GNN-based approach, we use node features composed of the electrode spatial coordinates concatenated with the weighted degree only (excluding PSD features), as this configuration provided better performance. Apart from the baseline methods (R-MDM and CSP), our proposed approaches exhibit a similar subject-wise trend to that observed in the MI-HG dataset, where the accuracy curves tend to increase and decrease consistently across methods. Overall, the approaches based on a learned Laplacian graph structure achieve the highest classification accuracies. These plots also highlight the variability in EEG-based classification across subjects and even between sessions for the same participant. Such variability can be attributed to several factors, including changes in the participant's mental state and slight differences in EEG cap placement between sessions, which may affect the signal quality and, consequently, the classification performance. In the case of mental workload classification, the choice of the best method is dependent on the subject and session. This motivates more the idea

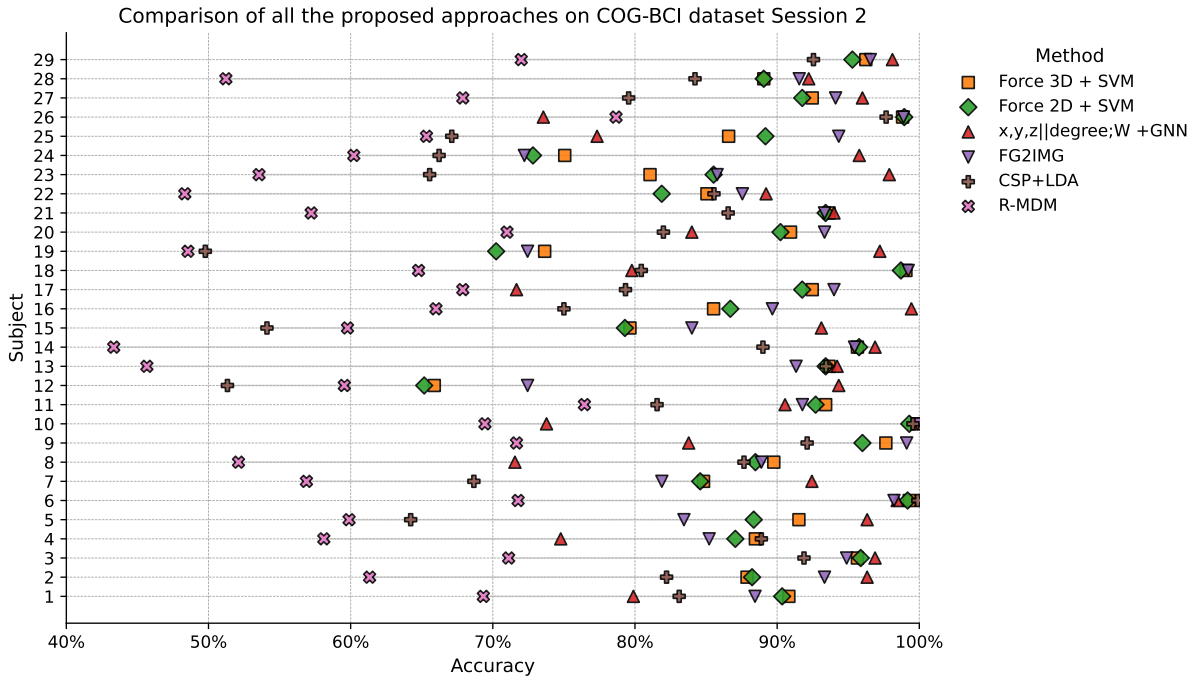



Figure B.3 – Results per subject of all the proposed approaches on COG-BCI S2

to adapt the hyperparameters (discussed earlier in the chapters) for each subject in order to achieve better outcomes.

# Bibliography


Abdulhussien, A. S., A. T. AbdulSaddaa, and K. Iqbal (2022), « Automatic seizure detection with different time delays using SDFT and time-domain feature extraction », *in*: *Journal of Biomedical Research* 36.*1*, p. 48.

Adamovich, T., I. Zakharov, A. Tabueva, and S. Malykh (2022), « The thresholding problem and variability in the EEG graph network parameters », *in*: *Scientific Reports* 12.*1*, p. 18659.

Alam, M. N., M. I. Ibrahimy, and S. M. A. Motakabber (2021), « Feature Extraction of EEG Signal by Power Spectral Density for Motor Imagery Based BCI », *in*: *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, pp. 234–237, DOI: 10.1109/ICCCE50029.2021.9467141.

Ali, Z. et al. (2020), « Deep learning in citation recommendation models survey », *in*: *Expert Systems with Applications* 162, p. 113790.

Aliyu, I. and C. G. Lim (2023), « Selection of optimal wavelet features for epileptic EEG signal classification with LSTM », *in*: *Neural Computing and Applications* 35.*2*, pp. 1077–1097.

Almohammadi, A. and Y.-K. Wang (2024), « Revealing brain connectivity: graph embeddings for EEG representation learning and comparative analysis of structural and functional connectivity », *in*: *Frontiers in Neuroscience* 17, p. 1288433.

Alotaiby, T. N., S. A. Alshebeili, F. M. Alotaibi, and S. R. Alrshoud (2017), « Epileptic seizure prediction using CSP and LDA for scalp EEG signals », *in*: *Computational intelligence and neuroscience* 2017.*1*, p. 1240323.

