GRAPH SIGNAL REPRESENTATIONS FOR EEG ANALYSIS AND MACHINE LEARNING CLASSIFICATION

A thesis submitted for the degree of Doctor of Philosophy

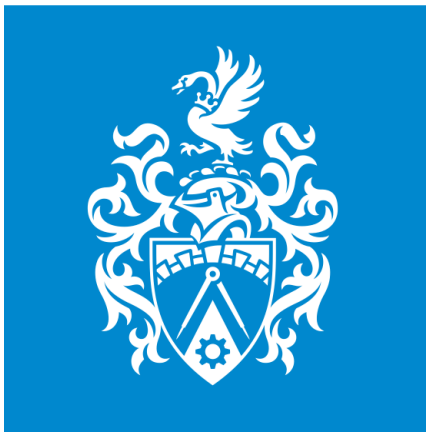
by

Cengiz Selçuk

College of Engineering, Design and Physical Sciences

Department of Electronic and Electrical Engineering

Brunel University London



4 June 2025

Abstract

This thesis investigates the dynamic characteristics of electroencephalography (EEG) signals using graph signal processing (GSP), focusing on modeling spatial and temporal variations in brain connectivity. Conventional approaches often rely on static graph structures, which do not adequately capture time-varying, task-specific, and subject-specific relationships among brain regions. To address these limitations, this thesis proposes a dynamic graph representation framework for EEG signals. This framework allows EEG signals to be represented with a broader set of spatial frequency components derived from multiple graphs.

A unified spatiotemporal frequency representation is developed to reflect how spatial patterns evolve over time by combining temporal frequency components with graph-based spatial frequencies. The proposed methods are evaluated in the contexts of imagined speech classification and biometric identification. The effects of clustering strategies and eigenvector selection are systematically examined, and their influence on classification performance is demonstrated.

The thesis also introduces a graph-based EEG data augmentation method designed to preserve channel-wise correlations while generating artificial EEG trials. The feasibility of using dynamic graph representations for biometric identification is further evaluated. Experimental results demonstrate that individual differences in EEG graph representations enable reliable subject identification, achieving 100% classification accuracy.

Dedication

To my mother, for her patience, boundless love, and remarkable skill in pretending to understand every word, making it seem like it all made sense.

Declaration

I declare that all the work presented in this thesis is my own, and none of it has been submitted for any other degree or qualification at any institution.

Acknowledgements

I would like to express my gratitude to my doctoral advisor, Dr. Nikolaos Boulgouris, for his guidance, patience, and invaluable support throughout the course of this research. I am also deeply thankful to my family for their unwavering encouragement and understanding during this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Interacting with devices and systems generally requires muscle movements. Whether one is typing on a keyboard to control a computer, using the steering wheel of a car, interacting with the touchscreen of a smartphone, or using a television remote to rewind an on-demand TV show, these actions all rely on physical interactions. Translating thoughts into commands that devices can interpret requires muscle movements to generate corresponding responses. In most cases, the hands serve as the main interface for interacting with devices. This constraint limits human productivity because the hands are often already engaged in other tasks. Even performing a quick internet search on a smartphone requires physical proximity to the device and manual interaction. This process is considerably slower than the speed of thought and does not permit the execution of tasks in parallel. Voice assistant applications partially mitigate these limitations. However, they do not fundamentally address the problem. Instead, they enable users to issue basic natural language commands for simple actions, such as initiating a call to a contact listed in a mobile phone.

Developing Brain-Computer Interfaces (BCIs) is an active research area in which various data types, such as speech imagery and motor imagery, are investigated to assess their applicability for enabling direct interaction with devices and systems without muscular involvement. Achieving device control at the speed of thought, without muscle movements, and with the ability to execute tasks in parallel remains an ambitious goal that is beyond the reach of current technology. However, these aspects are being investigated separately, and this reflects the direction in which the research area is progressing.

An important aspect of data collected from brain signals is the spatial relationships among different brain regions. Since the data does not lie in Euclidean space and exhibits a non-Euclidean structure, as explained in Section 1.3, treating it as if it were defined on a regular grid structure or as spreadsheet-style data neglects the spatial relationships and interconnectivities across different brain regions. For these reasons, it demands different tools to analyze not only temporal relationships in different regions but also spatial relationships, and it is essential to use the right tools to take those relationships

into account.

Signal processing on graphs has received significant attention from researchers over the last 10 years, as more and more data are collected, and new tools and data structures are required to analyze and process these data. Social networks, road maps, biological networks, infrastructure networks, chemical structures, and computer networks can all be represented with graph structures. Graph Signal Processing (GSP) is the effort to carry the knowledge and intuition of the well-established area of Signal Processing (SP) to the graph domain. One of the data types that can benefit from GSP is brain activity data, as it also resides on a non-Euclidean domain, which we explain in Section 1.3. This work focuses on applying GSP to brain activity data to analyze and model brain waves and interconnectivities of different regions of the brain, in order to propose a model for representing the network. Instead of analyzing data captured from different regions of the brain separately and analyzing the signals in the traditional way, GSP provides a framework to analyze the signals defined on a network structure through graph signals and graph frequencies. This is particularly important, as different regions of the brain are responsible for different tasks, and depending on the task, different regions exhibit different levels of activity. Thus, network activity changes depending on the task.

1.1 Graph Signal Processing

Every day, billions of people around the world generate data, even if they engage in no deliberate activity beyond mere existence. Credit scores, social media presence or absence, participation in healthcare systems, expenses, Wi-Fi routers detected by mobile phones during casual movements, emails, photos and numerous other factors collectively contribute to the big volumes of data collected globally. As a result of technological advancements over the past three decades, both the volume of data and the computational power to collect and process it have reached to unprecedented levels. On the other hand, these tools and advances are relatively recent. Further advancements and methodological developments are needed to represent, process, visualize, cluster and analyze those data to gain insight into what they represent. For these reasons, academic fields focused on big data, networks, and signal processing are expected to have a significant role in the development of analytical tools for big data and data residing on irregular domains.

Analyzing and processing the data generated by billions of people around the world requires sophisticated data structures and advanced tools [1]. Data arising from complex network structures reside on a non-Euclidean domain [2], and it is not feasible to treat these data as if it were tabular data. The complexity of big data requires a different approach [3]. Graphs can be used to represent the complex network structures and relationships between different entities.

Conventional *signal processing* (SP) provides a well-established theory to work over

continuous or discrete signals that are defined over time or space. The applications of SP are everywhere in our lives through communication, image processing, control, sensing and many others. Unlike the regular domains that SP deals with, much of today's data reside on a non-Euclidean domain, and classical tools have shortcomings to deal with this non-regular, complex nature of the data [4].

Graphs are mathematical tools for encoding pairwise relationships among different objects to model network structures [5, 6]. Graphs can model the relationships between people, social networks, academic collaborations, infrastructure networks, biological networks, chemical structures, road maps and any other network where we can represent the networks as pairwise connections between different objects [7, 8]. Academic fields like *network science* and *network processes* analyze the graphs themselves [9, 10]. Their focus is not analyzing the signals on graphs. GSP brings the knowledge and intuition from the well-established area of SP to the graph domain to process the signals that are defined on graphs [10, 11]. The need for using graphs as mathematical structures to represent data came after an enormous amount of data was collected with the advancements in technology [1]. Representing big data and processing its complex nature require complex data structures [2].

Graph theory was first investigated by the Hungarian mathematician Dénes Kőnig [12] during the 1930s. Compared to graph theory, GSP has a short history, having emerged in the 2010s with the publication of [4, 13, 14] and [15]. It became popular in both the SP and *machine learning* (ML) communities [11].

If we think about the main reasons behind the advancements of those tools, we can count two major advancements that happened in the last 35 years, namely *the internet* and *general-purpose GPU computing*. The internet became mainstream at the start of the 1990s and there is no need to mention how it became the main source of data collection. Another important advancement in technology was graphics cards and how they evolved to be used for general purpose computing. Graphics cards were initially designed to be used for playing computer games. Most of the objects in computer games and other graphics-heavy applications require matrix calculations. How those objects were presented, their positions and their movements were all represented using vectors and matrices. Obviously, they were not limited to just vectors and matrices. Vectors and matrices are one-dimensional and two-dimensional instances of tensors, and graphics cards were designed to make calculations on tensors. Graphics cards do those linear algebra operations fast and efficiently in parallel. At the end of the 1990s, researchers started using graphics cards to accelerate scientific computations. Tensors are in general the main data structure for scientific computing, therefore graphics cards were perfect devices to be used for scientific calculations. So, the devices that were first designed to make fast linear algebra operations for computer games, started to be used for accelerating scientific computing, which was called *general purpose GPU computing*. At the end of 2000s, with

the release of the CUDA platform by Nvidia, it became easy to directly use graphics cards for general-purpose GPU computing, which not only accelerated the linear algebra calculations, but also accelerated the research that depends heavily on linear algebra. The internet provided the tools to collect data from billions of users around the world and graphics cards provided tools to make linear algebra operations on those big datasets to process and analyze the data fast and in parallel.

As more and more data were collected, advancements to improve the representation of the data became necessary. The collected data that we analyze is not basic tabular data that can be checked in a spreadsheet application like Excel. Not only is processing the data difficult, but even just visualizing it is a challenge [4, 10]. The work on graphs and processing signals on graphs therefore became crucial to address the issue of representing the complex nature of the collected data and processing signals on graphs that represent complex network architectures.

Although the tools of GSP are crucial to process and represent big datasets [1], they are not limited to big data. Noticeably, in any application where spatial relationships between different entities matter, the tools of GSP can be effectively leveraged. One important use case of this is EEG signals [10].

1.2 Electroencephalography

Electroencephalography (EEG) is a non-invasive method to record postsynaptic potentials using the electrodes placed on top of a subject's scalp [16, 17]. EEG data is a multi-channel signal that is recorded with multiple EEG sensors, which are placed at different parts on top of the scalp. Therefore, each channel collects signals from a different part of the brain. When analyzing this data, spatial relationships between those channels are important as they represent the relationships between different regions of the brain. Each channel can have different characteristic and exhibit different activities across different frequency bands, and analyzing those channels can give insight into the undertaken task [18]. However, analyzing those channels separately ignores the channel-wise relationships, and therefore neglects the relationships between different regions of the brain. There is a lot of research on EEG applications [19–24], however most of the research in EEG applications does not take the spatial relationships into account. Graph modeling for brain signals acquired in EEG has been used in several past works [25–30], but they were used separately to get the spatial relationships without incorporating them with temporal features. GSP provides spatial frequencies, but these are calculated for a fixed time instant [4, 10, 14]. This means we get the spatial relationship only for a single time instant. However, these relationships evolve through time, and they change through task. Calculating both temporal and spatial frequencies together is vital to analyze the signals, how different regions of the brain network change through the time for different

tasks.

The main advantage of GSP is representing spatial features of signals on graphs. The conventional tools of SP give us tools to analyze the temporal frequencies of a system [31–33]. On the other hand, GSP provides us with the tools to analyze the spatial frequencies of a system [4, 34, 35]. Neither of them alone can provide both temporal and spatial frequencies together, so there is a tradeoff when using one or the other [9]. For any multi-channel signal, when we use GSP to analyze spatial frequencies, we can only calculate them for a fixed time instant, and when we analyze time frequencies of a signal using SP, we don't factor in the spatial relationships between neighboring channels. Some works use spatial features with temporal features as in [28], however, they were used as separate inputs for machine-learning-based classification methods. This again does not analyze the spatiotemporal relationships between different channels. Instead, it gives them as separate inputs to a neural network for classification tasks without providing a unified representation of spatial and temporal features.

1.3 Geometry of EEG Data

The terms *"Euclidean"* and *"non-Euclidean"* frequently appears in relevant literature [2–4] and will recur throughout this text. It is important to clarify why EEG data are considered as an example of non-Euclidean data [2]. If we summarize Euclid's five postulates [36]:

1. Any two points can be connected with a straight line segment.
2. Any straight line segment can be extended indefinitely as a straight line.
3. A circle can be drawn using any straight line segment as the radius and one of its endpoints as its center.
4. Every right angle is congruent to any other right angle.
5. If two lines intersect a third line such that the sum of the inner angles on one side is less than two right angles, those two lines will intersect on that side if the lines are extended far enough.

Euclidean geometry refers to the geometry that satisfies all those postulates. Euclid's fifth postulate [36, 37], also known as the *"parallel postulate"*, was not proven by mathematicians even though it appears self-evident, as illustrated in Figure 1.1. Mathematicians questioned whether it is fundamental and tried to derive it from the first four but they couldn't prove it, and later it was shown that alternative geometries can be created by altering the fifth postulate. The term *"non-Euclidean geometry"* was used by

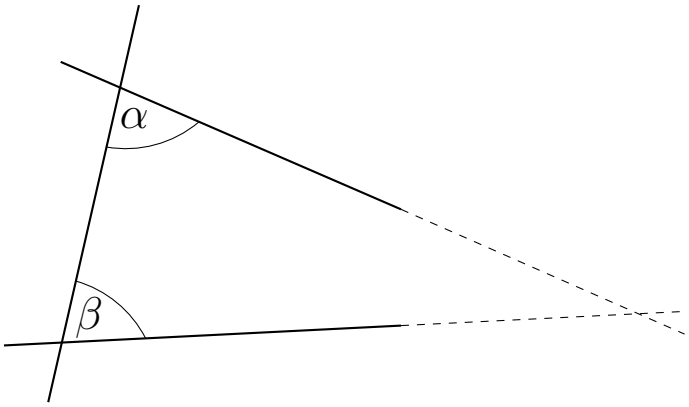


Figure 1.1: Euclid's fifth postulate (figure created by the author).

Gauss to refer to a geometry that does not use the fifth postulate [37]. Two mathematicians, János Bolyai from Hungary and Nikolai Lobachevsky from Russia, independently developed such a system, which is called Bolyai-Lobachevsky geometry, during the nineteenth century [37]. Later, famous mathematicians Bernhard Riemann from Germany and Ludwig Schläfli from Switzerland, proposed another system as an alternative. In 1871, the German mathematician Felix Klein systematized those variants and gave the names *"parabolic"*, *"hyperbolic"*, and *"elliptic"* for the systems of Euclid, Bolyai-Lobachevsky, and Riemann-Schläfli geometries [37].

A good illustration of hyperbolic geometry is the woodcut *Circle Limit III* [38], created by the Dutch artist M. C. Escher as shown in Figure 1.2. Strings of fish creates a tessellation of the hyperbolic plane where they get smaller as they reach to the edges of the circle which represents infinity. Escher, who was a family friend of prominent geometer H. S. M. Coxeter, was inspired by hyperbolic geometry and used it in his works [39].

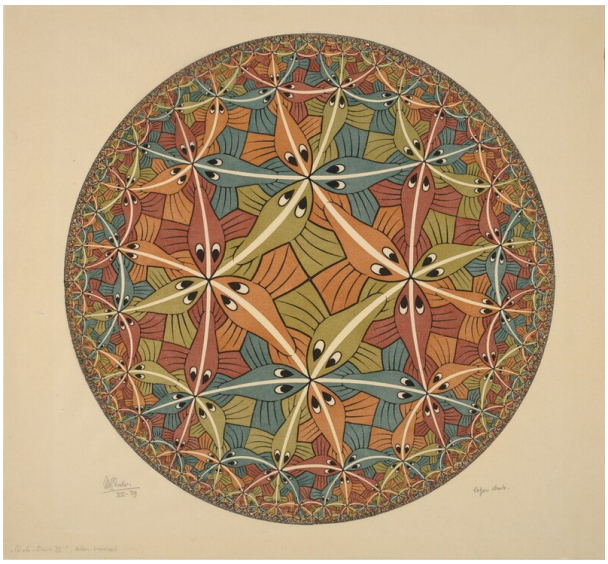


Figure 1.2: Escher's Circle Limit III [38].

Images are an example of Euclidean data where pixels are arranged in a grid-like structure. However, if we think about EEG data, which is a set of sensors placed on top of the scalp, the geometry of the data changes as it is not on a plane any more. To give an example, Figure 1.3 illustrates an octant of a sphere, representing one of its eight equal parts. The spherical triangle shown has three right angles, which is impossible in Euclidean geometry. This becomes more intuitive when considering the Earth. If we start from the North Pole and travel along a meridian heading south until we reach the equator, then turn east or west along the equator at a right angle, and go straight for a quarter of the length of the equator, later turn again at a right angle to go north along another meridian, we would eventually return to the starting point, forming a spherical triangle with three right angles.

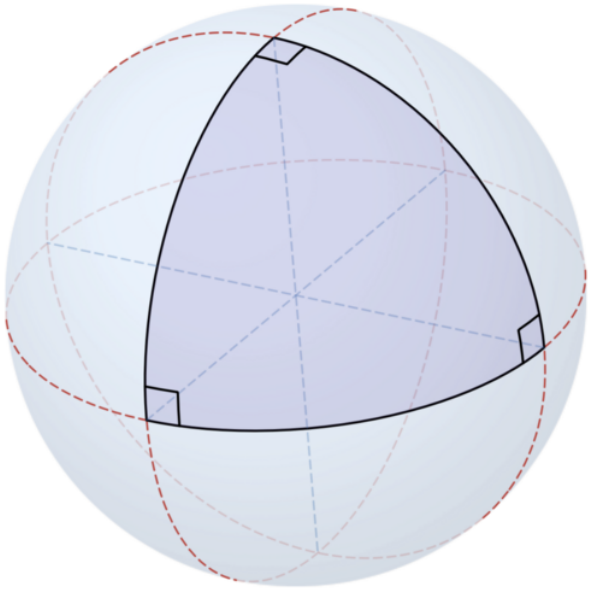


Figure 1.3: Octant of a sphere [40].

1.4 Imagined Speech

There are different data sources and scenarios for BCI implementations. For a BCI system, different types of data can be used such as imagined speech or motor imagery. In imagined speech, subjects imagines speaking a word, a vowel, a consonant or any other sound without making any movements. In motor imagery, subjects imagine moving their muscles, i.e. moving their feet, without actually making any movements. The goal of a BCI application is to process and analyze signals from a subject's brain to interact with computers or other devices, without depending on muscle movements. Imagined speech is one of these applications where brain activity data collected during subjects imagine

speaking a word or making a sound without actually speaking or moving their muscles, and the collected data is used for various control applications.

Creating imagined speech signals has some advantages over motor imagery signals. When subjects imagine moving their limbs or making a sound, same regions of their brain becomes active and signals created through imagination can interfere with the signals created by actual movements. For example, when driving a car, a subject is making actual movements to control their car while they are also trying to control another application through imagining making some limb movements, it will be difficult for the subject to imagine making different limb movement than the actual movements and it would also be difficult to analyze and discriminate the signals created by actual movements and signals created by imagining movements. This would limit the subject's mobility while using a BCI system. On the other hand, imagined speech would allow the subject to still use the car and imagine speaking a word or making a sound to control another device.

For silent communication cases where subjects need to communicate without making any sound, this again wouldn't be a constraint as the subject will not be speaking while using an imagined-speech-based BCI system. For example, in a combat scenario where a soldier quietly moving with a rifle while using a BCI system to communicate with other soldiers, it would be favorable to allow the subject to freely use their limbs to engage with their arms and surroundings. When they are supposed to be quiet but need to communicate, an imagined-speech-based BCI system would allow them to effectively communicate without revealing their position and without limiting their mobility.

BCI systems are also crucial for the cases of people with brain damage or amyotrophic lateral sclerosis (ALS) where the subjects are "locked-in" as they still have their cognitive activity but cannot control their muscles [41–43].

1.5 Motivation

It is important to emphasize that studying the nervous system and brain function falls outside the scope of this thesis. That would not be appropriate, and any conclusions would lack credibility, as such an investigation would require expertise from a different area. What we are interested in here is the network structure of the captured brain signals and representing the relationships between signals captured from different regions of the brain.

Analyzing different channels of EEG signals separately provides information about the area each sensor is located on. However, it is also important to consider how those signals change relative to other brain regions, and how those pairwise relationships change relative to other pairwise relationships between nodes. The focus of this work is the mathematical tools used to model those relationships, and representing the signals defined on the network of different regions of the brain.

There are shortcomings in the current graph representation methods in the literature [10, 11]. Graphs are used to represent the network structure of the brain. However, those graphs are static; they do not change through time, task, or individual.

There are standard sensor placement methods [44, 45], and they exist to ensure very similar sensor placement across different individuals, so that the same sensors stand on top of the same brain regions. Using spatial proximities between sensors to create graphs result in constructing almost identical graph structures for every individual. Furthermore, using that static graph based on the spatial proximities of the sensors also results in every application using the exact same graph, regardless of the research area. Consequently, the same graph is used across all applications and subjects, and it does not change over time. This approach is problematic, as it neglects subject-wise and activity-wise dependencies.

To address these limitations, this thesis investigates the research questions listed in the next section, focusing on dynamic-graph-based representations of EEG data and their applications to imagined speech classification and biometric identification tasks, in order to address the dynamic nature of EEG data.

1.6 Research Questions

This thesis investigates the following research questions:

1. How can the dynamic nature of EEG data be modeled using graph-based representations that capture temporal and task-related variations in the relationships among different brain regions?

2. How can spatial and temporal variations in EEG signals be integrated to extract spatiotemporal frequency features from graph-based EEG representations?

3. Do dynamic-graph-based EEG representations enhance imagined speech classification performance in comparison to static graph models and current state-of-the-art methods?

4. How do clustering strategies and eigenvector selection methods affect the performance of dynamic-graph-based EEG representations in imagined speech classification tasks?

5. How do different neural network architectures affect imagined speech classification performance, and how can artificial EEG data be generated while preserving channel-wise relationships among different brain regions?

6. Can individual differences in dynamic-graph-based EEG representations be used for biometric identification in imagined speech tasks?

1.7 Aims

The aims of this thesis are:

1. To develop a dynamic-graph-based modeling strategy for EEG signals that captures evolving spatial and temporal relationships among different brain regions.
2. To evaluate the effectiveness of dynamic-graph-based EEG representations in imagined speech classification and biometric identification.

1.8 Objectives

The objectives of this thesis are:

1. To propose a method for constructing dynamic graph representations of EEG data that model the time-varying, task-specific, and subject-specific relationships among different brain regions, independent of their physical proximity.
2. To extract and analyze spatiotemporal frequency features from the constructed dynamic graph representations of EEG signals.
3. To apply the proposed method to imagined speech classification and evaluate its performance against state-of-the-art methods and conventional static graph approaches.
4. To investigate the effect of different clustering and eigenvector selection strategies on the performance of dynamic graph modeling of EEG signals.
5. To investigate the impact of different neural network architectures on imagined speech classification performance, and to propose a method for generating artificial EEG data while preserving channel-wise relationships.
6. To evaluate whether individual differences in dynamic graph representations can be used for biometric identification in imagined speech tasks, and to compare the performance against state-of-the-art methods and conventional static graph approaches.

1.9 Contributions to Knowledge

1. A novel framework for dynamic graph representation of EEG signals is proposed, which captures time-varying, task-specific, and subject-specific inter-regional relationships, overcoming the limitations of static graph models.

2. A unified spatiotemporal frequency representation is introduced, enabling joing analysis of spatial and temporal dynamics of EEG data across different brain regions.

3. The proposed dynamic graph framework is applied to imagined speech classification, achieving superior performance compared to existing state-of-the-art methods and static graph-based approaches.

4. A comprehensive evaluation of clustering strategies and eigenvector selection methods is conducted, providing insights into their effect on classification performance.

5. A graph-based EEG data augmentation method is developed, preserving inter-channel relationships while increasing robustness in training neural networks.

6. The discrimination ability of dynamic graph representations is demonstrated in biometric identification tasks using EEG, achieving 100% classification accuracy across two different datasets.

7. A critical evaluation is provided regarding the limitations of EEG-based biometric systems for real-world deployment, outlining practical considerations for future research and applications.

1.10 Publications

The following publication has resulted from the work presented in this thesis and has been submitted for publication.

- Cengiz Selcuk and Nikolaos V Boulgouris, "Dynamic Graph Representation of EEG Signals for Imagined Speech Recognition," *Journal of Neural Engineering*, submitted, May 2025.

The following publication is also based on the work presented in this thesis and will be submitted for publication.

- Cengiz Selcuk and Nikolaos V Boulgouris, "Dynamic Graph Representation of EEG Signals for Biometric Identification", to be submitted, July 2025.

1.11 Structure of the Thesis

This thesis is structured into six chapters. Each chapter builds on the previous one, beginning with the theoretical foundations and literature review, progressing through the development of a dynamic graph representation methodology and its applications, and concluding with a summary of contributions, findings, open issues, and future directions.

Chapter 2, Literature Review, presents the theoretical foundations used in this work and shares the relevant details. The chapter provides a thorough analysis of related work, identifies the literature gap, explains how this work addresses the research gap, outlines its expected impact, and describes the methodological approach. This chapter also presents the basics of GSP theory and explains the necessity of GSP tools, their relevance to this work, and how they can be applied to EEG data.

Chapter 3, Dynamic Graph Representations of EEG Signals, presents a novel method of dynamic graph representations of EEG signals. The chapter illustrates how spatial variances change with different spatial frequencies, addresses the shortcoming of GSP, how GSP represents spatial frequencies only at a fixed time instant, and why this lacks the ability to capture spatiotemporal variations. The chapter also addresses the problem of using fixed Euclidean distances between EEG sensors and how this leads to almost identical graph structures across EEG applications for calculating graph frequencies. This chapter also discusses how the correlations between different brain regions vary through time and task, and why this is problematic.

Chapter 4, Machine Learning Architectures and Data Augmentation for EEG Signal Classification, applies the proposed dynamic graph representation theory to imagined speech classification. A thorough investigation is conducted to explore the effect of different numbers of clusters, different eigenvector selection strategies, and different neural network variations. The chapter presents experiment results reflecting the outcomes of these different experimental setups. Two baseline models, one without any graph operators and one with *independent component analysis* (ICA), are included to control how much of the accuracy and performance can be attributed to the dynamic graph representations. Additionally, the lack of EEG data availability, the challenges of creating big EEG datasets, and the necessity of data augmentation in EEG applications are discussed. The chapter also addresses the problem of preserving the channel-wise relationships between different channels, which is a critical constraint in the generation of artificial EEG data.

Chapter 5, Graph Signal Processing for Biometric Identification, applies the dynamic graph representation theory for biometric identification using EEG data. Two different imagined speech databases are used to identify the subjects from their EEG signals. Furthermore, limitations of current state-of-the-art methods in the literature and obstacles to making this a real-life application are discussed. The chapter addresses the difficulties of implementing EEG-based subject identification in real-life scenarios and discusses its shortcomings compared to other biometric identification methods, such as fingerprint-based subject identification.

Chapter 6, Conclusion, presents the concluding remarks on GSP and EEG applications, summarizes the findings of this thesis, and outlines open issues and potential future directions that can follow this work.

Chapter 2

Literature Review

Although *graph signal processing* (GSP), *electroencephalography* (EEG) and its applications, *machine learning* (ML) and other related areas are vital parts of this thesis, their details are way beyond the scope of this work. So, while we give a quick overview about the main concepts related to those areas, we keep them brief so that this work is neither a repetition of very well-known details of those very popular subjects, nor unnecessary details beyond the scope of this work. As we mentioned in Chapter 1, studying the nervous system and brain function also falls outside the scope of this thesis, as that would require the expertise of a different area. What we are interested is mainly the mathematical tools to represent the network structure and pairwise relationships between different regions of brain, and using those mathematical tools to investigate how to incorporate spatial and temporal variations of those graph signals.

We briefly, go over GSP and EEG to provide relative information that is required to understand the next chapters, and why we use GSP and EEG instead of other signal processing and data recording methods, how they are required and relative to our work. We briefly go over the common feature extraction methods even though we do not rely on them to show why they are not reliable to use instead of dynamic graph representations and why we do not depend on them. We later present relative machine learning architectures that we use in this work without focusing on the details of machine learning itself and what the advantages of different architectures are. We also go over different validation schemes as it is crucial to follow a really specific, thorough validation scheme to work on very limited data. Most important downside of working on EEG data is the size of the datasets. It is difficult to create large EEG datasets, and therefore the test results have too much bias. To have a strict validation scheme can give almost unbiased results and we give those vital details of machine learning algorithm validation schemes for small EEG datasets. Finally, we provide the details of related work in literature and refer to what they accomplished and why they are important in the related work section.

2.1 Graph Signal Processing

Graphs are mathematical entities to represent pairwise relationships between different objects to model network structures [5, 6]. Graphs have been used to represent social networks, road maps, biological networks, infrastructure networks, chemical structures, computer networks, as these can be represented with graph structures [7, 8]. Graphs give us the ability to store those relationships and also visualize the relations between different objects [2]. We can store the relations in a graph model and each connection can have a direction and weight.

For example, connections of airports can be modeled with graphs. One way to model it is to treat each airport as a vertex and if there is a flight between two airports there can be a connection between those vertices. In small cities there will not be many flights to different airports and instead they will be connected to a hub airport on a bigger city and there will be more flights from the hub airport to bigger cities. In that setup, small cities will have less connections and hub airports will have more connections. In the same way social networks can be modeled as each user can be a vertex and if one user follows another one there can be a vertex from one user to another and that relationship can be one way or two-ways [4, 9, 10]. Also drawing graphs by itself is a subject in graph theory as there are multiple ways to plot the same graph; and two different graphs can be *isomorphic* which would make them the same but look drawn different [5].

Graph theory and other academic fields like *network science* and *network processes*, focus on graphs themselves but not the signals that residing on graphs [9, 10]. *Graph Signal Processing* (GSP) is the adaptation of the theory and intuition of the Signal Processing (SP) to the graph domain to analyze the signals defined on graphs.

Just as SP and its advancements through last 75 years [11], *graph theory* also have a long history as they were first investigated by Hungarian mathematician Dénes Kőnig [12] during 1930s. Compared to SP and the graph theory itself, GSP is very new as it emerged after 2010s. After 2010s, with the collection of enormous data and new computational advancements, new mathematical tools to represent and process the data that residing on non-Euclidean domain became a necessity [10, 46]. GSP, emerged after 2010s with the publication of [4, 13, 14] and [15]. It quickly became popular in both *signal processing* (SP) and *machine learning* (ML) communities [2, 11]. It addressed the issues of dealing with the complex data and carry the intuition from the well-established area of SP to the graph domain to process the signals that defined over irregular domains instead of regular domains like time and space.

A complex data structure residing on non-Euclidean domain [2, 3, 47] cannot be processed with the classical tools of SP [11]. EEG signals are such cases of those signals and GSP is the suitable tool to analyze and process the EEG data to factor in the network structure and spatial relationships of signals from different regions of the brain.

2.1.1 Algebraic Representation of Graphs

A graph $\boldsymbol{\mathcal{G}} = \{\mathcal{V}, \mathcal{E}\}$ can be described by its edges $\mathcal{E}$ and vertices $\mathcal{V} = \{v_0, v_1, ..., v_{N-1}\}$, where N is the cardinality of the vertex set. A *graph signal* $\boldsymbol{s}$ is an N-dimensional signal whose values at each time instant are indexed using a graph $\boldsymbol{\mathcal{G}}$ [9] as shown in Figure 2.1.

$$\boldsymbol{s} = \begin{bmatrix} s_0 & s_1 & \dots & s_{N-1} \end{bmatrix}^T \in \mathbb{C}^N \tag{2.1}$$

The adjacency matrix $\boldsymbol{A}$ is an $N \times N$ matrix that encodes the connections between the vertices of the graph. The element a_{ij}, $i, j = 0, 1, ..., N-1$, of the adjacency matrix indicates whether there is a connection from vertex j to i; a_{ij} will be 1 if there is a connection and 0 when there is no incoming edge [10]. If the graph is weighted, then $a_{ij} = w_{ij}$, where w_{ij} is the weight for that edge. If the graph is undirected, then $\boldsymbol{A}$ will be symmetric, with $a_{ij} = a_{ji}$ and $w_{ij} = w_{ji}$ [10].

Assuming $\boldsymbol{A}$ has a complete set of eigenvectors, spectral decomposition of $\boldsymbol{A}$ can be written as in (2.2) [6, 48]. The matrix $\boldsymbol{V}$ holds the eigenvectors of $\boldsymbol{A}$ on its columns as in (2.3) and the matrix $\boldsymbol{\Lambda}$ holds the eigenvalues of $\boldsymbol{A}$ on its diagonal [48] as shown in (2.4) [49, 50].

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1} \tag{2.2}$$

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_0 & \boldsymbol{v}_1 & \dots & \boldsymbol{v}_{N-1} \end{bmatrix} \tag{2.3}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_0 & 0 & 0 & \dots & 0 \\ 0 & \lambda_1 & 0 & \dots & 0 \\ 0 & 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_{N-1} \end{bmatrix} \tag{2.4}$$

Degree matrix $\boldsymbol{D}$ is a diagonal matrix as shown in (2.5) with each element d_{ii} on its diagonal showing the total number of incoming edges (or weights) to vertex i from other vertices [49].

$$\boldsymbol{D} = \begin{bmatrix} d_{00} & 0 & 0 & \dots & 0 \\ 0 & d_{11} & 0 & \dots & 0 \\ 0 & 0 & d_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d_{N-1N-1} \end{bmatrix} \tag{2.5}$$

As a_{ij} is an edge from vertex j to vertex i, each element on the diagonal of $\boldsymbol{D}$ shows the total number of incoming connections or total number of weights [10], and it has all 0s

on the off-diagonal elements as defined in (2.6).

$$d_{ij} = \begin{cases} \sum_{j=0}^{N-1} w_{ij} & i = j \\ 0 & i \neq j \end{cases} \tag{2.6}$$

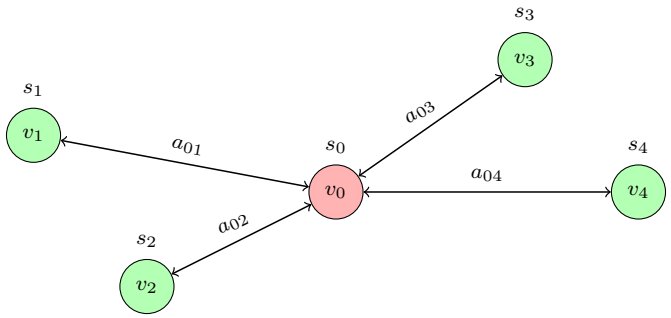


Figure 2.1: Signal $\boldsymbol{s} = [s_0, s1, s2, s3, s4]^\mathsf{T}$ is shown on graph $\boldsymbol{\mathcal{G}}$ with vertex set $\mathcal{V} = \{v_0, v_1, v_2, v_3, v_4\}$. Weight a_{ij} represents the weight for the edge from vertex j to vertex i (figure created by the author).

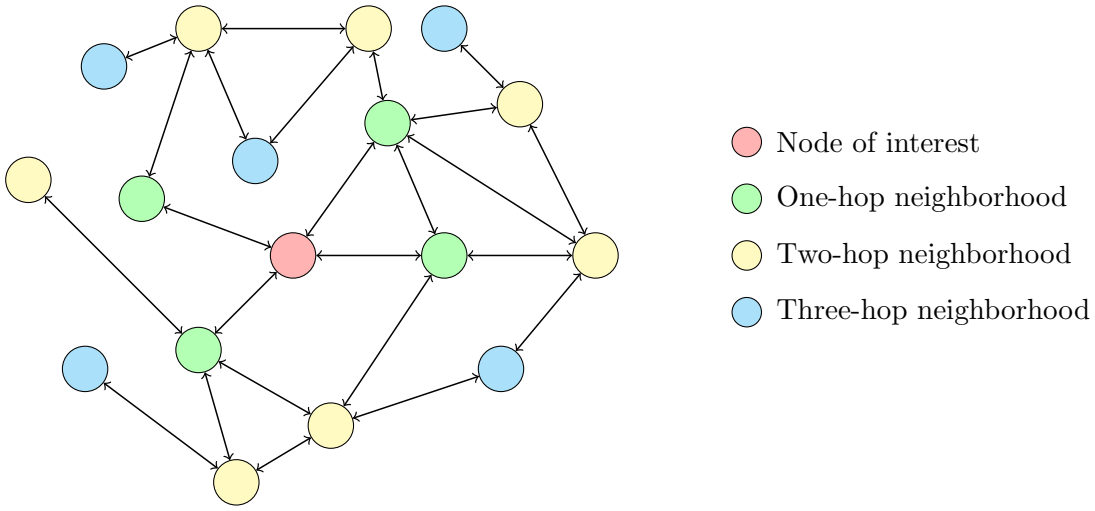


Figure 2.2: Vertex neighborhood (figure created by the author).

Other than encoding the connections of the vertices of the graph, the adjacency matrix $\boldsymbol{A}$ is an operator itself [6, 51]. When it is used on a graph signal, it provides the sum of the neighbor channels. This does not include the value at the vertex itself as long as there is no self-loop at that vertex, which makes diagonal values all zeros. So, for a graph signal $\boldsymbol{s}$, $\boldsymbol{As}$ provides the weighted sum of one hop neighborhood, $\boldsymbol{A}^2\boldsymbol{s}$ gives the weighted sum of the two hop neighborhood and so on [51], which can be seen from Figure 2.1 and Figure 2.2. Depending on the degree of the operator, the signal at a particular vertex propagates to its neighbors with the range assigned by the degree of the operator. As an operator, degree matrix $\boldsymbol{D}$ amplifies the signal at a vertex by the total weight around

the neighborhood [6], and the Laplacian matrix $\boldsymbol{L}$, which is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, gives a sense of variance for a vertex around its neighborhood [6, 52]. It also depends on the number of vertices in the vertex neighborhood as it is not normalized. Therefore, when the Laplacian operator applied to a graph signal, a vertex with higher degree will have a greater value when the signal at that vertex is different than the neighbor vertices, which gives a sense of variance.