Andac, D., K.-A. Toshiaki, W. Ye, and E. Deniz (2022), « EEG-GAT: graph attention networks for classification of electroencephalogram (EEG) signals », *in*: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, pp. 30–35.

Ang, K. K. et al. (2012), « Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b », *in*: *Frontiers in neuroscience* 6, p. 39.

Aristimunha, B. et al. (2025), *Mother of all BCI Benchmarks*, version v1.2.0, DOI: 10.5281/zenodo.10034223, URL: https://github.com/NeuroTechX/moabb.

Azad, F., S. B. Shouraki, S. Nazari, and M. Chan (2025), « Lightweight CNN-based Seizure Classification via Leveraging Chimera States in iEEG Recordings », *in*: *Results in Engineering*, p. 106000.

Barachant, A., S. Bonnet, M. Congedo, and C. Jutten (2010a), « Riemannian geometry applied to BCI classification », *in*: *International conference on latent variable analysis and signal separation*, Springer, pp. 629–636.

— (2012), « Multiclass Brain–Computer Interface Classification by Riemannian Geometry », *in*: *IEEE Transactions on Biomedical Engineering* 59.*4*, pp. 920–928, DOI: 10.1109/TBME.2011.2172210.

Barachant, A., S. Bonnet, M. Congedo, and C. Jutten (2010b), « Common spatial pattern revisited by Riemannian geometry », *in*: *2010 IEEE international workshop on multimedia signal processing*, IEEE, pp. 472–476.

Bashivan, P., I. Rish, M. Yeasin, and N. Codella (2015), « Learning representations from EEG with deep recurrent-convolutional neural networks », *in*: *arXiv preprint arXiv:1511.06448*.

Bastian, M., S. Heymann, and M. Jacomy (2009), *Gephi: An Open Source Software for Exploring and Manipulating Networks*, URL: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

Bastos, A. M. and J.-M. Schoffelen (2016), « A tutorial review of functional connectivity analysis methods and their interpretational pitfalls », *in*: *Frontiers in systems neuroscience* 9, p. 175.

Blinowska, K. and P. Durka (2006), « Electroencephalography (eeg) », *in*: *Wiley encyclopedia of biomedical engineering* 10, p. 9780471740360.

Bo, D. et al. (2023), « A survey on spectral graph neural networks », *in*: *arXiv preprint arXiv:2302.05631*.

Boonyakitanont, P., A. Lek-Uthai, K. Chomtho, and J. Songsiri (2020), « A review of feature extraction and performance evaluation in epileptic seizure detection using EEG », *in*: *Biomedical Signal Processing and Control* 57, p. 101702.

Borghini, G. et al. (2014), « Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness », *in*: *Neuroscience & Biobehavioral Reviews* 44, pp. 58–75.

Brandes, U., P. Kenis, J. Lerner, and D. Van Raaij (2009), « Network analysis of collaboration structure in Wikipedia », *in*: *Proceedings of the 18th international conference on World wide web*, pp. 731–740.

Brandes, U. and C. Pich (2007), « Centrality estimation in large networks », *in*: *International Journal of Bifurcation and Chaos* 17.*07*, pp. 2303–2318.

Bresenham, J. E. (1965), « Algorithm for computer control of a digital plotter », *in*: *IBM Systems Journal* 4.*1*, pp. 25–30, DOI: 10.1147/sj.41.0025.

Briels, C. T. et al. (2020), « Reproducibility of EEG functional connectivity in Alzheimer's disease », *in*: *Alzheimer's research & therapy* 12, pp. 1–14.

Britton, J. W. et al. (2016), « Electroencephalography (EEG): An introductory text and atlas of normal and abnormal findings in adults, children, and infants », *in*.

Bronstein, M. M. et al. (2017), « Geometric deep learning: going beyond euclidean data », *in*: *IEEE Signal Processing Magazine* 34.*4*, pp. 18–42.

Brouwers, J. M. (2016), *PyGeodesy: Pure Python geodesy tools*, https://github.com/mrJean1/PyGeodesy, A pure Python implementation of geodesy tools for ellipsoidal and spherical earth models.

Carrara, I. and T. Papadopoulo (2024), « Classification of BCI-EEG based on the augmented covariance matrix », *in*: *IEEE Transactions on Biomedical Engineering* 71.*9*, pp. 2651–2662.

Cheong, S.-H. and Y.-W. Si (2020), « Force-directed algorithms for schematic drawings and placement: A survey », *in*: *Information Visualization* 19.*1*, pp. 65–91.

Chevallier, S. et al. (2024), « The largest EEG-based BCI reproducibility study for open science: the MOABB benchmark », *in*: *arXiv preprint arXiv:2404.15319*.

Clerc, M., L. Bougrain, and F. Lotte (2016), *Brain-computer Interfaces: Foundations and Methods*, ISTE Limited.

Corsi, M.-C., S. Chevallier, F. D. V. Fallani, and F. Yger (2022), « Functional Connectivity Ensemble Method to Enhance BCI Performance (FUCONE) », *in*: *IEEE Transactions on Biomedical Engineering* 69.*9*, pp. 2826–2838, DOI: 10.1109/TBME.2022.3154885.