We can get the eigendecomposition of the Laplacian matrix [10] as

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H \tag{2.7}$$

where $(\cdot)^H$ denotes the Hermitian operation and $\boldsymbol{U}$ is the matrix containing the eigenvectors of $\boldsymbol{L}$ in its columns [6, 48]. So, the Graph Fourier matrix is given by [6, 10]

$$\boldsymbol{F} = \boldsymbol{U}^H \tag{2.8}$$

and the graph Fourier transform of a graph signal $\boldsymbol{s}$, where the signal is indexed by the vertices of the graph $\boldsymbol{\mathcal{G}}$, can be calculated as [10]

$$\hat{\boldsymbol{s}} = \boldsymbol{F}\boldsymbol{s} \tag{2.9}$$

A vertex set $\mathcal{V} = \{v_0, v_1, ..., v_{N-1}\}$ of a graph can be labeled in any order, however, once it is set, it fixes the graph structure as there can be two graphs with different vertex labeling but with the same structure which would make them isomorphic graphs [5]. A graph signal $\boldsymbol{s}$ then can be defined as in (2.10) on the vertices of the graph [9, 10, 14].

$$\boldsymbol{s} = \begin{bmatrix} s_0 & s_1 & \dots & s_{N-1} \end{bmatrix}^T \in \mathbb{C}^N \tag{2.10}$$

2.1.2 Graph Operators

As in *signal processing* (SP), we can define a linear graph filter as a linear operator that acts on the input signal $\boldsymbol{x}$ to get the output $\boldsymbol{y}$ [10, 14, 53].

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \tag{2.11}$$

where $\boldsymbol{H}$ is an NxN linear graph filter. Below we summarize some of the one-hop operators and what they accomplish in the vertex domain [6].

Adjacency Matrix

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{2.12}$$

$$y(i) = \sum_{j \in N_i} a_{ij} x(j) \tag{2.13}$$

which is the sum over all the neighbors of vertex i. Note that the sum does not include the value at the vertex itself.

Random Walk Matrix If we average the sum [6],

$$y^{'}(i) = \frac{1}{d_i} \sum_{j \in N_i} a_{ij} x(j) \tag{2.14}$$

where $d_i = |N_i|$ if the graph is unweighted. More compactly [6]

$$\boldsymbol{y}^{'} = \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{x} \tag{2.15}$$

$$\boldsymbol{Q} = \boldsymbol{D}^{-\mathbf{1}}\boldsymbol{A} \tag{2.16}$$

$\boldsymbol{Q}$, random walk matrix, replaces the signal entry by the average of its neighbors' values.

Random Walk Laplacian If we average the Laplacian [6],

$$\boldsymbol{D}^{-1}\boldsymbol{L} = \boldsymbol{D}^{-1}\boldsymbol{B}\boldsymbol{B}^{T} \tag{2.17}$$

$$\boldsymbol{D}^{-1}(\boldsymbol{D} - \boldsymbol{A}) = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{Q} \tag{2.18}$$

$$\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{Q} \tag{2.19}$$

This can be used as a metric for frequency and signal smoothness [6].

$$\boldsymbol{T}\boldsymbol{x} = \boldsymbol{I}\boldsymbol{x} - \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x} \tag{2.20}$$

$$\boldsymbol{T}\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{Q}\boldsymbol{x} \tag{2.21}$$

$$y^{''}(i) = x(i) - \underbrace{\frac{1}{d_i} \sum_{j \in N_i} a_{ij} x(j)}_{Qx \text{ weighted average}} = x(i) - y^{'}(i) \tag{2.22}$$

$y^{''}(i)$ can be seen as a prediction error [6]. If $y^{''}(i)$ is close to the neighborhood average, then $y^{''}(i)$ will be close to zero [6].

If $|\boldsymbol{T}\boldsymbol{x}|$ is close to zero, then the signal is smooth, and it is low frequency and if it is large then the signal is non-smooth and it is high frequency [6].

Using $y^{''}(i)$ as a prediction error would be problematic as it does not depend on the number of neighbors and their weights [6].

Combinatorial Laplacian Instead of random walk Laplacian, we can define a weighted prediction error $z(i)$.

$$z(i) = d_i \left(x(i) - \frac{1}{d_i} \sum_{j \in N_i} a_{ij} x(j) \right) \tag{2.23}$$

or in matrix form

$$\boldsymbol{Z} = \boldsymbol{DTx} = \boldsymbol{D}(\boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A})\boldsymbol{x} = (\boldsymbol{D} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{Lx} \tag{2.24}$$

where $\boldsymbol{L}$ is the combinatorial graph Laplacian and $\boldsymbol{T}$ is the random walk Laplacian [6].

Symmetric Normalized Laplacian When we normalize the Laplacian, the normalization normalizes the rows of the operator and operator loses its symmetry [6].

$$\boldsymbol{D}^{-1}\boldsymbol{L} = \boldsymbol{D}^{-1}(\boldsymbol{D} - \boldsymbol{A}) = \boldsymbol{I} - \underbrace{\boldsymbol{D}^{-1}\boldsymbol{A}}_{\text{Loses symmetry}} = \boldsymbol{I} - \boldsymbol{Q} = \boldsymbol{T} \tag{2.25}$$

To preserve the symmetry, we can normalize the Laplacian from both sides, so the combinatorial graph Laplacian is defined as [34]

$$\boldsymbol{\mathcal{L}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} \tag{2.26}$$

and

$$\boldsymbol{\mathcal{L}} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{D}^{-1/2} = (\boldsymbol{D}^{1/2} - \boldsymbol{D}^{-1/2}\boldsymbol{A})\boldsymbol{D}^{-1/2}$$

$$\boldsymbol{\mathcal{L}} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \tag{2.27}$$

with

$$\boldsymbol{\mathcal{L}}_{ij} = -a_{ij}\frac{1}{\sqrt{d_i}\sqrt{d_j}},\ i \neq j \tag{2.28}$$

We can use $\boldsymbol{z}' = \boldsymbol{\mathcal{L}x}$ as a prediction error at vertex i

$$z'(i) = x(i) - \sum_{j \in N_i} \frac{a_{ij}}{\sqrt{d_i}\sqrt{d_j}} x(j) \tag{2.29}$$

Again, the prediction error does not depend on the size of the neighborhood as in random walk Laplacian ($\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{Q}$), but the weights have been normalized [6]. Also, unlike random walk Laplacian, prediction weights do not add to 1 [6].

As a summary of the graph operators [6],

$$\begin{aligned} y &= \boldsymbol{A}\boldsymbol{x} \quad \boldsymbol{A} : \text{Adjacency matrix} \\ y^{'} &= \boldsymbol{Q}\boldsymbol{x} \quad \boldsymbol{Q} : \text{Random walk matrix} \\ y^{''} &= \boldsymbol{T}\boldsymbol{x} \quad \boldsymbol{T} : \text{Random walk Laplacian} \\ z &= \boldsymbol{L}\boldsymbol{x} \quad \boldsymbol{L} : \text{Combinatorial Laplacian} \\ z^{'} &= \boldsymbol{\mathcal{L}}\boldsymbol{x} \quad \boldsymbol{\mathcal{L}} : \text{Symmetric normalized Laplacian} \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{D}^{-1}\boldsymbol{A} \\ \boldsymbol{T} &= \boldsymbol{I} - \boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A} \\ \boldsymbol{L} &= \boldsymbol{B}\boldsymbol{B}^{T} = \boldsymbol{D} - \boldsymbol{A} = \boldsymbol{D}\boldsymbol{T} \\ \boldsymbol{\mathcal{L}} &= \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2} \end{aligned}$$

where $\boldsymbol{B}$ is the incidence matrix.

2.1.3 Graph Frequency Domain

As the goal of GSP is to bring the knowledge and intuition of DSP to graph domain, it is important to describe filtering operations on graphs [54]. To have a parallelism between DSP and GSP, we can see the adjacency matrix as a shift operator [9, 10, 54] and define filter h as in (2.30).

$$s_{out} = h(z) \cdot s_{in} \tag{2.30}$$

As we described the graph signal $\boldsymbol{s}$ in (2.10), we can use the matrix notation to describe the filtering operation as in (2.31) [10, 54].

$$\boldsymbol{s}_{out} = \boldsymbol{H}\boldsymbol{s}_{in} \tag{2.31}$$

We are interested in signals that are eigenfunctions of the shift invariant filters, and we had the eigendecomposition of $\boldsymbol{A}$ as in (2.2) based on the assumption that $\boldsymbol{A}$ has N distinct eigenvalues and complete set of eigenvectors; we can show that eigenvectors of $\boldsymbol{A}$ are eigenfunctions of the filter $\boldsymbol{H}$ [10] as in (2.32).

$$\boldsymbol{H} = h(\boldsymbol{A}) = \boldsymbol{V}h(\boldsymbol{\Lambda})\boldsymbol{V}^{-1} \tag{2.32}$$

So, if we apply the filter $\boldsymbol{H}$ on n^{th} eigenvector v_n as in (2.33), we can see that indeed v_n is an eigenfunction of the filter $\boldsymbol{H}$ [9, 10, 14].

$$\boldsymbol{H}v_n = h(\boldsymbol{A})v_n = \boldsymbol{V}h(\boldsymbol{\Lambda})\boldsymbol{V}^{-1}v_n \tag{2.33}$$

$$\boldsymbol{H}v_n = h(\lambda_n)v_n \tag{2.34}$$

2.1.4 Graph Fourier Transform

If we apply the filter $\boldsymbol{H}$ on the input signal $\boldsymbol{s}_{in}$, we will have (2.35).

$$\boldsymbol{s}_{\text{out}} = \boldsymbol{H}\boldsymbol{s}_{\text{in}} = \boldsymbol{V}h(\boldsymbol{\Lambda})\boldsymbol{V}^{-1}\boldsymbol{s}_{\text{in}} \tag{2.35}$$

Graph Fourier transform (GFT) is defined as (2.36). Applying Fourier transform on input signal can be represented as a matrix multiplication (2.37). Then the inverse graph Fourier transform will be straightforward [6, 10, 51] as (2.38), and we can recover the signal with (2.39).

$$\boldsymbol{F} = \boldsymbol{V}^{-1} \tag{2.36}$$

$$\hat{\boldsymbol{s}} = \boldsymbol{F}\boldsymbol{s} = \boldsymbol{V}^{-1}\boldsymbol{s} \tag{2.37}$$

$$\boldsymbol{F}^{-1} = \boldsymbol{V} \tag{2.38}$$

$$\boldsymbol{s} = \boldsymbol{F}^{-1}\hat{\boldsymbol{s}} = \boldsymbol{V}\hat{\boldsymbol{s}} \tag{2.39}$$

We can see (2.35), from right to left with different operations on the input signal $\boldsymbol{s}_{\text{in}}$ as shown in (2.40). We can first take the graph Fourier transform, then we can filter the signal in graph frequency domain and then using inverse graph Fourier transform, we can turn it back to vertex domain [10].

$$\boldsymbol{s}_{\text{out}} = \boldsymbol{H}\boldsymbol{s}_{\text{in}} = \underbrace{\boldsymbol{V}\ \overbrace{h(\boldsymbol{\Lambda})}^{\text{Filtering in graph Fourier space}}\ \underbrace{\boldsymbol{V}^{-1}\boldsymbol{s}_{\text{in}}}_{\text{Fourier transform}}}_{\text{Inverse Fourier transform taking it back to vertex domain}} \tag{2.40}$$

In this sense, as the eigenvalues define the graph frequencies; we can define low-pass, high-pass and band-pass filters. Contrary to frequency definition on DSP, eigenvalues and therefore, frequencies of the graph are not easily interpreted with their values, which makes it difficult to carry out the intuition from DSP to GSP [9, 10]. Different eigenvalues have different spectral complexities and depending on the formulation on spectral variation, if eigenvalues are ordered, defining low-pass, high-pass and band-pass filters is straightforward [10].

2.1.5 Graph Laplacian and Laplacian Based Frequency Representation

As there are different ways to define graph frequencies and each has different advantages and disadvantages, frequency representation of graphs is still an active research area. Using adjacency matrix has the advantage of being defined for both directed and undirected graphs, and a directed circular graph can define the shift operation on the

vertices and taking the Graph Fourier Transform of a signal on circular graph is same as taking the DFT of the same signal as it is a time-series. In that setup, circular graph actually encodes the temporal aspect of the signal and with its spatial relations between its vertices.

Other than adjacency matrix, another popular way to define frequencies of graphs is based on graph Laplacians. Laplacian matrix is defined as in (2.41) and shown with its elements in (2.42). As the eigenvalues for spectral decomposition of Laplacian matrix (2.43) are ordered, it is more intuitive to interpret the graph frequencies and the first eigenvector is constant as DC component. Contrary to adjacency matrix, it is only defined for undirected graphs. Note that when we mention *Laplacian*, we refer to the *combinatorial graph Laplacian* $\boldsymbol{L}$. If we normalize the Laplacian, we can get the *symmetric normalized Laplacian* $\boldsymbol{\mathcal{L}}$, which is defined as in (2.26), however that would not capture the variance based on the degree of the vertex neighborhood, i.e. a vertex with small number of neighbors and a vertex with a greater number of vertices would have the same variance value as the variance would be averaged. For this reason, we will use Laplacian matrix as a metric to measure the variance of a vertex to define the spatial frequencies.

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \tag{2.41}$$

$$\boldsymbol{L} = \begin{bmatrix} d_{00} - a_{00} & -a_{01} & -a_{02} & \dots & -a_{0(N-1)} \\ -a_{10} & d_{11} - a_{11} & -a_{12} & \dots & -a_{1(N-1)} \\ -a_{20} & -a_{21} & d_{22} - a_{22} & \dots & -a_{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{(N-1)0} & -a_{(N-1)1} & -a_{(N-1)2} & \dots & d_{(N-1)(N-1)} - a_{(N-1)(N-1)} \end{bmatrix} \tag{2.42}$$

Since the Laplacian matrix is symmetric semidefinite matrix, all its eigenvalues are real and non-negative. If we get the eigendecomposition of the Laplacian matrix

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{2.43}$$

with $\boldsymbol{U}$ holding the eigenvectors of the Laplacian on its columns

$$\boldsymbol{U} = \begin{bmatrix} | & | & & | \\ \boldsymbol{u}^{(0)} & \boldsymbol{u}^{(1)} & \dots & \boldsymbol{u}^{(N-1)} \\ | & | & & | \end{bmatrix} \tag{2.44}$$

We can define the graph Fourier matrix $\boldsymbol{F}$ as in (2.45).

$$\boldsymbol{F} = \boldsymbol{U}^T \tag{2.45}$$

We can define the Laplacian based graph Fourier transform of an input signal $\boldsymbol{s}$ as in (2.46).

$$\hat{\boldsymbol{s}} = \boldsymbol{F}\boldsymbol{s} = \boldsymbol{U}^T\boldsymbol{s} \tag{2.46}$$

The inverse of the graph Fourier transform (2.45) will be (2.47). Note that we use the transpose operation instead of Hermitian operation when we are dealing with the Laplacian. We were using the Hermitian operation when we had the eigendecomposition of the adjacency matrix $\boldsymbol{A}$. As we stated before, the Laplacian matrix is symmetric semi-definite and all its eigenvalues are real and non-negative and therefore it is safe and practical to use the transpose operation instead of the Hermitian operation.

$$\boldsymbol{F}^{-1} = \boldsymbol{U} \tag{2.47}$$

The inverse Fourier transform to recover the original signal $\boldsymbol{s}$ would be straightforward as shown in (2.48).

$$\boldsymbol{s} = \boldsymbol{F}^{-1}\hat{\boldsymbol{s}} = \boldsymbol{U}\hat{\boldsymbol{s}} \tag{2.48}$$

To have an intuition about the Laplacian based graph frequency analysis, we created a 16 by 16 grid graph with the connections of eight nearest neighbors as shown in Figure 2.3. It can be seen that each vertex is connected to their 8 neighbors with smallest Euclidean distances, which is known as k-nn graph with k=8. There are 256 vertices in that grid graph, and we present the eigenvalues of that graph as a stem plot in Figure 2.4, where the horizontal axis has the indexes of 256 eigenvector/eigenvalue pairs and the vertical axis shows the eigenvalues. As we stated that Laplacian matrix as a frequency matrix gives a better intuition to compare with the conventional signal processing (SP), the first eigenvalue is zero in the plot, however the distance between two eigenvalues cannot be interpreted as in conventional SP.

To have an understanding and intuition about how the Laplacian based graph frequencies show spatial variations, we showed the graph eigenvectors on the graph vertices in Figures 2.5, 2.6, 2.7 and 2.8. Figure 2.5, shows the eigenvector 1 (u_1) as an example with its values. Each value of the eigenvector is represented on the respective graph node with the same index. For simplicity, in Figures 2.6, 2.7, 2.8 we only showed the plots without the values to illustrate how the spatial patterns change with increasing spatial frequencies in a regular grid graph. Figure 2.6 presents the first 32 eigenvectors of the grid graph, where the first eigenvector has constant value and the variation increases as the graph frequencies increases. Figure 2.7 presents the middle 32 eigenvectors out of

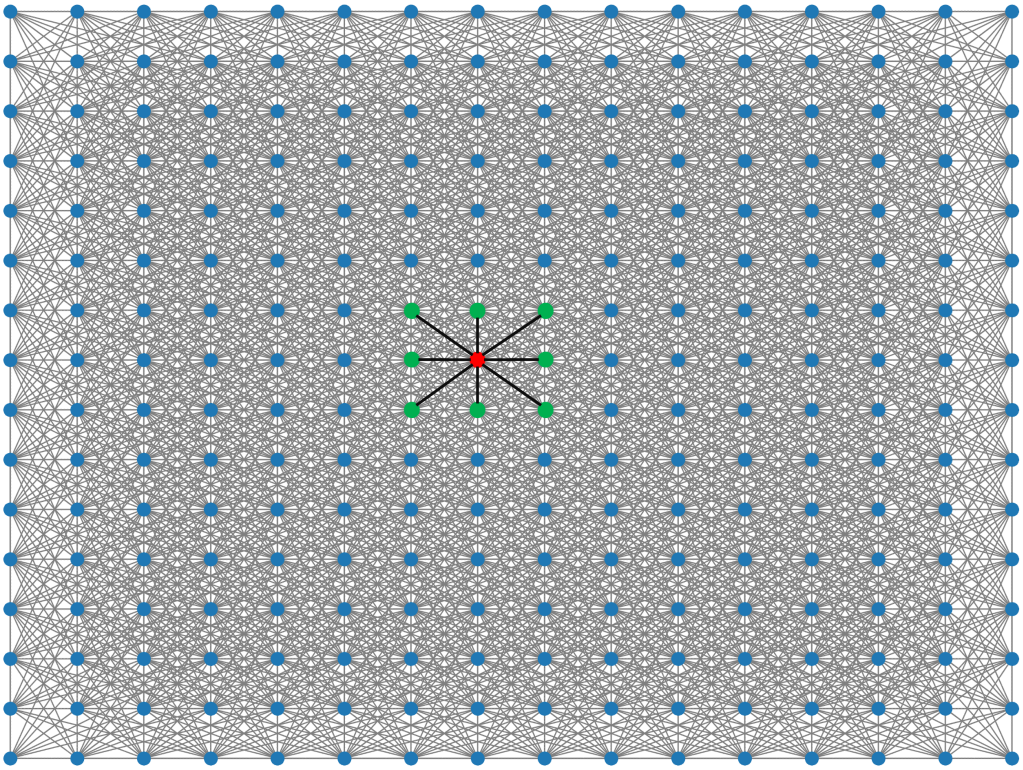


Figure 2.3: A 16 by 16 grid k-nn graph with k=8. There are 256 vertices and each vertex is connected to 8 of their neighbors with smallest Euclidean distances. The red point shows a sample vertex with its 8 nearest neighbors (figure created by the author).

the 256 eigenvectors, and they have more variance compared to first 32 eigenvectors that we presented in Figure 2.6. Figure 2.8 shows the last 32 eigenvectors of the grid graph. Although we stated that the spatial variance increases with the eigenvectors with higher eigenvalues, last 32 eigenvectors do not necessarily have greater variance than the middle 32 eigenvectors that we have in Figure 2.7.

Throughout this work, we will use Laplacian based graph frequency definition to assess the spatial variations of the graphs. We presented the illustrations of eigenvectors to have an intuition about how the graph frequencies affect the spatial variations between adjacent vertices. Although we will not depend on Euclidean distances for the remaining part of this work, it is important to have a visual representation to capture the details of graph frequencies and how it affects the proximity of different vertices. When the connections do not depend on Euclidean distances, it is not going to visualize the variations between adjacent vertices because different vertices in different parts of the brain can have stronger connections than adjacent vertices. This is because we do not limit the connections based on their Euclidean distances.

Through the illustration we shared, we showed that low spatial frequencies are associated with low spatial variances. The variation between adjacent vertices increases with graph frequencies however we cannot guarantee highest graph frequencies will represent

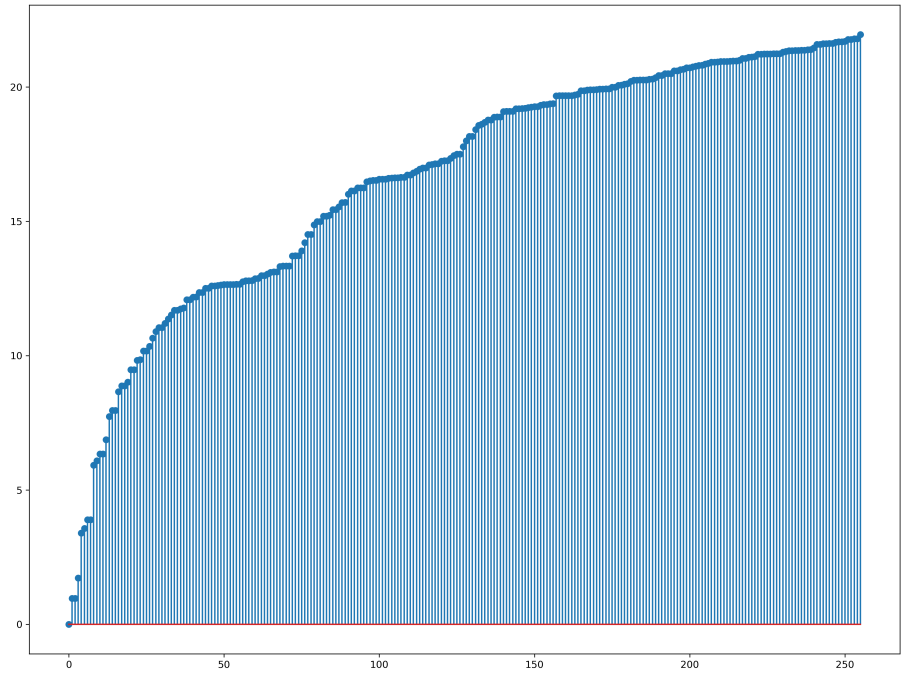


Figure 2.4: Eigenvalues of the 16 by 16 grid k-nn graph with k=8, which is presented in Figure 2.3. Horizontal axis has the indexes of 256 eigenvector/eigenvalue pairs and the vertical axis shows the eigenvalues. We can have an analogy with conventional signal processing as the first eigenvalue is 0 and the graph frequencies are ordered, however, the ratio or difference between eigenvalues does not represent the change of variation between different graph frequencies (figure created by the author).

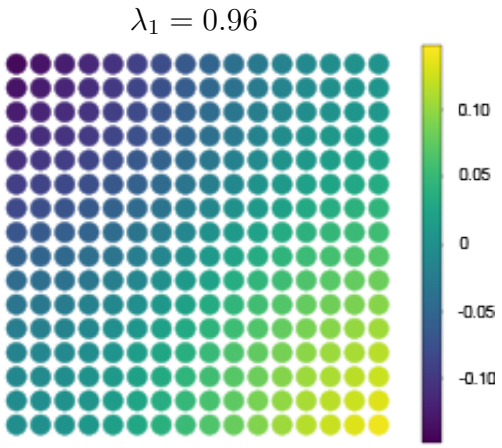


Figure 2.5: Eigenvector 1 (u_1) for the k-nn grid graph shown in Figure 2.3.

higher variation between adjacent vertices compared to middle graph frequencies. Although we try to carry our intuition from conventional signal processing, we do not have the same ratio of variance between different graph frequencies unlike signal processing. The difference of eigenvalues does not represent the difference on spatial variances.

In Chapter 3, we build upon those details to present a novel method of dynamic

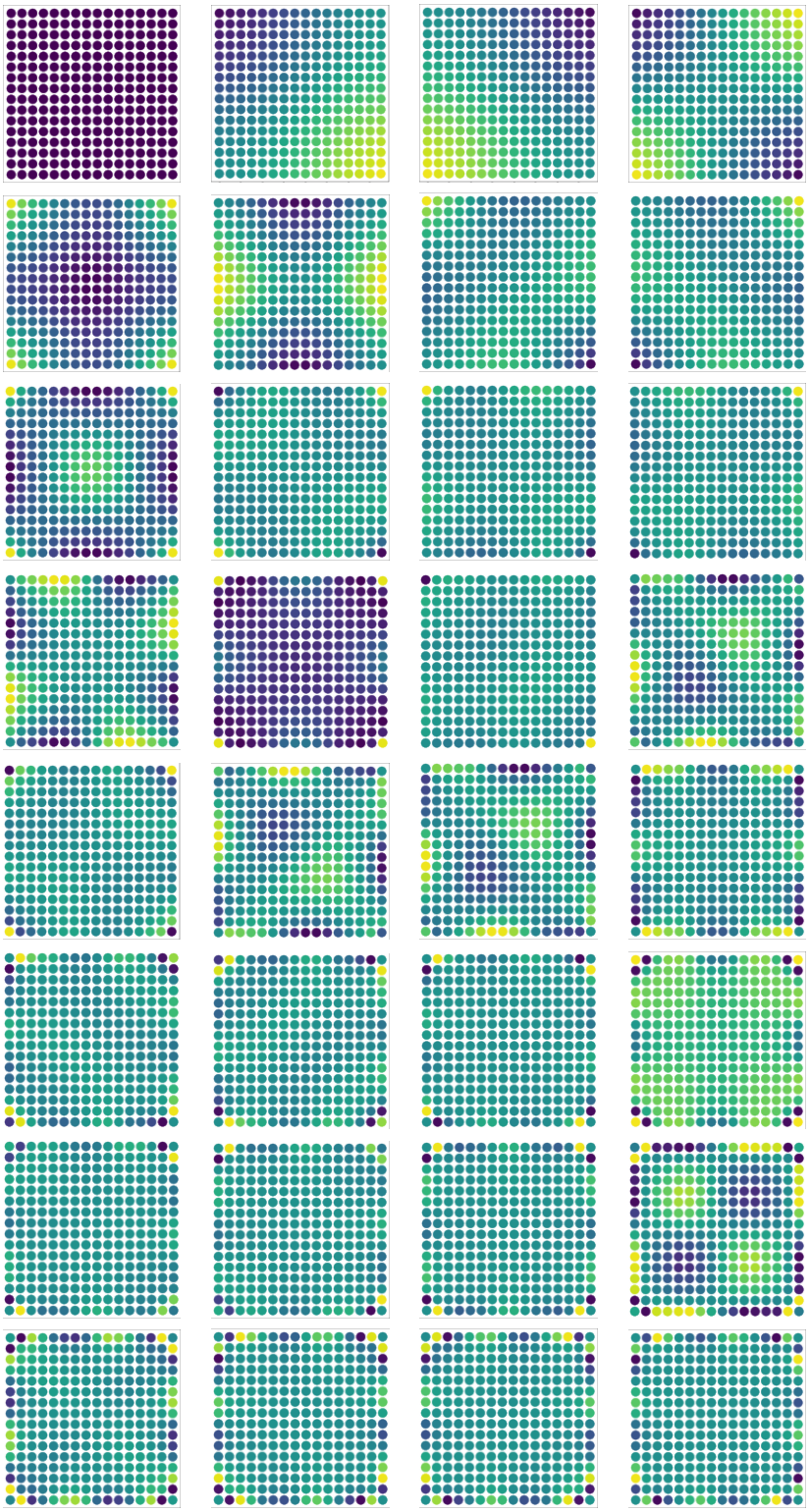


Figure 2.6: First 32 eigenvectors (0 to 31) of the sample grid graph. The first eigenvector has constant value for all the vertices and the variation between vertices increases with the graph frequencies, i.e., difference between neighboring vertices increases. For the lower graph frequencies, the neighboring vertices have very similar values, and for the greater eigenvalues, the difference between the neighboring channels increases (figure created by the author).

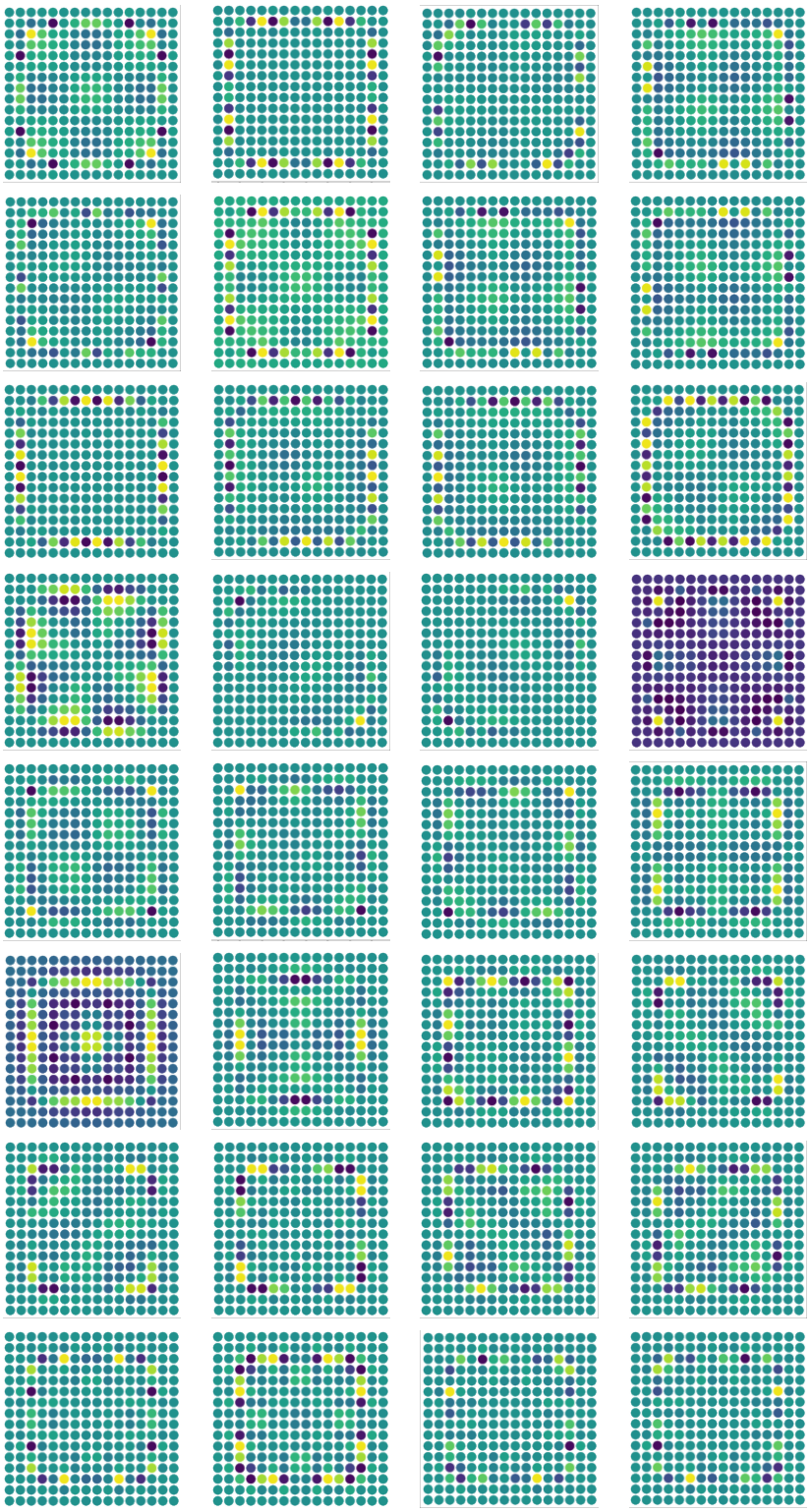


Figure 2.7: Middle 32 eigenvectors (112 to 143) of the sample grid graph. Compared to Figure 2.6, the variation between the neighboring vertices is greater (figure created by the author).

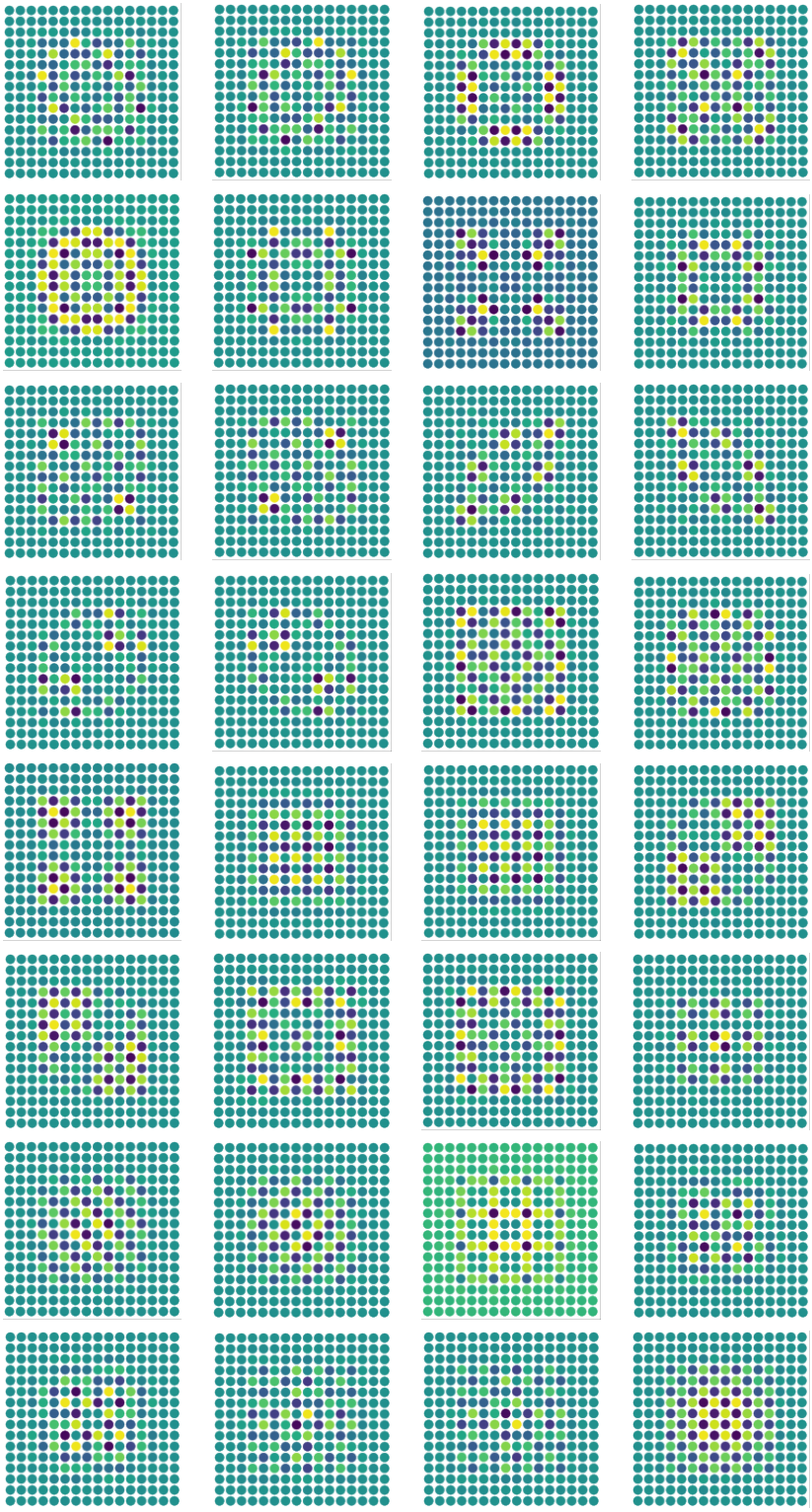


Figure 2.8: Last 32 eigenvectors (224 to 255) of the sample grid graph. Although we stated that the variation between neighboring vertices increases with graph frequencies, it can be seen that the variation of the last 32 eigenvectors and the middle 32 eigenvectors in Figure 2.7 is not much. In this particular case, it looks like this is also reflected in their eigenvalues which we presented in Figure 2.4 as the difference between eigenvalues of the middle 32 eigenvalues and the last 32 eigenvalues is not as great as the difference between first 32 eigenvalues and the middle 32 eigenvalues, however, this is not always the case and the ratio between different eigenvalues does not necessarily reflect the difference in variation as one might expect by their intuition from conventional signal processing (figure created by the author).

graph representations of EEG signals and how to incorporate spatial frequencies with the temporal frequencies to capture the spatiotemporal variations of graph signals.

2.2 Electroencephalography

There are invasive and non-invasive methods to collect brain data (Figure 2.9, and Figure 2.10). Different methods offer distinct advantages and disadvantages in terms of cost, risks and mobility. Electroencephalography (EEG) is a non-invasive method to capture the postsynaptic potentials through electrodes placed on top of the scalp (Figure 2.12). It has been in use for more than a century, and its versatility, mobility and accessibility compared to other methods make it highly popular for a large spectrum of applications [55, 56].