Coulomb, C.-A. d. (1785), « Premier mémoire sur l'électricité et le magnétisme », *in*: *Histoire de l'Académie Royale des Sciences*, First dissertation on electricity and magnetism. (In French), pp. 569–577.

Courellis, H. S., J. R. Iversen, H. Poizner, and G. Cauwenberghs (2016), « EEG channel interpolation using ellipsoid geodesic length », *in*: *2016 IEEE biomedical circuits and systems conference (BioCAS)*, IEEE, pp. 540–543.

Craik, A., Y. He, and J. L. Contreras-Vidal (2019), « Deep learning for electroencephalogram (EEG) classification tasks: a review », *in*: *Journal of neural engineering* 16.*3*, p. 031001.

Demir, A. et al. (2021), « EEG-GNN: Graph neural networks for classification of electroencephalogram (EEG) signals », *in*: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, pp. 1061–1067.

Dempster, A. P. (Mar. 1972), « Covariance Selection », *in*: *Biometrics* 28.*1*, Special Multivariate Issue, pp. 157–175, URL: https://www.jstor.org/stable/2528966.

Devi, C. N., R. Krishnamoorthi, and S. Nayak (2025), « Development of EEG-Based Prosthetic Limb for Paraplegic and Quadriplegic Patients », *in*: *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*, pp. 1782–1785, DOI: 10.1109/ICMLAS64557.2025.10968343.

Ding, Y. and C. Guan (2023), « GIGN: Learning graph-in-graph representations of EEG signals for continuous emotion recognition », *in*: *2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, pp. 1–5.

Ding, Y., N. Robinson, et al. (2023), « LGGNet: Learning from local-global-graph representations for brain–computer interface », *in*: *IEEE Transactions on Neural Networks and Learning Systems* 35.*7*, pp. 9773–9786.

Do, T. H. et al. (2019), « Matrix Completion with Variational Graph Autoencoders: Application in Hyperlocal Air Quality Inference », *in*: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7535–7539, DOI: 10.1109/ICASSP.2019.8683787.

DocCheck (2023), *10-20-System*, Accessed: 2025-06-30, URL: https://www.doccheck.com/de/detail/photos/42650-10-20-system.

Dong, X., D. Thanou, M. Rabbat, and P. Frossard (2019), « Learning graphs from data: A signal representation perspective », *in*: *IEEE Signal Processing Magazine* 36.*3*, pp. 44–63.

Duhan, M. et al. (2020), « Feature Extraction and Classification of EEG Signals Using Machine Learning Algorithms for Biometric Systems », *in*: *International Journal of Computer Information Systems and Industrial Management Applications* 12, pp. 10–10.

Duong, C. T. et al. (2019), « On node features for graph neural networks », *in*: *arXiv preprint arXiv:1911.08795*.

Duvenaud, D. K. et al. (2015), « Convolutional networks on graphs for learning molecular fingerprints », *in*: *Advances in neural information processing systems* 28.

Eades, P. (1984), « A heuristic for graph drawing », *in*: *Congressus numerantium* 42.*11*, pp. 149–160.

Easley, D., J. Kleinberg, et al. (2010), *Networks, crowds, and markets: Reasoning about a highly connected world*, vol. 1, Cambridge university press Cambridge.

Edla, D. R., K. Mangalorekar, G. Dhavalikar, and S. Dodia (2018), « Classification of EEG data for human mental state analysis using Random Forest Classifier », *in*: *Procedia computer science* 132, pp. 1523–1532.

Egilmez, H. E., E. Pavez, and A. Ortega (2017), « Graph learning from data under Laplacian and structural constraints », *in*: *IEEE Journal of Selected Topics in Signal Processing* 11.*6*, pp. 825–841.

Euler, L. (1736), « Solutio problematis ad geometriam situs pertinentis », *in*: *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8, pp. 128–140.

Fernandez, J., B. Innocenti, and B. López (2024), « Covariance Matrices and Case-Based Reasoning Synergy for Interpretable EEG Classification in Neurological Disorders », *in*.

Fitzgibbon, S. P., D. M. Powers, K. J. Pope, and C. R. Clark (2007), « Removal of EEG noise and artifact using blind source separation », *in*: *Journal of Clinical Neurophysiology* 24.*3*, pp. 232–243.

Freeman, L. C. (1978), « Centrality in social networks conceptual clarification », *in*: *Social networks* 1.*3*, pp. 215–239.

Friedman, J., T. Hastie, and R. Tibshirani (2008), « Sparse inverse covariance estimation with the graphical lasso », *in*: *Biostatistics* 9.*3*, pp. 432–441.

Fruchterman, T. M. and E. M. Reingold (1991), « Graph drawing by force-directed placement », *in*: *Software: Practice and experience* 21.*11*, pp. 1129–1164.

Gopan K, G., N. Sinha, and D. Babu J (2022), « Inter-channel Covariance Matrices Based Analyses of EEG Baselines », *in*: *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 1303–1307, DOI: 10.23919/EUSIPCO55093.2022.9909861.

Gosala, B. et al. (2023), « Wavelet transforms for feature engineering in EEG data processing: An application on Schizophrenia », *in*: *Biomedical Signal Processing and Control* 85, p. 104811.