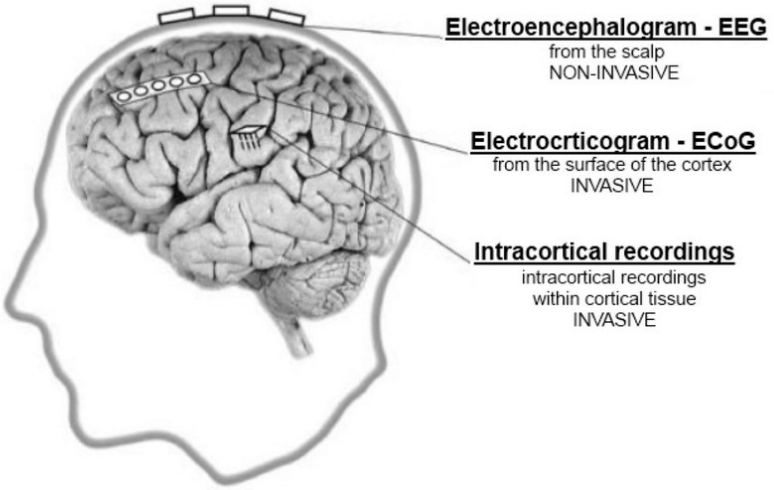


Figure 2.9: Different methods to record electrical activity of brain [57].

Within the brain, electrical signals propagate through neurons as transient changes in *membrane potential*, which is the difference in electrical potential across the neuronal membrane [59]. In the resting state, neurons have a higher negative charge relative to the outside of the cell. In the event of neurons being excited, a transient change in membrane potential occurs, and this causes nerve impulses to propagate through their axons [59] as seen in Figure 2.11.

Independent of the strength of the stimuli, individual nerve impulses have a uniform shape and amplitude. However, stronger stimuli result in a higher frequency of those impulses [59].

To carry information through longer distances, axons utilize an elementary unit of nerve impulses called *action potentials*. The intensity of the propagating signal is represented by the frequency of action potentials. Although the frequency of action potentials

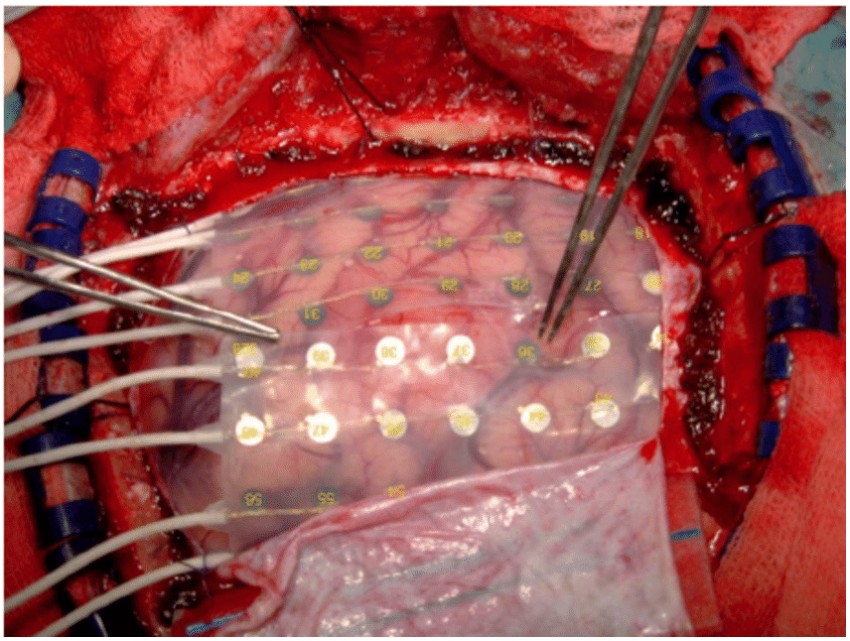


Figure 2.10: A grid of electrodes placed on brain cortex for electrocorticography [58].

is the main method to represent the signal intensity, the timing of action potentials may also represent significant information [59].

Graded potentials are membrane potentials that change in magnitude continuously. *Synaptic potentials* and *receptor potentials* are two types of graded potentials. Synaptic potentials are caused by the release of neurotransmitters from presynaptic neighbors and occur at postsynaptic sites. Receptor potentials are caused by sensory stimuli at peripheral endings of sensory neurons [59]. Contrary to action potentials, the amplitudes of graded potentials can change based on the strength of input stimuli and sensitivity of sensory neurons to these stimuli [59].

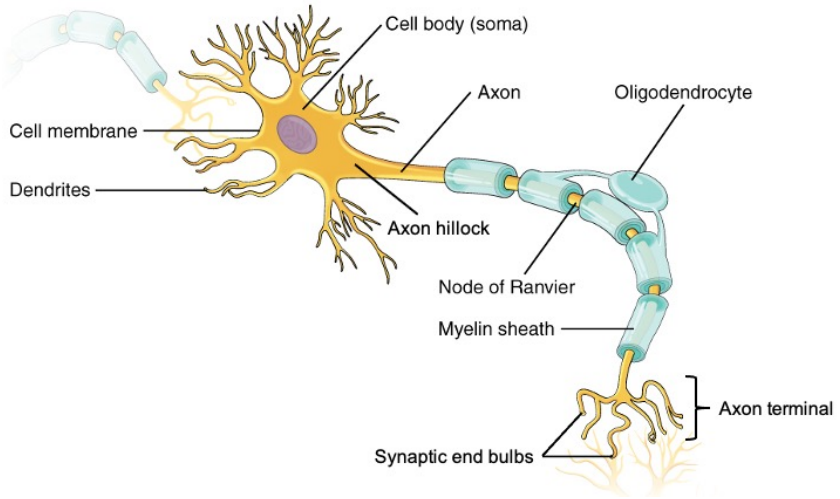


Figure 2.11: Parts of a neuron from central nervous system [60].

EEG enables the real-time monitoring of changes in electrical potentials. However, it does not measure action potentials, which are current flows from the soma through the axon, changing its quintessential resting potential value from -70 mV to -55 mV [55]. Action potentials are the greatest potential changes happening in neurons, but their extracellular amplitudes are so small that a large number of simultaneous action potentials in neurons is needed to create enough potential change to be detected with EEG. They happen very fast (approximately 1 ms) and the sum of action potentials cannot create enough potential, which makes it not possible to capture them through EEG [17]. Unlike action potentials, which occur rapidly and result in a time window too brief to produce sufficient potential change for detection, postsynaptic potentials, with their time span of up to 10ms, create sufficient potential change to be captured through EEG [17].

Figure 2.12: An experimental setup for EEG acquisition and brain activity recording [61].

2.2.1 EEG Feature Extraction Methods

There are various ways to calculate hand-crafted features of EEG signals. Throughout this thesis, we do not rely on hand-crafted features, and instead use spatial and temporal frequencies to represent the EEG signals. Although one can argue that calculating frequencies is also a way of calculating hand-crafted features, what we mean when we refer to hand-crafted features is mainly statistical features of signals, i.e., mean, variance. In Chapter 3 and subsequent chapter, we use the dynamic graph representations of the multi-channel EEG signals, which incorporates the spatial relationships and temporal variations. This means, we do not limit the information with the hand-crafted features, but we provide richer representations and details about the nature of the signals that are not available on the original recording. For example, when we feed the multi-channel

EEG signal directly into a neural network for a classification task, neural network has no information about the spatial relationships between different sensors which represent the pairwise relationships between different regions of the brain (Figures 2.13, 2.14). On the other hand, using hand-crafted statistical features is very limiting and the information we feed into neural network would be mainly lost. Nevertheless, we share some very common statistical feature extraction methods below for the sake of completeness.

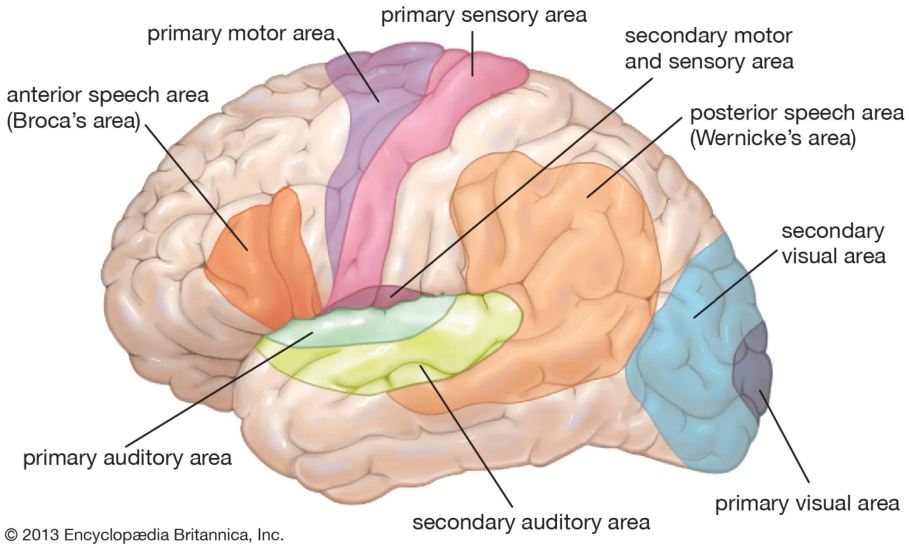


Figure 2.13: Broca's and Wernicke's areas [62].

2.2.2 Statistical Features

A common feature extraction method for EEG signals is calculating the statistical features of the EEG signals. Some examples of these statistical features are mean, variance, skewness, kurtosis, zero-crossings, zero-crossing rate, absolute area under signal, peak to peak value, amplitude spectral density, power spectral density and power of each frequency band.

Although we share the details of those features, in this work we avoid using hand-crafted features. Using hand-crafted features is not efficient. A significant amount of information is lost when features are used in place of the original signals. Machine-learning-based methods are more efficient in learning these features than manually hand-crafting them. Another issue about using the hand-crafted features is doing this per-channel basis. When we calculate those features, we only calculate them for each channel separately and this totally ignores the network relationships between different channels and therefore ignores the pairwise relationships between different regions of the brain. Most of the work in literature select some of the channels for classification tasks and later concatenate the features from those channels to get a new representation. This totally ignores the spatial variations of the EEG signals. Another issue is that this

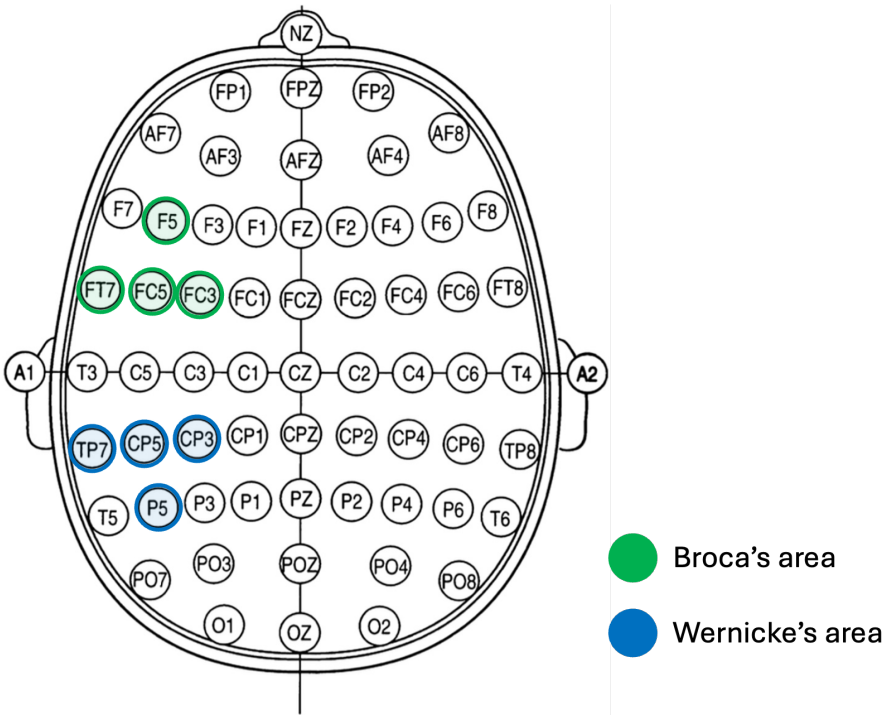


Figure 2.14: Sensors on Broca's and Wernicke's areas marked on a figure adapted from [45] for the international 10-10 electrode system. Sensors F5, FT7, FC5, FC3 are placed on top of Broca's area and marked with green circles. Sensors TP7, CP5, CP3, P5 are placed on top of Wernicke's area and marked with blue circles (figure modified by the author).

approach requires manual selection of channels. Although we can use all the channels, a common method in literature is to use hand-crafted statistical features for manually selected channels and selecting different channels for different subjects. In a real-life scenario, we cannot make per-subject adjustments. If using EEG data for biometric identification is considered, such as collecting data during visa applications for later use in border control, the approach should not require per-subject adjustments. It would not be feasible to select EEG channels for each subject separately. For an application to work in a real-life scenario, there should not be any per-subject adjustments.

For all these reasons, we do not use any statistical features and we think relying on those features or making any kind of per-subject adjustments is not a good strategy.

We briefly introduce some of those statistical features with their formulas below. Generally, those features are calculated for windows of a sequence, which means getting sequences of windows with those features. At the end of the calculations, the new representation becomes sequences of those features for each channel instead of the original sequences of each channel. This shortens the sequence length too. For example, if we have 64 channels with the length of eight seconds of recordings and we use a window size of two seconds with 50% overlap, we would have a sequence of seven values. If we had a sampling frequency of 1000Hz, that means instead of a sequence of 8000 values, we

now have only seven timesteps with each timestep having different number of features. In terms of tensor dimensions, we can represent the original signal as (channels, signals) where the original signal would be a tensor with the size of (64, 8000) and if we use 11 statistical features with two seconds of windows and 50% overlap as we mentioned above, the new representation would be (channels, time-steps, features) which would be a tensor with the size of (64, 7, 11). Most of the time, those features with channels dimensions are concatenated to have a representation of (time-steps, features), which in this case would be a tensor with the size of (7, 704), and the training data would be generally in the form of (samples, time-steps, features) or (batch-size, time-steps, features) when we feed the input into a neural network as batches of data. These hand-crafted features and concatenating the channel dimension with the features dimension to get all the features for a window totally ignores the spatial relationships between different channels. That is the main reason we avoid using this method. Although we explained the reasons we avoid using this method, we share their meanings and formulas below for rigor.

Mean A very basic statistical feature is the mean of a sequence, which shows the average value of the sequence. As we stated above, this is generally used for sequences of windows alongside other statistical features.

Variance Another very basic statistical feature is the *variance* which is the average of squared distance of each element to the mean value.

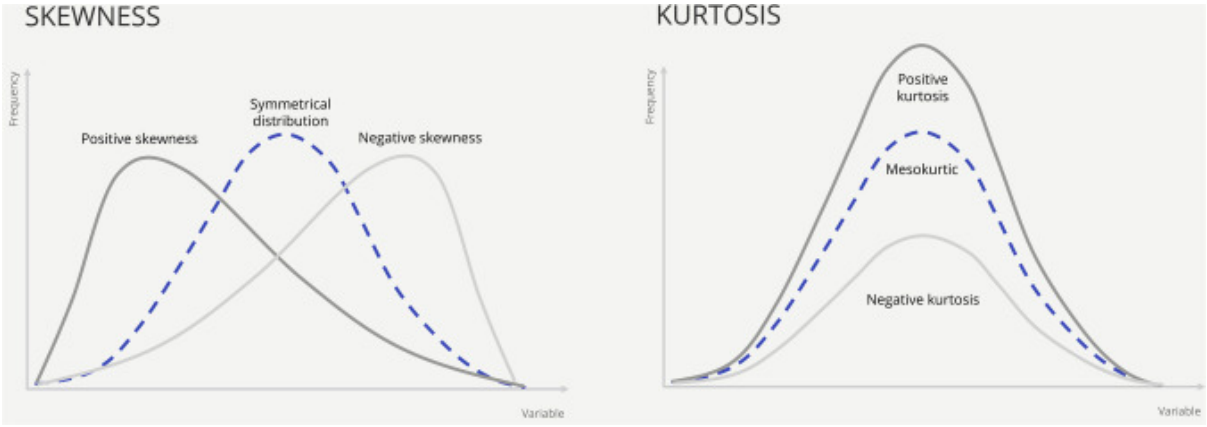


Figure 2.15: Skewness and kurtosis [63].

Skewness gives a measure to quantify the asymmetry of the probability distribution around the mean value. In the left-hand side of Figure 2.15, compared to symmetrical distribution curve, the positive skewness has a mean value that is greater than the peak of the curve. On the other hand, for the case of negative skewness, the mean value is less than the peak value.

Kurtosis There are variations of kurtosis, and we will refer to *excess kurtosis* as *kurtosis* because that is the common version used in the literature as a feature to use with EEG signals. Kurtosis provides a measure to quantify how the outliers in the statistical

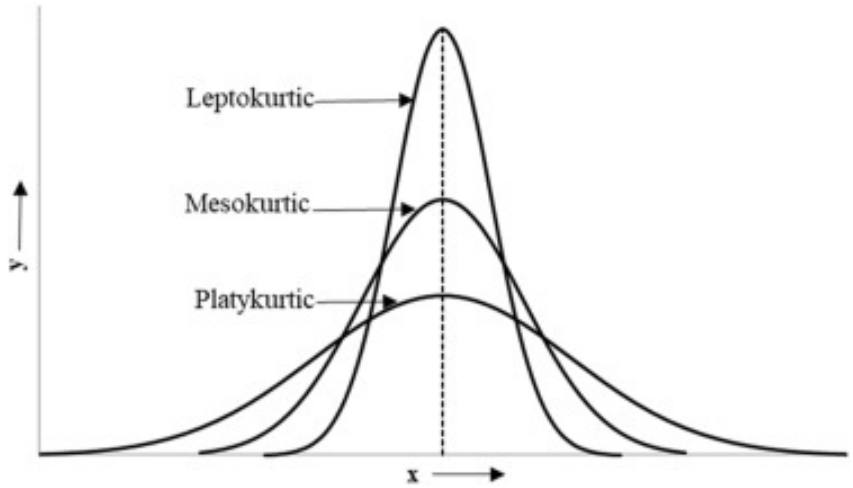


Figure 2.16: Kurtosis [64].

distribution is spread. In Figure 2.16, we can see three cases as *leptokurtic*, *mesokurtic* and *platykurtic*. The mesokurtic curve corresponds to *normal distribution*. Leptokurtic curve has heavier tails and more outliers and represented with a positive kurtosis value as shown in the right-hand side of Figure 2.15, while the platykurtic curve has lighter tails which corresponds to a negative kurtosis.

Zero-crossing Zero-crossing gives an idea about the oscillation around the x axis, which corresponds to how many times the sign of the sequence changes. An important thing to mention is that depending on the normalization strategy, this feature can vary too much. If the mean of the sequence is far away from zero, when we normalize the sequence, the sequence will be shifted and we can get a completely different value instead of the original one. For example, if we have oscillations around zero and the total number of zero-crossings is 100 while the mean value is -2, when we subtract the mean from the original sequence the oscillation will be shifted by 2 and will be around $y = 2$ line. Depending on the sequence, this might change the number of zero-crossings from 100 to 0, or it might stay the same. So, when using number of zero-crossings as a feature for a sequence, it is important to consider how this might be affected by the normalization method.

Zero-crossing rate Zero-crossing rate is the averaged version of the zero-crossings, where N is the number of samples in the sequence. We divide by $2N$ as each sample is used two times in the calculation.

Absolute area under signal Another common feature used as a statistical feature for EEG signals is the absolute area under signal using Simpson's rule.

Peak to peak Peak to peak value just measures the difference between the maximum and minimum values of a sequence.

Amplitude spectral density Amplitude spectral density provides a measure for the

amplitude of a signal for different frequencies.

Power spectral density Power spectral density provides a measure for signal's power over frequency.

Power of each frequency band Power of each frequency band gives power for a range of frequencies like *alpha* band or *beta* band signals. We summarized the statistical features for EEG signals in the Table 2.1.

Table 2.1: Commonly used statistical features for EEG signals [18, 65].

Statistical features for EEG signals	
Feature name	**formula**
Mean	$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$
Variance	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2$
Skewness	$S = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^3}{(\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu)^2)^{3/2}}$
Kurtosis	$K = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^4}{(\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2)^2} - 3$
Zero-crossing	$zc = \sum_{i=-\infty}^{\infty} \lvert sign(x_i) - sign(x_{i-1})\rvert$
Zero-crossing rate	$zcr = \frac{1}{2N}\sum_{i=-\infty}^{\infty} \lvert sign(x_i) - sign(x_{i-1})\rvert$
Absolute area under signal	$simps = \int_a^b \lvert f(x)\rvert dx$
Peak to peak	$peak2peak = max(x) - min(x)$
Amplitude spectral density	$\hat{X}(w) = \frac{1}{\sqrt{T}}\int_0^T x(t)exp^{-iwt}dt$
Power spectral density	$S_{xx}(w) = \lim_{T\to\infty}\left(E\lvert\hat{X}(w)\rvert^2\right)$
Power of each frequency band	$P = \frac{1}{\pi}\int_{w1}^{w2} S_{xx}(w)dw$

2.2.3 Independent Component Analysis

Independent component analysis (ICA) is a common method to analyze the EEG signals. It does not reduce the dimensionality, but separates the multivariate signal into independent components. As an example for illustration, we used the EEG data of the first subject from Kara One database [66] and applied the ICA to get different components from the signals. The original recording has 62 channels, and therefore we had 62 independent components. We used the [67] for calculating and plotting the components.

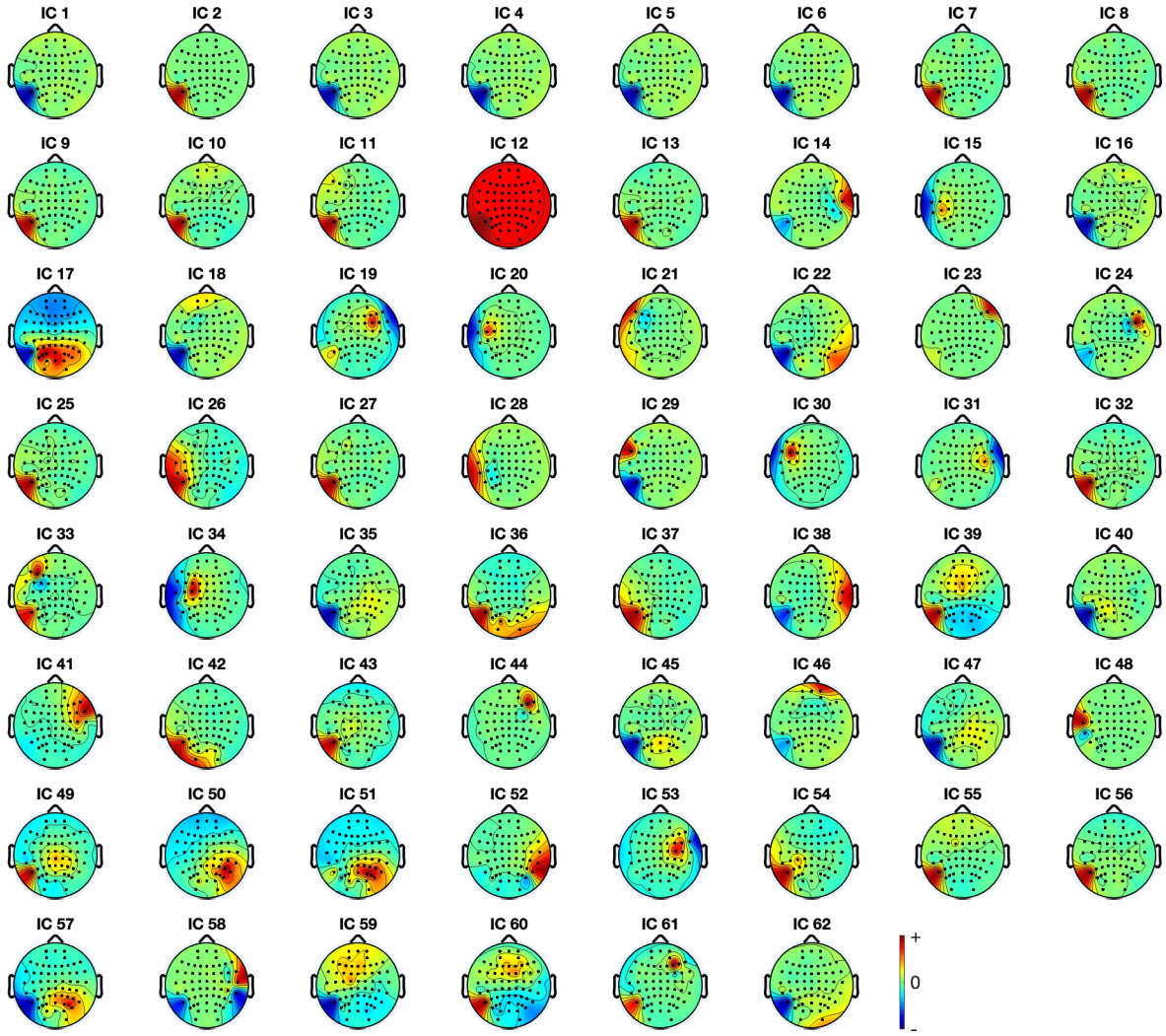


Figure 2.17: 2D illustration of the independent components for the subject 1 of the Kara One database [66] (figure created by the author using the EEGLAB package for MATLAB [67]).

In Figure 2.17, we shared the two-dimensional version of the independent components to show how different activity around different regions of the brain changes for different components.

We also calculated the component spectra for different states of the EEG recordings, namely *clearing*, *stimulus*, *thinking*, *speaking* states. We presented the activity for different states in different frequencies in Figures 2.18, 2.19, 2.20 and 2.21.

Although ICA is a popular method for analyzing the EEG signals and can give insights, the paper [66] is itself proof that ICA alone does not provide a significant contribution as the authors had only 18% accuracy for a binary classification task where they only used EEG data without using any audio or video data.

In Chapter 4, we used ICA as one of the baseline methods to be used alone without any graph operators to see how much of the classification accuracy and the performance

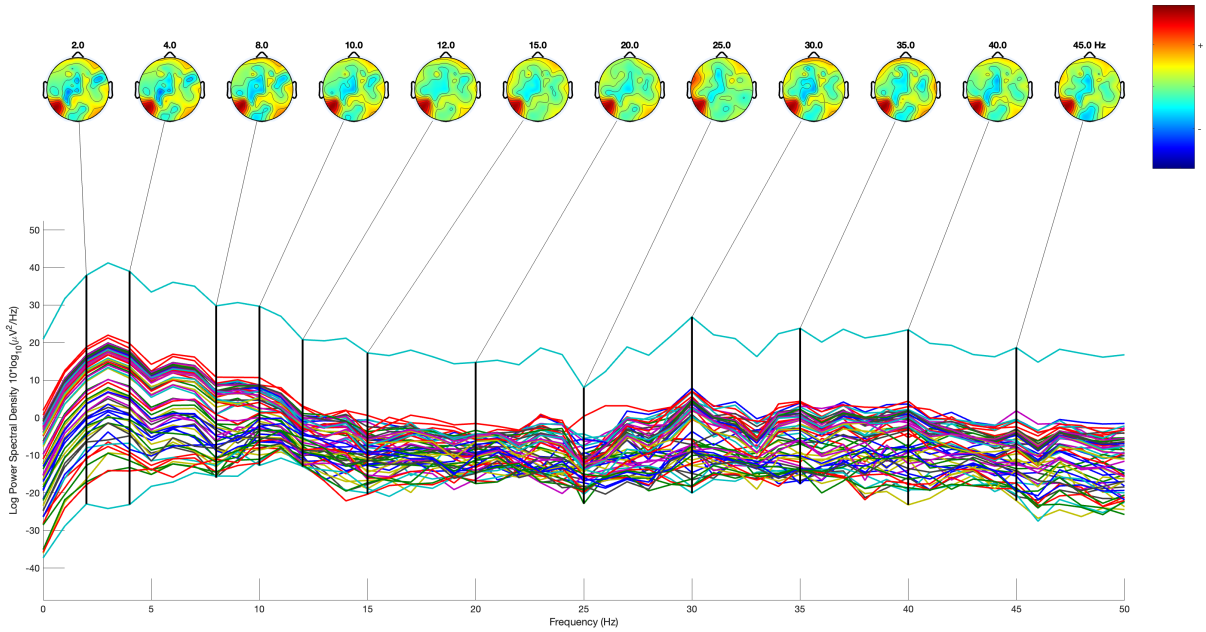


Figure 2.18: Component spectra during clearing state of the subject 1 of the Kara One database [66] (figure created by the author using the EEGLAB package for MATLAB [67]).

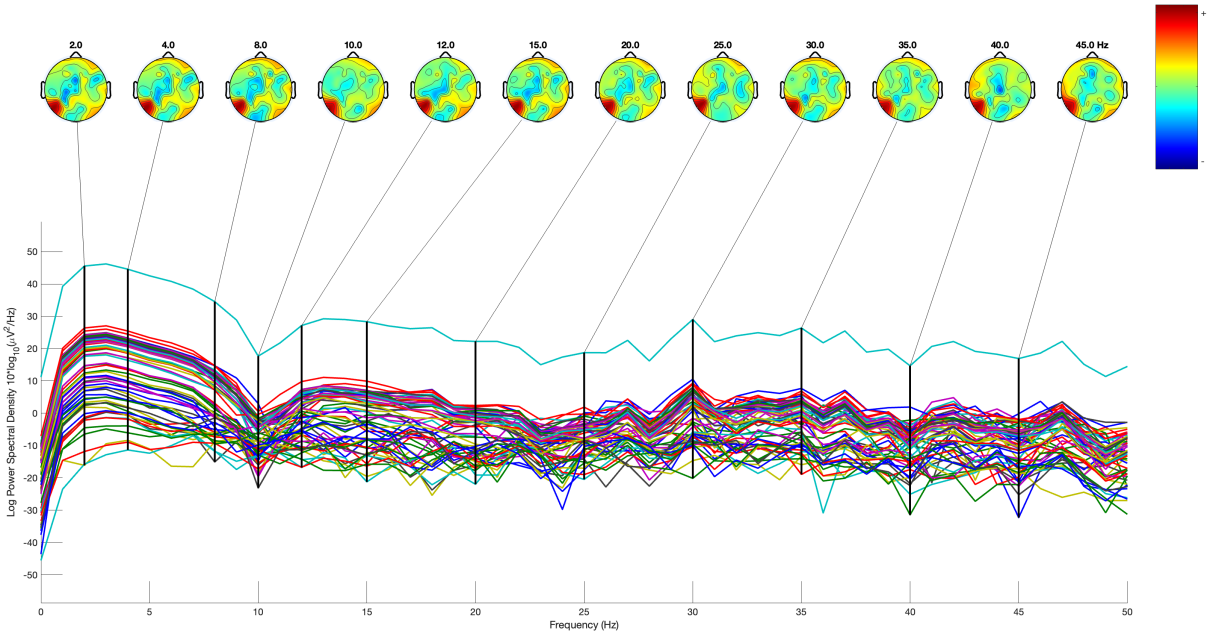


Figure 2.19: Component spectra during stimulus state of the subject 1 of the Kara One database [66] (figure created by the author using the EEGLAB package for MATLAB [67]).

come from the dynamic graph representations of the EEG signals.

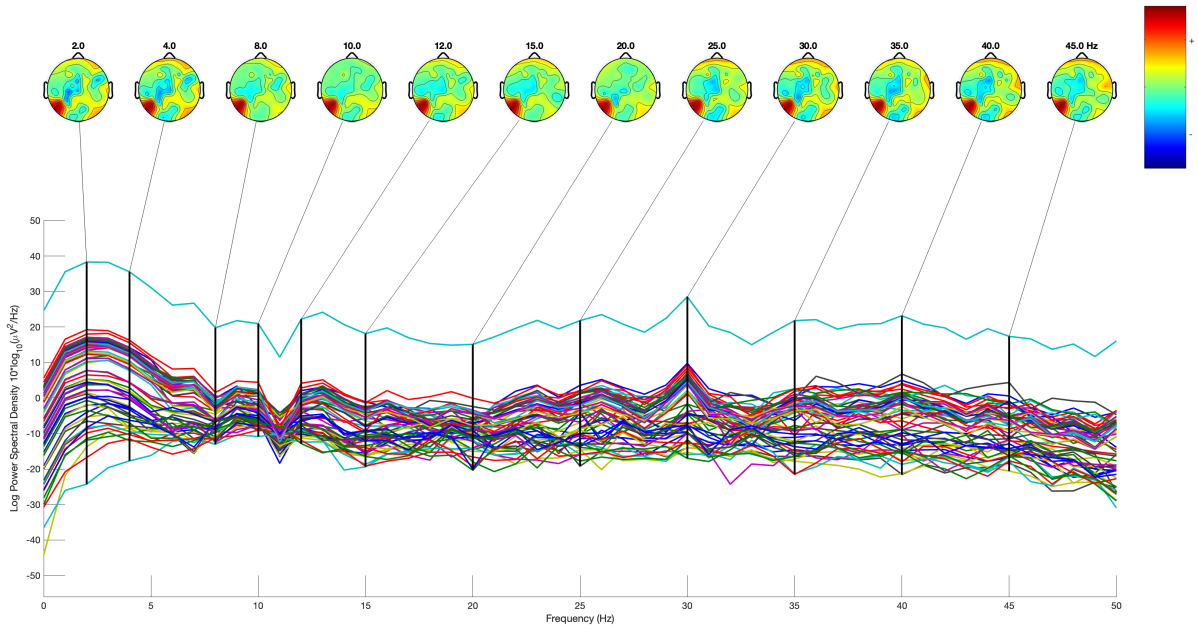


Figure 2.20: Component spectra during thinking state of the subject 1 of the Kara One database [66] (figure created by the author using the EEGLAB package for MATLAB [67]).

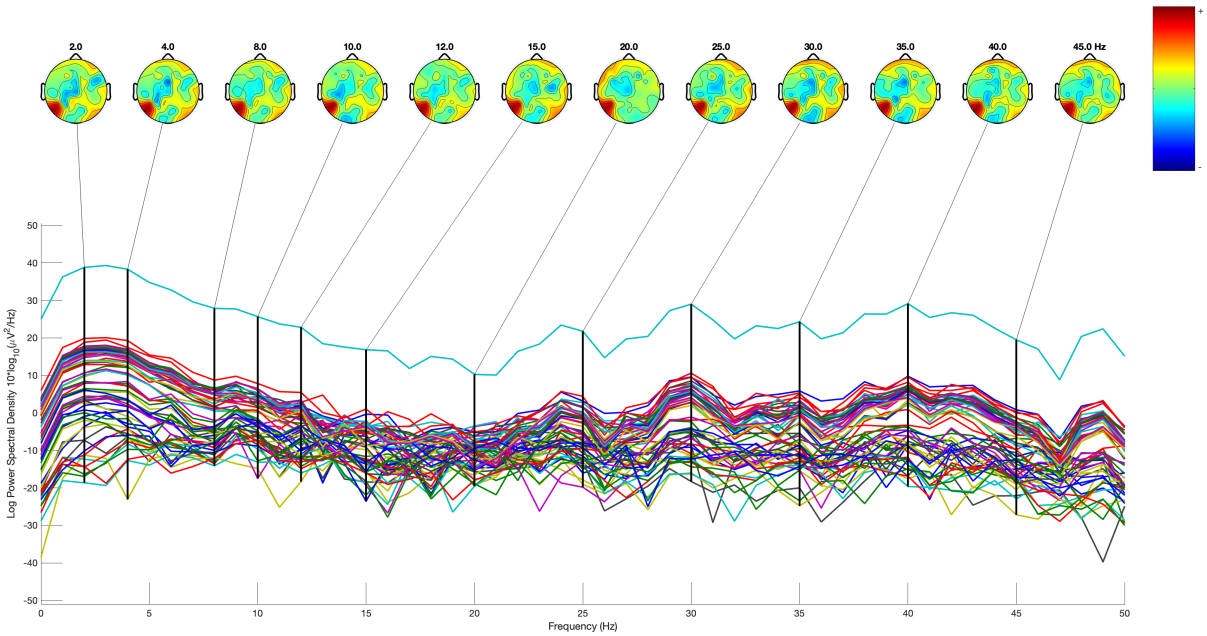


Figure 2.21: Component spectra during speaking state of the subject 1 of the Kara One database [66] (figure created by the author using the EEGLAB package for MATLAB [67]).

2.3 Machine Learning and Deep Learning

Deep learning [68] is a subfield of machine learning which is itself a subfield of artificial intelligence [69]. Artificial intelligence can be described as the automation of tasks that

typically require human intelligence. Although machine learning is an important part of AI, it was not mentioned in the literature until the 1980s. The definition of AI is very broad, and it includes symbolic AI which does not include machine learning but instead hard coded procedures [69]. Machine learning, as a subfield of AI, differs from symbolic AI. Contrary to symbolic AI, which relies on hardcoded rules to produce results, machine learning is the process of taking the data as input and getting the rules as output by inferring rules from data.

Machine learning is a broad subject by itself too and includes different methods to learn from data. For simple tasks, some basic machine learning algorithms like SVMs are effective, however, as the data and tasks get more complicated, there is a need for more comprehensive techniques to learn from data [69].

Another important part of this work is using deep learning for learning network structures and applying deep learning techniques for classification task regarding brain activities. Compared to imposing assumed network structures based on the activity (visual, motor, etc.), it is a better strategy to learn the network structure using deep learning techniques as it does not induce any bias based on activity type. Deep learning also allows us to analyze the long-term dependencies on sequences of changing graph activities and therefore change of graph frequencies. It is important to capture the change of activity levels for different regions of interest and different network activities as it allows us to match those activities with different tasks performed by different subjects.