Gramfort, A. et al. (2013), « MEG and EEG data analysis with MNE-Python », *in*: *Frontiers in Neuroinformatics* 7, p. 267.

Grosse-Wentrup, M. and M. Buss (2008), « Multiclass common spatial patterns and information theoretic feature extraction », *in*: *IEEE transactions on Biomedical Engineering* 55.*8*, pp. 1991–2000.

Guler, I. and E. D. Ubeyli (2007), « Multiclass support vector machines for EEG-signals classification », *in*: *IEEE transactions on information technology in biomedicine* 11.*2*, pp. 117–126.

Hagberg, A. A., D. A. Schult, and P. J. Swart (2008), « Exploring Network Structure, Dynamics, and Function using NetworkX », *in*: *Proceedings of the 7th Python in Science Conference*, ed. by G. Varoquaux, T. Vaught, and J. Millman, Pasadena, CA USA, pp. 11–15.

He, J. et al. (2022), « Spatial–temporal seizure detection with graph attention network and bi-directional LSTM architecture », *in*: *Biomedical Signal Processing and Control* 78, p. 103908.

Hinss, M. F. et al. (2023), « Open multi-session and multi-task EEG cognitive Dataset for passive brain-computer Interface Applications », *in*: *Scientific Data* 10.*1*, p. 85.

Hook, R. (1679h), « De Potentia Restitutiva, or of Spring. Explaining the Power of Springing Bodies », *in*: *The Royal Society: London, UK* 1678.

Huang, F. et al. (2025), « A review of studies on constructing classification models to identify mental illness using brain effective connectivity », *in*: *Psychiatry Research: Neuroimaging* 346, p. 111928.

Hussein, R., H. Palangi, R. K. Ward, and Z. J. Wang (2019), « Optimized deep neural network architecture for robust detection of epileptic seizures using EEG signals », *in*: *Clinical Neurophysiology* 130.*1*, pp. 25–37.

İnce, R., S. S. Adanır, and F. Sevmez (2021), « The inventor of electroencephalography (EEG): Hans Berger (1873–1941) », *in*: *Child's Nervous System* 37.*9*, pp. 2723–2724.

Jacomy, M., T. Venturini, S. Heymann, and M. Bastian (2014), « ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software », *in*: *PloS one* 9.*6*, e98679.

Jang, S., S.-E. Moon, and J.-S. Lee (2018), « EEG-based video identification using graph signal modeling and graph convolutional neural network », *in*: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, pp. 3066–3070.

Jannoud, I. A. and M. Z. Masoud (2014), « On understanding centrality in directed citation graph », *in*: *Advanced Computer and Communication Engineering Technology: Proceedings of the 1st International Conference on Communication and Computer Engineering*, Springer, pp. 43–51.

Jia, J. et al. (2022), « CR-GCN: channel-relationships-based graph convolutional network for EEG emotion recognition », *in*: *Brain Sciences* 12.*8*, p. 987.

Jia, M. et al. (2022), « Efficient graph convolutional networks for seizure prediction using scalp EEG », *in*: *Frontiers in Neuroscience* 16, p. 967116.

Jia, Z. et al. (2021), « Multi-view spatial-temporal graph convolutional networks with domain generalization for sleep stage classification », *in*: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29, pp. 1977–1986.

Jiang, W. and J. Luo (2022), « Graph neural network for traffic forecasting: A survey », *in*: *Expert systems with applications* 207, p. 117921.

Jiang, X., G.-B. Bian, and Z. Tian (2019), « Removal of artifacts from EEG signals: a review », *in*: *Sensors* 19.*5*, p. 987.

Jiao, Z. et al. (2018), « Deep convolutional neural networks for mental load classification based on EEG data », *in*: *Pattern Recognition* 76, pp. 582–595.

Jothiraj, S. N., C. Mills, Z. C. Irving, and J. W. Kam (2025), « Detection of freely moving thoughts using SVM and EEG signals », *in*: *Journal of neural engineering* 22.*2*, p. 026021.

Kalousios, S. et al. (2025), « Memory optimized random forest classifier for EEG seizure detection in implantable monitoring and closed-loop neurostimulation devices », *in*: *Journal of Neural Engineering* 22.*3*, p. 036023.

Kamada, T., S. Kawai, et al. (1989), « An algorithm for drawing general undirected graphs », *in*: *Information processing letters* 31.*1*, pp. 7–15.

Karney, C. F. (2013), « Algorithms for geodesics », *in*: *Journal of geodesy* 87.*1*, pp. 43–55.

Kivelä, M. et al. (2014), « Multilayer networks », *in*: *Journal of complex networks* 2.*3*, pp. 203–271.

Kleinberg, J. M. (1999), « Authoritative sources in a hyperlinked environment », *in*: *Journal of the ACM (JACM)* 46.*5*, pp. 604–632.

Klepl, D., F. He, et al. (2022), « EEG-based graph neural network classification of Alzheimer's disease: An empirical evaluation of functional connectivity methods », *in*: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 30, pp. 2651–2660.

Klepl, D., M. Wu, and F. He (2024), « Graph neural network-based eeg classification: A survey », *in*: *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.

Kobourov, S. G. (2012), « Spring Embedders and Force Directed Graph Drawing Algorithms », *in*.