With the development and accessibility of the Internet, the amount of collected data increased. As the hardware and graphical processing units got better over time it allowed the execution of much more complicated computational tasks which were not possible in the past due to the computational cost.

Increased amount of data with higher computational power made using more complicated machine learning algorithms on big data possible. Deep learning is a subfield of machine learning to learn more complex patterns through data with more complex structures. The term "deep" does not signify a "deeper" understanding, but instead refers to the presence of consecutive layers of perceptrons acting on data as cascaded filters to process the data and extract better representations from the data [68].

Deep learning made it possible to solve problems such as image classification and other computer vision tasks. As its consecutive layers start with simpler features and increasingly more complicated features, it provides a better representation of data for tasks like classification, detection and prediction [68].

Although EEG signals provide data to interpret the processes taking place in the brain, it is difficult to detect the patterns without using deep learning [70–72].

Detecting long term dependencies in sequence data is difficult as when calculating the derivatives for the backpropagation, the gradients become either too small or too large as a result of calculating through chain rule which are referred as vanishing gradient or

exploding gradient problems. LSTM networks [73] provide a framework to detect long and short-term dependencies in sequence data effectively.

Since EEG data exhibit temporal dynamics of the long-term and short-term variations of signals, LSTM networks provide a framework to detect long and short-term dependencies through sequences of EEG signals.

2.3.1 Convolutional Neural Networks

Convolutional neural networks (CNN) address a shortcoming of the *dense layers* or *fully connected layers* [74]. Dense layers map each input directly without considering the locality of the features [68]. For example, if we think about a picture, for a dense layer a specific pixel is a feature and when an object changes its place on a picture, it becomes a completely different sets of inputs for the dense layer. So, if we get the mirror image of the original image, the sets of features will be completely different as all the pixels will change their places, unless the image is symmetric. On the other hand, convolutional neural networks use kernels for local processing, i.e., they have local filters moving around the input image and processes each pixel with their adjacent pixels based on the kernel size [68]. This makes convolutional neural networks more robust against the movement of the features in the input, i.e., being able to recognize the image even when the object is at a different part of the input image. As an example, if we want the neural network to recognize traffic lights in an input video stream, we would like the neural network to be able to recognize the traffic lights whether they are at the top or the middle section of the video stream. For these reasons, convolutional neural networks became very successful in *computer vision* applications.

2.3.2 Long Short Term Memory Networks

Long short term memory (LSTM) networks are a type of *recurrent neural networks* [73, 75]. They are used on sequential data, which is a set of data where the order matters. For example, natural language is sequential data, as the order of the words is important and the change of the order of the words or sentences can change the meaning of the words or sentences completely. When dealing with the sequential data, learning both short and long term dependencies is really important.

However, to process, classify or generate sequential data has an important challenge. When learning long term dependencies, during the backpropagation process where the derivatives taken for each parameter, calculating a chain of derivatives creates a *vanishing gradient* or *exploding gradient* problem. While calculating the chain of derivatives, the gradient can become very small and goes to zero, or it can be too big and goes to infinity [73]. For these reasons, learning long term dependencies is a challenge. Long

short term memory networks address this problem and they are robust in learning long term dependencies. For these reasons, they are used in sequence learning [75].

2.3.3 Attention Mechanism and Transformers

In a machine learning task, some of the input features are more important than the others [76]. For example, in an image, some sections of the image can be more important than the other sections. If we think about the way we read a paper, we might go to the conclusion section first, or we can go through images and results tables before reading the paper in depth. In a similar way, the attention mechanism helps the network focus on the most relevant features instead of treating all input features equally [76].

The other advantage of attention is that it can help with understanding the context [76–78]. For example, when translating a sentence to another language, understanding in what context a word is used is important. A *station* can be a radio station or a train station. To understand in which way the word is used, self-attention uses the surrounding words in the sentence to calculate a new representation of the original vector that represents the word. For this reason, when working on sequential data, especially in *natural language processing*, it became very popular to use attention mechanism at the end of LSTM layers or other RNNs [76].

Before the publication of [77], using RNNs with the attention mechanism defined in [79] and [76] was the popular method, however, after the publication of [77], transformers became the main architecture to be used for dealing with sequential data. Inspired by *The Beatles* song *all you need is love*, the paper was named *attention is all you need* [80]. As the name suggests, transformers use only attention mechanism without any convolutions or recurrent neural networks. The transformer architecture described in [77] has received over 140,000 citations, according to Google Scholar. They form the foundational architecture for modern natural language processing applications, including widely used chatbots.

Although transformers are very efficient in natural language processing, in other sequential data types like EEG time-sequences they are not that popular. In natural language processing, words are represented as vectors in a pre-defined dictionary [81], and the available text data is very big because of the Internet. A transformer architecture with multiple head attention [77] can learn distinctive features from data, and that kind of a large data is available through the Internet. However, EEG datasets are very small [18, 82]. This can cause the neural network to overfit, not learn generalized features and perform poorly on the test data.

2.4 Machine Learning Algorithm Validation Schemes for EEG Classification Tasks

To validate a method, the testing set should be sufficiently large to represent the statistical distribution of the samples of the training data. When we deal with a small dataset, it is difficult to ensure that the test set is representative enough to have unbiased test results. For this reason, we need to validate the results with a more complicated validation method instead of just randomly choosing some of the samples from the dataset for testing. This is particularly important when the dataset is very heterogenous and some of the samples are much different than the others. In that case, difficulty of prediction varies through different samples and depending on which samples we choose, the test results may vary too much. As it is not straightforward to label which samples are easy or hard to predict, we need to use a validation method to ensure that the test set is statistically diverse enough and does not include a small subset of the dataset where very similar samples are present in the training set. Two of the very common validation methods are namely leave one out validation and k-fold cross validation.

2.4.1 Leave One Out Validation Scheme

Leave one out is a very simple validation method that, as the name suggests, excludes only one sample from the dataset and uses the rest of the dataset for training [66, 83]. That one excluded sample is used for testing. After that, the same procedure is repeated for each sample in the dataset. Consequently, the results will be binary as the test result will be either 0% or 100%. Afterwards, the same procedure is repeated for each sample and the individual test results are averaged.

A huge downside of this is the fact that the number of runs for the experiments increases to the number of samples in the dataset. However, when the sample size is very limited, that might not be a concern and depending on the computational resources and the use case, it can be a viable method to validate the results without having too much bias.

2.4.2 K-fold Cross Validation Scheme

K-fold cross validation is another common validation scheme where the dataset is divided into K partitions and each partition is used as a test set while the remaining data is used for the training [83]. That means we need to run the experiments K times using different training and testing partitions. This can help with decreasing the bias, as the experiments are repeated K times with K different subsets of the dataset instead of one time.

To further make sure the testing set is diverse enough to have unbiased results, we

can shuffle the dataset and repeat the K-fold cross validation multiple times. This also changes the combination of the samples of each subset that we use for testing and gives more reliable results. However, this increases the number of times we need to run the experiments and increases the computational cost. Depending on the use case, if the dataset has a limited number of samples and we have enough computation resources, it is a viable option to get more reliable testing results.

2.4.3 Nested Cross Validation and Validation with a Limited Sample Size

Collecting EEG data is costly, takes time and requires subjects to sit in a controlled environment. It is not as easy as collecting images from the internet to construct an image dataset. For this reason, EEG datasets are very limited compared to datasets on other fields. This raises concerns about the validity and potential bias of the results.

Nested cross-validation [83, 84] addresses this issue by excluding the test data completely from any kind of calculations, i.e., feature selection and model development, before the classification phase as shown in Figure 2.22. It suggests that the cross validation methods should not only be applied for the machine learning classification but also for all the calculations, feature selections, model development, parameter tuning and other calculations that come before the machine learning classification takes place. That means all the calculations before the classification procedure is also repeated K times to make sure that the test samples are never used at any stage prior to the machine learning classification phase. According to [83], this gives almost unbiased results on EEG datasets with a limited number of samples [85].

2.5 Related Work

2.5.1 Graph Signal Processing for EEG Applications

Graph modeling of brain signals acquired through EEG has been used in several past works [25–30]. All such works construct a graph by determining the graph weights on the basis of the spatial proximity between the respective vertices, where vertices that are farther apart are linked by edges with smaller weights. However, since the topology of the graph in the modeling of EEG signals matches the location (on the head of a subject) of the EEG electrodes, which remains fixed, the resultant graphs are static and cannot model different behaviors in situations where signals from different vertices exhibit varying correlation over time.

Using a static graph is a common method to model and represent EEG signals. A popular way to assign weights for the graph adjacency matrix is the thresholded Gaussian

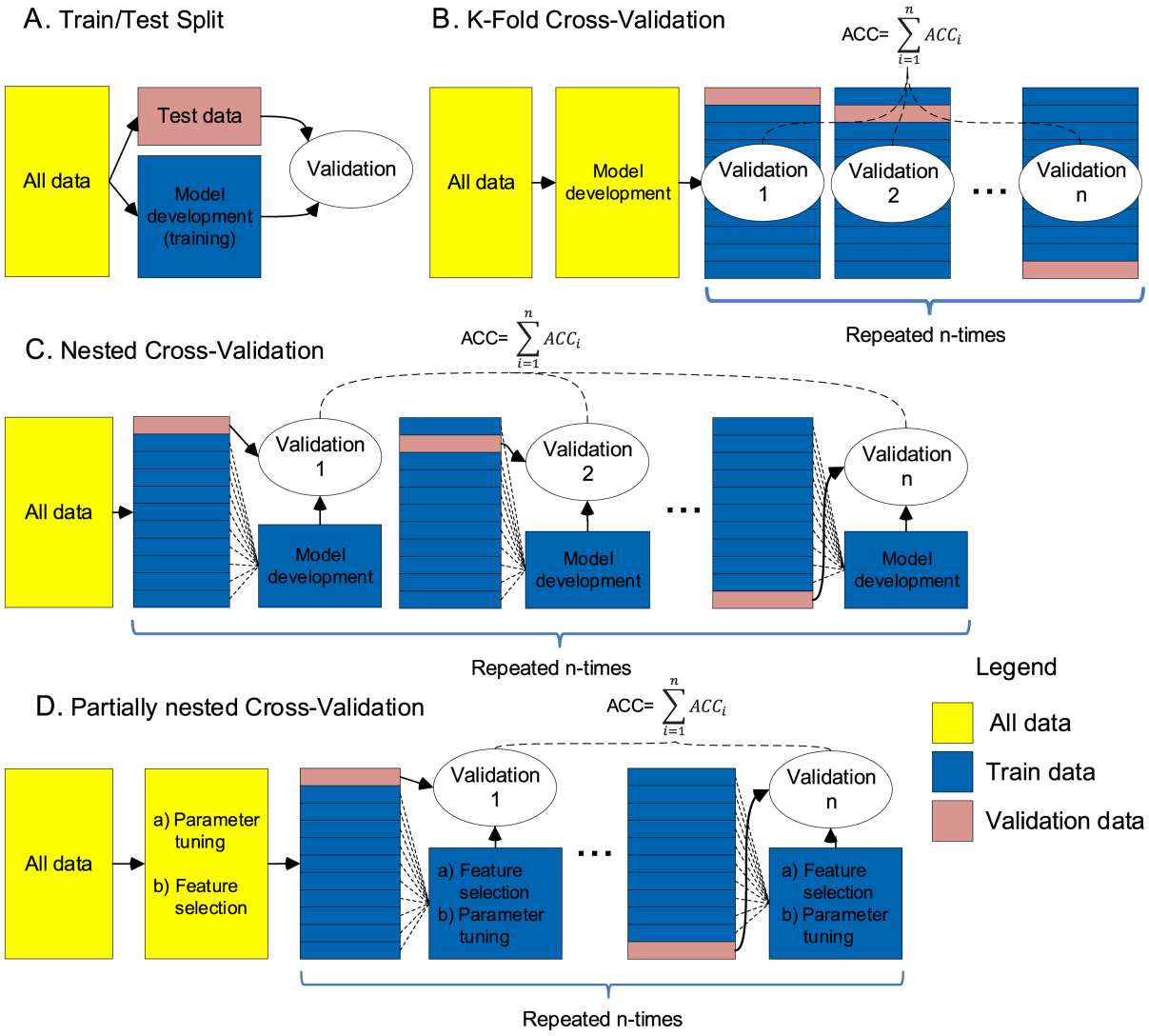


Figure 2.22: Comparison of different validation methods [85].

kernel weighting function [4]. For example, [25] used this method to analyze EEG data from patients with disorders of consciousness (DOC) using a GSP-based approach. This method creates a static graph for all subjects and ignores any possible variation.

Similarly, [29] creates a static graph based on Cartesian 3D coordinates of the EEG sensors and assigns the weights based on the distances between sensor pairs.

As a further example, authors in [28] use a graph-based spatio-temporal attention neural network for emotion recognition using EEG data. The connections for the graph are based on the inverse of the Euclidean distances between sensor pairs. This approach leads to the same static graph for each subject and cannot model dynamic processes. The neural network architecture proposed in [28] introduces a novel approach where spatial features and temporal features are processed separately. Specifically, temporal features are processed using LSTM [73, 75], while spatial features are processed using the multi-column convolutional neural network (MCNN) [86]. Although they employ both spatial

and temporal features, they use them as separate inputs to two different neural network parts and later concatenate the features and feed them into dense layers. This does not provide a unified representation of spatial and temporal features. Instead, they are isolated and processed through two different neural network parts before being concatenated, which contrasts with our aim of creating a unified spatiotemporal representation of graph signals. This graph design overlooks the temporal evolution of spatial features.

Several studies have employed GSP for EEG-based classification tasks. For example, GSP has been applied to EEG data for emotion classification [87], for the detection of Attention Deficit Hyperactivity Disorder (ADHD) on children [27], and for the cross-subject mental task classification [88].

In [89], authors used a graph Fourier subspace for the classification of two-class motor imagery EEG data. The proposed method analyzes the network using a static graph and represents the graph signals employing a subspace of the Fourier matrix.

Authors in [90] have a GSP framework to classify temporal brain data. They apply an extension of the Fukunaga-Koontz transform and use these features to train a decision tree for the classification task on the autism spectrum disorder database ABIDE. They also use a static graph, with weights assigned as the inverse of the Euclidean distances, and apply it to resting-state functional magnetic resonance imaging data. They compared their method with Spatial Filtering Method (SFM) and claimed that they had around 4% increase in performance.

In contrast to EEG-focused studies, authors in [91] use GSP to analyze atrial fibrillation. They have a grid of electrodes on the surface of the heart and use 8-nearest-neighbors to create a graph based on Euclidean distances. They also normalize the distances so that the largest weight equals one. They use a joint graph and short-time Fourier transform to demonstrate how the frequency characteristics of the graph change based on the existence of atrial fibrillation and concluded that spatial variations decrease during atrial fibrillation as high temporal frequencies reduced. Although this work does not concern EEG-based data, it demonstrates the typical use case of the static graph structure explained throughout this chapter with a static k-nearest-neighbor graph with proximity-based weight assignments.

As an EEG-based classification task, authors in [65] use a 3-layer LSTM network with an attention mechanism to classify hand movements. They removed non-symmetrical channels and obtained the average of the signals of symmetric channels to decrease the number of features. They extracted statistical features such as skewness and kurtosis for binary classification using the LSTM network and the attention mechanism. They achieved 96% accuracy with their network and obtained better classification accuracies than other methods they compared with, which have 76% accuracy. However, the figure representing the database is not accurate and the number of epochs is reported as 10, making the results difficult to reproduce.

Although not directly relevant as a GSP application, a very interesting study in [92], which follows the paper [93], recreates seen images with generative adversarial networks [94] from EEG data. In [93], the authors use CNNs to learn a discriminative brain activity manifold, which is later used as a feature extraction method for capturing features related to human conception. In [92] they took it one step further to use those features to regenerate seen images from EEG data.

2.5.2 Imagined Speech

A particular EEG application, which has received relatively less attention, is the recognition of imagined speech, i.e., the interpretation of the specific words *"imagined"* by a subject [95–97]. This can have important applications, such as facilitating communication for coma patients and supporting other brain-computer interface systems. Although some methods focusing on this application have been proposed in the past [66, 98–102], these methods typically rely on signal analysis approaches that extract features from each EEG channel separately. Such analysis neglects the structure of the network of sensors as well as the spatial relationships between them.

Imagined speech recognition [103], which is the application we focus on in the present work, is a research area receiving increasing attention. As one of the first works, [66] used hand-crafted statistical features and classified imagined words and phonemes using a deep-belief network (DBN) and support vector machines (SVM). Classification was based on EEG data from subjects who imagine speaking those words without making any sound or movements. The work in [66] presented five different binary classification experiments, and reported 18.08% accuracy for the vowel-only vs consonant (C/V) experiment and 79.16% accuracy for the presence of a vowel. This approach has limited applicability in a real-life scenario, as it is not feasible to manually choose features for each subject/application. Authors in [102], used the same dataset as [66] and employed a multi-channel convolution neural network to classify the thought word based on their grammatical classes.

The specific use of EEG-based GSP for imagined speech has not been widely explored. The work in [104] uses a graph approach based on a static graph. The database in [98] was used for training and testing. The works in [29] and [105] use GSP for imagined speech classification. However, both use static graphs with adjacency matrices constructed based on spatial proximities between sensor pairs. Other works are those in [99–102], which do not use graph representations, but use the popular *"Kara One"* database [66].

Table 2.2 provides a structured comparison of these methods, including the datasets used, methodological approaches, and performance metrics.

Table 2.2: Comparison of directly related methods in literature. [66, 98–101, 103, 104] are directly used for benchmarking while others are related for graph construction as explained in Section 2.5.

Performance metrics for short vs long word experiment.				
Work	Task	Method	Dataset	Performance metrics
[98]	Imagined speech	Covariance matrix descriptors	[98]	Accuracy
[103]	Imagined speech	Standardization-refinement domain adaptation	[98]	Accuracy
[104]	Imagined speech	Imagined speech functional connectivity graph	[98]	Accuracy
[66]	Imagined speech	Deep-belief network, SVM	[66]	Accuracy
[99]	Imagined speech	Restricted Boltzmann machine	[66]	Accuracy
[100]	Imagined speech	CNN, LSTM	[66]	Accuracy
[101]	Imagined speech	Correntropy spectral density	[66]	Accuracy
[102]	Imagined speech	Multi-channel CNN	[66]	Accuracy
[28]	Emotion recognition	GSP and STANN	[106]	Accuracy
[27]	ADHD detection	GSP	[27]	Accuracy
[29]	Imagined speech	SP and GSP	[29]	Accuracy
[26]	Motor imagery	GSP, cross-frequency coupling	[107]	Accuracy
[88]	Mental task classification	GSP	[108]	Accuracy, precision, recall, F-score

2.5.3 EEG-based Biometric Identification

Although our main focus is imagined speech, imagined speech datasets are not widely employed in biometrics applications. For example, in recent EEG-based biometric studies, [109] and [110] use the DEAP dataset [106] for emotion recognition, while [111, 112] and [113] use the Physionet database [114]. As the Physionet database contains 109 subjects and has already been used in previous studies, it serves as a more suitable benchmark dataset for comparing our method with existing approaches in the literature.