Koles, Z. J. (1991), « The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG », *in*: *Electroencephalography and clinical Neurophysiology* 79.*6*, pp. 440–447.

Kooijman, C. M. et al. (2000), « Phantom pain and phantom sensations in upper limb amputees: an epidemiological study », *in*: *Pain* 87.*1*, pp. 33–41.

Kuanar, S. et al. (2018), « Cognitive analysis of working memory load from EEG, by a deep recurrent neural network », *in*: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 2576–2580.

Kumar, N., K. Alam, and A. H. Siddiqi (2017), « Wavelet Transform for Classification of EEG Signal using SVM and ANN », *in*: *Biomedical and Pharmacology Journal* 10.*4*, pp. 2061–2069.

Larson, E. et al. (July 2025), *MNE-Python*, version v1.10.0, DOI: 10.5281/zenodo.15928841, URL: https://doi.org/10.5281/zenodo.15928841.

Li, R. et al. (2021), « Graph signal processing, graph neural network and graph learning on biological data: a systematic review », *in*: *IEEE Reviews in Biomedical Engineering* 16, pp. 109–135.

Lin, Z. et al. (2021), « Fatigue driving recognition based on deep learning and graph neural network », *in*: *Biomedical Signal Processing and Control* 68, p. 102598.

Liu, H., J. Zhang, Q. Liu, and J. Cao (2022), « Minimum spanning tree based graph neural network for emotion classification using EEG », *in*: *Neural Networks* 145, pp. 308–318.

Liu, Y. et al. (2023), « Scalp EEG-based automatic detection of epileptiform events via graph convolutional network and bi-directional LSTM co-embedded broad learning system », *in*: *IEEE Signal Processing Letters* 30, pp. 448–452.

Lotte, F., L. Bougrain, et al. (2018), « A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update », *in*: *Journal of neural engineering* 15.*3*, p. 031005.

Lotte, F., M. Congedo, et al. (2007), « A review of classification algorithms for EEG-based brain–computer interfaces », *in*: *Journal of neural engineering* 4.*2*, R1.

Ma, Y.-N. et al. (2025), « Integrative neurorehabilitation using brain-computer interface: From motor function to mental health after stroke », *in*: *Bioscience trends*.

Ma, X., H. Zhou, and Z. Li (2021), « On the resilience of modern power systems: A complex network perspective », *in*: *Renewable and Sustainable Energy Reviews* 152, p. 111646.

Maitin, A. M., A. Nogales, P. Chazarra, and Á. J. García-Tejedor (2023), « EEGraph: An open-source Python library for modeling electroencephalograms using graphs », *in*: *Neurocomputing* 519, pp. 127–134.

Makris, N. et al. (2008), « MRI-based anatomical model of the human head for specific absorption rate mapping », *in*: *Medical & biological engineering & computing* 46.*12*, pp. 1239–1251.

Manjusha, M. and R. Harikumar (2016), « Performance analysis of KNN classifier and K-means clustering for robust classification of epilepsy from EEG signals », *in*: *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, IEEE, pp. 2412–2416.

Martínez Vásquez, D. A., H. F. Posada-Quintero, and D. M. Rivera Pinzón (2023), « Mutual information between EDA and EEG in multiple cognitive tasks and sleep deprivation conditions », *in*: *Behavioral Sciences* 13.*9*, p. 707.

Modak, S., K. Samanta, S. Halder, and S. Chatterjee (2025), « EEG Rhythm-Based Functional Brain Connectivity for Automated Detection of Schizophrenia Employing Deep Learning », *in*: *IEEE Transactions on Instrumentation and Measurement*.

Moon, S.-E. et al. (2020), « Emotional EEG classification using connectivity features and convolutional neural networks », *in*: *Neural Networks* 132, pp. 96–107.

Morris, C. et al. (2019), « Weisfeiler and leman go neural: Higher-order graph neural networks », *in*: *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 01, pp. 4602–4609.

Mouches, P. et al. (2024), « Time CNN and Graph Convolution Network for Epileptic Spike Detection in MEG Data », *in*: *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, IEEE, pp. 1–5.

Müller-Putz, G. R. (2020), « Electroencephalography », *in*: *Handbook of clinical neurology* 168, pp. 249–262.

Noack, A. (2007), « Energy models for graph clustering », *in*: *Journal of graph algorithms and applications* 11.*2*, pp. 453–480.

Nogales, A., Á. J. García-Tejedor, P. Chazarra, and A. Ugalde-Canitrot (2023), « Discriminating and understanding brain states in children with epileptic spasms using deep learning and graph metrics analysis of brain connectivity », *in*: *Computer Methods and Programs in Biomedicine* 232, p. 107427.

Nolte, G. et al. (2004), « Identifying true brain interaction from EEG data using the imaginary part of coherency », *in*: *Clinical neurophysiology* 115.*10*, pp. 2292–2307.

Onnela, J.-P., J. Saramäki, J. Kertész, and K. Kaski (2005), « Intensity and coherence of motifs in weighted complex networks », *in*: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 71.*6*, p. 065103.

Ortiz-Catalan, M. et al. (2016), « Phantom motor execution facilitated by machine learning and augmented reality as treatment for phantom limb pain: a single group, clinical trial in patients with chronic intractable phantom limb pain », *in*: *The Lancet* 388.*10062*, pp. 2885–2894.

Osipowicz, K., M. R. Sperling, A. D. Sharan, and J. I. Tracy (2016), « Functional MRI, resting state fMRI, and DTI for predicting verbal fluency outcome following resective surgery for temporal lobe epilepsy », *in*: *Journal of neurosurgery* 124.*4*, pp. 929–937.

Pedregosa, F. et al. (2011), « Scikit-learn: Machine Learning in Python », *in*: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pfurtscheller, G., G. R. Müller-Putz, R. Scherer, and C. Neuper (2008), « Rehabilitation with brain-computer interface systems », *in*: *Computer* 41.*10*, pp. 58–65.

Plechawska-Wójcik, M., M. Tokovarov, M. Kaczorowska, and D. Zapała (2019), « A three-class classification of cognitive workload based on EEG spectral data », *in*: *Applied sciences* 9.*24*, p. 5340.

Puranik, P., R. Kshirsagar, and S. Motdhare (2018), « Elementary time frequency analysis of EEG signal processing », *in*: *EAI Endorsed Transactions on Pervasive Health and Technology* 4.*14*.

Qu, G. et al. (2025), « Integrated brain connectivity analysis with fMRI, DTI, and sMRI powered by interpretable graph neural networks », *in*: *Medical Image Analysis* 103, p. 103570.

Raeisi, K. et al. (2022), « A graph convolutional neural network for the automated detection of seizures in the neonatal EEG », *in*: *Computer methods and programs in biomedicine* 222, p. 106950.

Rahman, M. K., M. H. Sujon, and A. Azad (2020), « Force2Vec: Parallel force-directed graph embedding », *in*: *2020 IEEE international conference on data mining (ICDM)*, IEEE, pp. 442–451.

Rakotomamonjy, A., V. Guigue, G. Mallet, and V. Alvarado (2005), « Ensemble of SVMs for improving brain computer interface P300 speller performances », *in*: *International conference on artificial neural networks*, Springer, pp. 45–50.

Ramachandran, V. S. and D. Rogers-Ramachandran (1996), « Synaesthesia in phantom limbs induced with mirrors », *in*: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 263.*1369*, pp. 377–386.

Reiser, P. et al. (2022), « Graph neural networks for materials science and chemistry », *in*: *Communications Materials* 3.*1*, p. 93.

Roy, R. N. et al. (2022), « Retrospective on the first passive brain-computer interface competition on cross-session workload estimation », *in*: *Frontiers in Neuroergonomics* 3, p. 838342.

Rubinov, M. and O. Sporns (2010), « Complex network measures of brain connectivity: uses and interpretations », *in*: *Neuroimage* 52.*3*, pp. 1059–1069.

Salama, E. S., R. A. El-Khoribi, M. E. Shoman, and M. A. W. Shalaby (2018), « EEG-based emotion recognition using 3D convolutional neural networks », *in*: *International Journal of Advanced Computer Science and Applications* 9.*8*.

Salvador, R. et al. (2005), « Neurophysiological architecture of functional magnetic resonance images of human brain », *in*: *Cerebral cortex* 15.*9*, pp. 1332–1342.

Sarkis, M., M. Rizkallah, and S. Moussaoui (2025a), « Force-Driven Graph Learning and Classification », *in*: *IEEE Signal Processing Letters*, Manuscript under review.

— (2025b), « Réseaux de neurones sur graphes pour l'analyse de signaux EEG », *in*: *Colloque sur le Traitement du Signal et des Images (GRETSI)*, Accepted for publication.

Sarkis, M., M. Rizkallah, and S. Moussaoui (June 2024a), « A Graph Signal Processing Framework based on Graph Learning and Graph Neural Networks for Mental Workload Classification from EEG signals », *in*: *7th Graph Signal Processing Workshop (GSP)*, https://hal.science/hal-04638684, Delft, Netherlands.

— (2024b), « Functional Graph Image Representation applied to EEG-based Mental Workload Classification », *in*: *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1–4, DOI: 10.1109/EMBC53108.2024.10781733.

— (2025c), « Graph Force Learning from EEG Signals for Mental Workload Classification », *in*: *IEEE Statistical Signal Processing Workshop SSP hal-05058873*, URL: https://hal.science/hal-05058873v1.

Schirrmeister, R. T. et al. (2017), « Deep learning with convolutional neural networks for EEG decoding and visualization », *in*: *Human brain mapping* 38.*11*, pp. 5391–5420.

Selvaraju, R. R. et al. (2017), « Grad-cam: Visual explanations from deep networks via gradient-based localization », *in*: *Proceedings of the IEEE international conference on computer vision*, pp. 618–626.

Shan, X. et al. (2022), « Spatial–temporal graph convolutional network for Alzheimer classification based on brain functional connectivity imaging of electroencephalogram », *in*: *Human Brain Mapping* 43.*17*, pp. 5194–5209.

Shuman, D. I. et al. (2013), « The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains », *in*: *IEEE signal processing magazine* 30.*3*, pp. 83–98.

Soler, A., V. Naas, A. Giri, and M. Molinas (2024), « EEG Source Imaging Enhances Motor Imagery Classification », *in*: *ESANN*.

Solomon Jr, O. M. (1991), « PSD computations using Welch's method », *in*: *NASA STI/Recon Technical Report N* 92, p. 23584.

Sporns, O. (2002), « Network analysis, complexity, and brain function », *in*: *Complexity* 8.*1*, pp. 56–60.

Stankovic, L. et al. (2019), « Graph signal processing–Part I: Graphs, graph spectra, and spectral clustering », *in*: *arXiv preprint arXiv:1907.03467*.

Sultana, O. F., M. Bandaru, M. A. Islam, and P. H. Reddy (2024), « Unraveling the complexity of human brain: Structure, function in healthy and disease states », *in*: *Ageing Research Reviews* 100, p. 102414.

Sun, K., J. Liu, et al. (2020), « Graph force learning », *in*: *2020 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 2987–2994.

Sun, K., F. Xia, et al. (2022), « Attributed graph force learning », *in*: *IEEE Transactions on Neural Networks and Learning Systems* 35.*4*, pp. 4502–4515.

Sun, M. et al. (2020), « Graph convolutional networks for computational drug development and discovery », *in*: *Briefings in bioinformatics* 21.*3*, pp. 919–935.

Tang, H. et al. (2023), « A comprehensive survey of complex brain network representation », *in*: *Meta-radiology* 1.*3*, p. 100046.

Tang, S. et al. (2021), « Self-Supervised Graph Neural Networks for Improved Electroencephalographic Seizure Analysis », *in*: *arXiv e-prints*, arXiv–2104.

Tian, C. and F. Zhang (2025), « EEG-based epilepsy detection with graph correlation analysis », *in*: *Frontiers in Medicine* 12, p. 1549491.

Tran, M. A. et al. (2023), « EEG-Based Emotion Feature Extraction Using Power Spectral Density », *in*: *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1–4, DOI: 10.1109/KSE59128.2023.10299508.

Turnip, A. and E. Junaidi (2014), « Removal artifacts from EEG signal using independent component analysis and principal component analysis », *in*: *2014 2nd International Conference on Technology, Informatics, Management, Engineering & Environment*, IEEE, pp. 296–302.

Tutte, W. T. (1963), « How to draw a graph », *in*: *Proceedings of the London Mathematical Society* 3.*1*, pp. 743–767.

Van Erp, J., F. Lotte, and M. Tangermann (2012), « Brain-computer interfaces: beyond medical applications », *in*: *Computer* 45.*4*, pp. 26–34.

Van Vliet, M. et al. (2012), « Designing a brain-computer interface controlled video-game using consumer grade EEG hardware », *in*: *2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*, IEEE, pp. 1–6.

Vermeille, H. (2002), « Direct transformation from geocentric coordinates to geodetic coordinates », *in*: *Journal of Geodesy* 76.*8*, pp. 451–454.

Vidal, J. J. (1973), « Toward direct brain-computer communication », *in*: *Annual review of Biophysics and Bioengineering* 2.*1*, pp. 157–180.

Vilamala, A., K. H. Madsen, and L. K. Hansen (2017), « Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring », *in*: *2017 IEEE 27th international workshop on machine learning for signal processing (MLSP)*, IEEE, pp. 1–6.

Vrbancic, G. and V. Podgorelec (2018), « Automatic classification of motor impairment neural disorders from EEG signals using deep convolutional neural networks », *in*: *Elektronika ir Elektrotechnika* 24.*4*, pp. 3–7.

Wang, X. et al. (2024), « An in-depth survey on deep learning-based motor imagery electroencephalogram (EEG) classification », *in*: *Artificial intelligence in medicine* 147, p. 102738.

Wang, Z. et al. (2018), « Short time Fourier transformation and deep neural networks for motor imagery brain computer interface recognition », *in*: *Concurrency and Computation: Practice and Experience* 30.*23*, e4413.

Wei, L., T. Qiu, Z. Wang, and W. Mei (2025), « A functional connectivity metric method for EEG time series via nonlinear symbolization », *in*: *Biomedical Signal Processing and Control* 103, p. 107498.

Wei, X. et al. (2018), « Automatic seizure detection using three-dimensional CNN based on multi-channel EEG », *in*: *BMC medical informatics and decision making* 18.*Suppl 5*, p. 111.

Wu, A., Y. Zhang, Y. Yu, and L.-L. Zeng (2024), « EEG-ARNet: An Autoregressive Pretraining Model for Extracting EEG Features », *in*: *2024 5th International Conference on Computers and Artificial Intelligence Technology (CAIT)*, pp. 255–259, DOI: 10.1109/CAIT64506.2024.10962980.

Wu, S.-L. et al. (2013), « Common spatial pattern and linear discriminant analysis for motor imagery classification », *in*: *2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*, IEEE, pp. 146–151.

Wu, W., X. Gao, and S. Gao (2006), « One-versus-the-rest (OVR) algorithm: An extension of common spatial patterns (CSP) algorithm to multi-class case », *in*: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, IEEE, pp. 2387–2390.

Wu, Z. et al. (2020), « A comprehensive survey on graph neural networks », *in*: *IEEE transactions on neural networks and learning systems* 32.*1*, pp. 4–24.

Xu, T., W. Dang, J. Wang, and Y. Zhou (2023), « DAGAM: a domain adversarial graph attention model for subject-independent EEG-based emotion recognition », *in*: *Journal of Neural Engineering* 20.*1*, p. 016022.

Yanagimoto, M. and C. Sugimoto (2016), « Recognition of persisting emotional valence from EEG using convolutional neural networks », *in*: *2016 IEEE 9th International*

*Workshop on Computational Intelligence and Applications (IWCIA)*, IEEE, pp. 27–32.

Yger, F., M. Berar, and F. Lotte (2016), « Riemannian approaches in brain-computer interfaces: a review », *in*: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.*10*, pp. 1753–1762.

Zhang, C., D. Florêncio, and P. A. Chou (2015), « Graph signal processing-a probabilistic framework », *in*: *Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2015-31*.

Zhang, K. et al. (2020), « Application of transfer learning in EEG decoding based on brain-computer interfaces: a review », *in*: *Sensors* 20.*21*, p. 6321.

Zhang, Y. et al. (2018), « Temporally constrained sparse group spatial patterns for motor imagery BCI », *in*: *IEEE transactions on cybernetics* 49.*9*, pp. 3322–3332.

Zhang, Z., Y. Li, H. Song, and H. Dong (2021), « Multiple dynamic graph based traffic speed prediction method », *in*: *Neurocomputing* 461, pp. 109–117.

Zhao, X. et al. (2019), « A multi-branch 3D convolutional neural network for EEG-based motor imagery classification », *in*: *IEEE transactions on neural systems and rehabilitation engineering* 27.*10*, pp. 2164–2177.

Zhdanov, M., S. Steinmann, and N. Hoffmann (2022), « Investigating brain connectivity with graph neural networks and GNNExplainer », *in*: *2022 26th International Conference on Pattern Recognition (ICPR)*, IEEE, pp. 5155–5161.

Zhong, P., D. Wang, and C. Miao (2020), « EEG-based emotion recognition using regularized graph neural networks », *in*: *IEEE Transactions on Affective Computing* 13.*3*, pp. 1290–1301.

COLLEGE SCIENCES
DOCTORAL DE L'INGENIERIE
PAYS DE LA LOIRE ET DES SYSTEMES

CENTRALE NANTES

**Titre :** Représentations basées graphes incluant l'information spatiale pour la classification des signaux EEG.

**Mot clés :** Électroencéphalographie, connectivité fonctionnelle, traitement du signal sur graphe, réseaux de neurones sur graphes, classification, charge mentale, imagerie motrice mentale.

**Résumé :** La connectivité fonctionnelle, estimée à partir de l'électroencéphalographie (EEG), constitue un outil privilégié pour analyser les interactions neuronales grâce à sa haute résolution temporelle et à son caractère non invasif. La modélisation de ces signaux par des graphes est apparue récemment comme une approche prometteuse, mais la plupart des travaux existants négligent encore l'intégration explicite de la configuration spatiale des électrodes. Cette thèse vise à développer des représentations basées sur les graphes pour la classification de la connectivité fonctionnelle à partir de signaux EEG. L'originalité des approches proposées réside dans l'intégration conjointe de la disposition spatiale des électrodes, et des motifs de connectivité. Ces méthodes sont appliquées dans le cadre de classification de la charge mentale et de l'imagerie motrice dans des interfaces cerveau ordinateur (ICO). Plus précisément, nous proposons (i) une représentation sous forme d'images de la connectivité fonctionnelle, intégrant à la fois la position, le degré et la connectivité des électrodes, permettant une classification par réseaux convolutifs (CNN) ; (ii) une représentation vectorielle fondée sur des forces d'attraction et de répulsion entre électrodes, utilisée avec des classifieurs classiques tels que les machines à vecteurs de support (SVM) ; et (iii) l'exploration de différentes approches de graphes traités par réseaux de neurones sur graphes (GNN), combinant informations spatiales et diverses mesures de distance, afin d'identifier la représentation la plus adaptée pour ce type de modèles.

**Title:** Spatial-aware Graph-based Representations for EEG Signals Classification.

**Keywords:** Electroencephalography, functional connectivity, graph signal processing, graph neural networks, classification, mental workload, mental motor imagery.

**Abstract:** Functional connectivity, estimated from electroencephalography (EEG), is a useful tool for analyzing neural activity due to its high temporal resolution and non-invasive nature. Modeling EEG signals using graphs has emerged as a promising approach; however, most existing studies still overlook the electrode spatial configuration in graph construction. This thesis aims to develop graph-based representations for the classification of functional connectivity from EEG signals. The originality of the proposed approaches lies in the joint integration of electrode spatial arrangement and connectivity patterns. These methods are applied to the classification of mental workload and motor imagery within brain–computer interface (BCI) systems. More specifically, we propose: (i) an image-based representation of functional connectivity that incorporates electrode position, degree, and connectivity, enabling classification with convolutional neural networks (CNNs); (ii) a vector-based representation grounded in attractive and repulsive forces between electrodes, combined with classical classifiers such as support vector machines (SVM); and (iii) the exploration of various graph types classified with graph neural network (GNN), leveraging spatial information and different distance measures, to identify the most suitable graph representation for such models.