For biometric identification tasks, [112,115–118] and [119] use their own datasets. To enable comparison with state-of-the-art methods, we used the Physionet dataset [114] alongside the Kara One [66] dataset even though Physionet is a motor imagery dataset.

[119] uses a cognitive event-related biometric recognition method for biometric identification. They use their own datasets, however, their recordings for different individuals

are customized, so that each subject can create distinct EEG signals through different tasks and this works as a "pass-thought". Although they have reached 100% accuracy, the dataset itself makes it easier to classify EEG data of different individuals.

Authors in [112] use functional connectivity metrics to identify different individuals. Their approach is based on graphs, however, they do not employ GSP. Instead, they calculate individual variabilities through their proposed connectivity metric.

The study in [113] uses a deep learning-based approach to classify the identities of individuals, while [111] uses ADEN, a feature selection method, on the Physionet dataset and applies linear discriminant analysis and support vector machines for classification.

These studies highlight current approaches in EEG-based biometric identification. However, none of these studies employ GSP or utilize imagined speech data for biometric identification.

2.6 Summary

As discussed throughout this chapter, GSP is not widely employed for the recognition of imagined speech. Some studies have applied GSP to imagined speech, but they rely on the proximities of sensor pairs to construct graphs. The graph structures employed in these studies are all static and based on Euclidean distances (proximities) between different signal channels. Furthermore, when GSP is employed, creating a unified framework that integrates both spatial and temporal features is another challenge. This is a significant area that warrants further investigation.

In addition to imagined speech classification, GSP is also not widely employed for EEG-based biometric identification. The existing approaches are based on relatively simple methods such as linear discriminant analysis. However, GSP methods are not leveraged for EEG-based identification. This presents another area that requires greater attention and investigation. In the following chapters, we address these issues by employing a dynamic-graph-based approach to better model the signals defined on a network of multi-channel EEG recordings.

Chapter 3

Dynamic Graph Representations of EEG Signals

3.1 Introduction

Analyzing brain signals and brain activity is a compelling research area. It gives insight into the cognitive activities, consciousness, and affective states. Among different methods to monitor brain activity, electroencephalography (EEG) is the most popular method to capture brain signals. It is a non-invasive method that captures postsynaptic potentials through electrodes placed on top of the scalp [16]. Invasive methods like electrocorticogram and intracranial recordings require medical operations to place sensors on the surface of the cortex or within the cortical tissue. This makes them impractical and costly. Other methods like fMRI require subjects to be inside expensive fMRI machines for long times, and data collection is not feasible when the subject is mobile. EEG has advantages over those methods as it is non-invasive, does not require expensive medical devices, and can be used for applications where the subject is mobile.

Processing and analyzing EEG data have a variety of different applications [20, 21]. Some of these applications include disease and neurological disorder classification, sleep analysis, brain-computer interfaces (BCI), and more generally, human-machine interfaces (HMI) [23]. Using EEG for cognitive activity classification is one of the research areas where EEG can be used to identify a subject's brain state. This can be used for BCI applications, create control applications to help locked-in patients, control applications for gaming, silent speech applications [19, 22] and many other applications where creating different EEG signals can be used as an input method to control a variety of applications [20]. Using EEG signals is convenient not only for patients with stroke or locked in patients, but also for non-disabled individuals whose hands are already occupied with another control task or who cannot verbally communicate.

Although there is considerable promise and substantial research on EEG applications,

they are not yet part of daily life, as many of them are not feasible in practical scenarios. EEG is practical, however, the trade-off is that the signal captured on the scalp is highly susceptible to noise and analyzing the multi-channel signals is not straightforward.

One of the challenges of processing and analyzing EEG signals is dealing with its multi-channel nature. The EEG data resides on a non-Euclidean domain and analyzing channels separately neglects the spatial features of the signals.

Graph Signal Processing (GSP) [4, 6, 9–11, 14, 120] is still an emerging area where it can be leveraged to gain insight into spatial features and relations of a network. However, when GSP is used, it can be used in spatial dimension only, which means it gives the spatial variations for a time instant and neglects the time-variations.

The application of GSP on brain activity data aims to better analyze and model brain waves and interconnectivities of different regions of the brain and give a more accurate model for the representation of signals that have been acquired from a network of sensors. Such models are particularly suited for brain signals, which tend to have specific spatial relationships between different regions of the brain. Therefore, instead of analyzing different regions of the brain individually and analyzing the signals in the traditional way, GSP provides a framework to analyze the network through graph signals and graph frequencies.

Most EEG classification tasks depend on statistical hand-crafted features and various deep learning techniques. Network structure is generally ignored in most of the works. Graph domain features are typically used in conjunction with time-frequency features, rather than being employed separately. It is not possible to inspect how much of the gain comes from graph features.

Two things are crucial to leverage GSP for EEG signal processing and analysis. First, we need to define a graph to model the network of the different regions of the brain. Second, we need a methodology to incorporate spatial variations between sensors and temporal variations through time. Defining the graph itself is the modeling part for any application where we want to analyze the signals on a network structure. However, most of the work in the literature overlooks that vital primary part where we define the graph to model and analyze the EEG signals. If we were dealing with a simple setup like temperatures of different cities, a simple static graph would be enough for the task, however, that is not a viable option to model the brain network.

The uniqueness and the dynamic nature of the graph representations of EEG signals are two important overlooked aspects. There is not one unique graph representation of EEG signals. As stated in Chapter 2, most of the studies construct graphs based on similar approaches. However, graph representation of a signal depends on choosing the graph. Different graphs create completely different models, different eigenvectors, and different graph frequencies. Therefore, the method of creating the graph itself is the part where we define the graph frequencies and what kind of signal variations on the graph we

are measuring. We cannot state that there is a unique subject independent graph EEG signal representation that depends on Euclidean distances of EEG sensors. That would require one static graph, which is based on 10-20 [44] or 10-10 [45, 121] sensor placement systems, and use it to model and analyze all the EEG data for all the applications and all the subjects around the world with all the data ever recorded (Figures 3.1, 3.2, 3.3). This causes all the applications to have almost the same static EEG graph based on 10-10 or 10-20 system. That would mean using the eigenvectors of the same graph to interpret EEG signals for all tasks and all subjects (Figure 3.4).

Although one can have variations of that graph representation using different strategies like using different numbers of neighbors with different k-NN graphs (Figure 3.3, 3.5, 3.6), that would still ignore a lot of dynamics of the non-Euclidean data, and how the network structure changes depending on the application. Using the same graph for different EEG applications like emotion recognition, biometric identification and authentication, motor imagery and imagined speech classification would ignore the dynamics of the brain and how different regions of the brain cooperate based on the task, and how the network structure changes depending on the task. In applications like emotion recognition, again, one static graph assumes that the network structure is the same when the subject has different affective states. It is neither realistic nor practical to assume that a subject has the same brain network structure when they are anxious, bored, angry or irritated. Moreover, two related regions do not necessarily have to be topographically adjacent to each other to be correlated. For these reasons, it is important to realize that the graph representation of EEG signals is not unique for all the applications, and choosing the graph is the modeling itself.

The second overlooked aspect of the graph representation is that the network structure and connectivities between nodes are task-dependent. We might have different strategies to model the network structure, however, it still will be one static graph. Choosing one graph to model the network throughout an activity like speaking or imagined speech classification, assumes that the network structure does not change during that activity. For example, when a subject speaks different words during a conversation, it is unrealistic to assume that the graph structure remains unchanged over time. For these reasons, instead of using one static graph, a dynamic graph representation of the brain signals is vital.

As we stated above, GSP only captures the spatial variations for a time instant. While we can use GSP to analyze the spatial frequencies, we still need to take the temporal variations into account. Although GSP is popular and many studies in the literature use graph features as separate features of a system to analyze EEG signals, they use spatial and temporal frequencies separately. When spatial features are fed into one neural network block and temporal features into another neural network block, the system actually does not analyze the signals as a whole, it is not a spatiotemporal graph representation

of EEG signals. This is a limitation in effectively leveraging GSP. For these reasons, developing a more robust system to extract spatiotemporal features for the analysis of EEG signals is of critical importance.

In this chapter, we propose a graph signal representation methodology for the recognition of imagined speech from EEG signals. Our approach is based on multiple graphs, which allow a set of EEG signals to be represented using different graph models. This approach is particularly efficient for representing signals that exhibit varying behavior which cannot be accurately modeled through a static graph model. The EEG signals represented using the proposed graph representation are classified using machine learning techniques. The resulting system is shown to have advantages in comparison to previous methods for imagined speech recognition.

Specifically, the contributions of the present chapter are:

- A systematic approach for the construction of multiple graphs that are jointly used for the modeling of multidimensional EEG signals. The proposed dynamic graph representation of EEG data takes into account the time-varying interdependencies between channels and constructs EEG signal representations that are appropriate for capturing the underlying brain functions.
- A methodology for the spatiotemporal representation of these graph signals and the formulation of compact, highly discriminative features suitable for subsequent classification.

The structure of the chapter is as follows:

- Section 3.2 presents the representation of EEG signals using graphs.
- Section 3.3 presents the dynamic graph representation of EEG signals.
- Section 3.4 presents the spatiotemporal representations of the graph signals to demonstrate the analysis of the time variations of dynamic graph frequencies.
- Finally, the conclusions are drawn in Section 3.5.

3.2 Graph Representation of EEG Signals

Let $\boldsymbol{S}$ be a matrix representing EEG data with N channels, where each row of $\boldsymbol{S}$ includes the signal captured by an EEG electrode.

$$\boldsymbol{S} = \begin{bmatrix} \text{——} & Channel\ 0 & \text{——} \\ \text{——} & Channel\ 1 & \text{——} \\ & \vdots & \\ \text{——} & Channel\ N-1 & \text{——} \end{bmatrix}$$

Although each channel exhibits significant temporal variations, analyzing each channel individually ignores the spatial relationships between different nodes. Therefore, a multi-channel EEG signal $\boldsymbol{s}$ can be represented as a sequence of graph signals for each time frame (vectors) that include the signal samples from all channels at time t. A time frame representing the signal at time t is defined as

$$\boldsymbol{s}^{(t)} = \begin{bmatrix} s_0^{(t)} & s_1^{(t)} & \dots & s_{N-1}^{(t)} \end{bmatrix}^T \in \mathbb{C}^N \tag{3.1}$$

where $t = 0, \dots \zeta - 1$ is the temporal index of the respective time frame and ζ is the total number of time frames. The sequence of signal vectors for all time indices can be represented as

$$\boldsymbol{S} = \begin{bmatrix} | & & | & & | \\ \boldsymbol{s}^{(0)} & \dots & \boldsymbol{s}^{(t)} & \dots & \boldsymbol{s}^{(\zeta-1)} \\ | & & | & & | \end{bmatrix} \tag{3.2}$$

For each time frame t, the respective *Graph Fourier Transform* (GFT) frequency vector is obtained as

$$\hat{\boldsymbol{s}}^{(t)} = \boldsymbol{F}\boldsymbol{s}^{(t)} \tag{3.3}$$

Therefore, the final GFT representation of the entire EEG data is a temporal sequence of consecutive graph frequency vectors, which can be compactly expressed as

$$\hat{\boldsymbol{S}} = \boldsymbol{F}\boldsymbol{S} = \begin{bmatrix} | & & | & & | \\ \hat{\boldsymbol{s}}^{(0)} & \dots & \hat{\boldsymbol{s}}^{(t)} & \dots & \hat{\boldsymbol{s}}^{(\zeta-1)} \\ | & & | & & | \end{bmatrix} \tag{3.4}$$

These details are explicitly shown to emphasize that the graph Fourier transform does not capture temporal variations. It is calculated for each individual time frame, applied on a snapshot of the data at a specific time instant. This would not be a problem if the data did not exhibit significant temporal variations. If we were measuring the temperatures from a server room and wanted to check for a variation between neighboring sensors, we wouldn't care much about the time variations, and a high graph frequency or a high Laplacian value for a node could mean a faulty sensor or an anomaly at that point [6]. However, for a dynamic system like the brain, temporal relationships are critically important as information flows through different regions. Graph frequencies at a single time instant do not convey sufficient information about the system. In contrast, observing how the graph frequencies evolve over time provides insight into the temporal dynamics of the spatial frequency components.

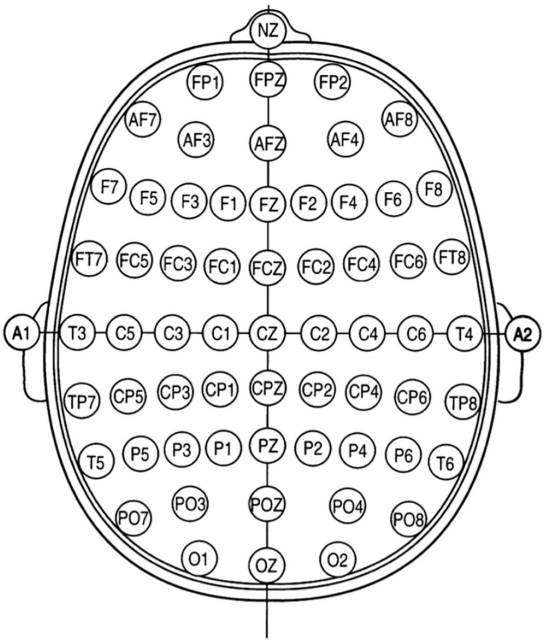


Figure 3.1: An illustration of the international 10-10 electrode system according to [45] and [121].

In the ensuing section, we describe our proposed method for dynamic graph representation and modeling of the brain signals. Such modeling and representation are suitable for imagined speech recognition.

3.3 Dynamic Graph Representation of EEG Signals

To define and analyze the graph $\boldsymbol{\mathcal{G}}$, the connections between graph nodes should be specified. Different connections between nodes and different weights for those connections result in completely different network structures. There are several methods to construct a graph and set the connections such as using a thresholded Gaussian kernel weighting function [4]. This can model a network where the neighboring nodes have high correlations, such as temperature in neighboring geographical locations. However, such modeling would not be suitable in cases where the connections and correlations between different nodes do not always depend on geographical distances between nodes. EEG data is an example of such a case, where the connections between different brain locations may not depend on the proximity between those locations. For this reason, modeling the network structure of the brain requires a different approach.

In addition to determining the connections and weights of the graph $\boldsymbol{\mathcal{G}}$, a further modeling challenge is that the graph structure is not static and depends on the undertaken brain activity. A static graph can model a network where the connections between the nodes do not change through time, such as routes between different airports in different

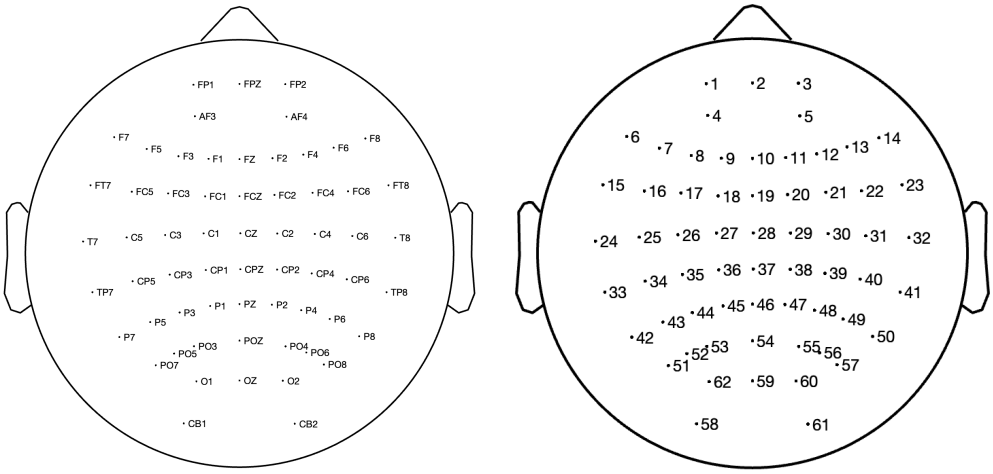
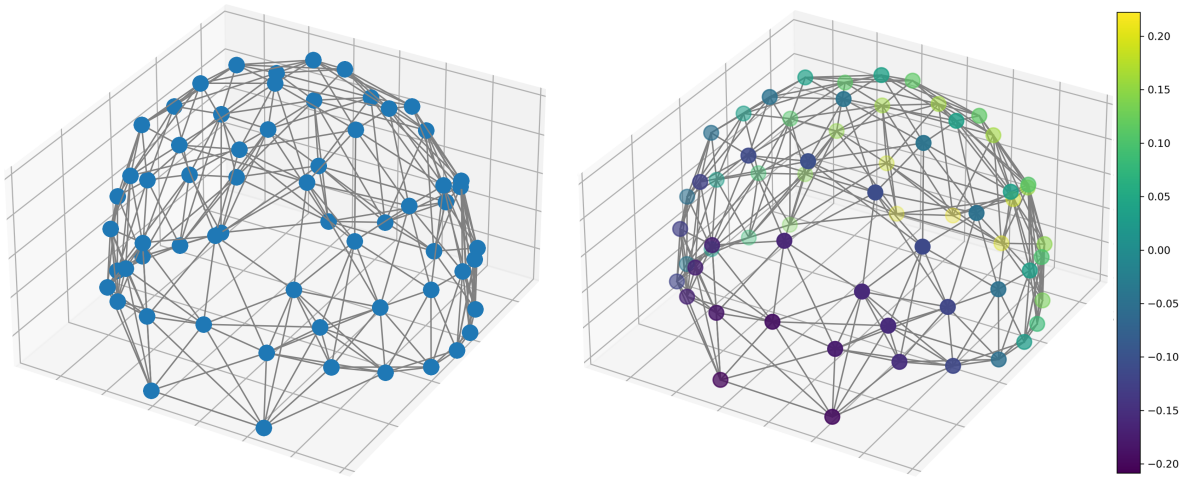


Figure 3.2: Sensor placement for Subject 1 of the Kara One database [66]. Left figure shows the channel names, and the right figure shows the channel numbers (figure created by the author using the EEGLAB package for MATLAB [67]).



(a) (b)

Figure 3.3: (a) A sample graph using the EEG sensor locations for the first subject of the database [66]. (b) eigenvector 1 u_1 for the same graph (figures created by the author using the PyGSP package for Python [122]).

cities. However, when a graph is used to model brain signals, one has to consider that brain activity depends on the brain task undertaken and, therefore, different network structures may need to be used for different underlying brain activities. For this reason, we do not rely on one static graph to model the brain. Instead, we use a dynamic graph representation to model the brain connections by creating multiple graphs to represent the network structure. It should be noted that our dynamic graph modeling, based on multiple graphs, is different from *"hypergraphs"* [5, 123], where an edge can connect multiple vertices at once instead of connecting two vertices, and therefore edges can have

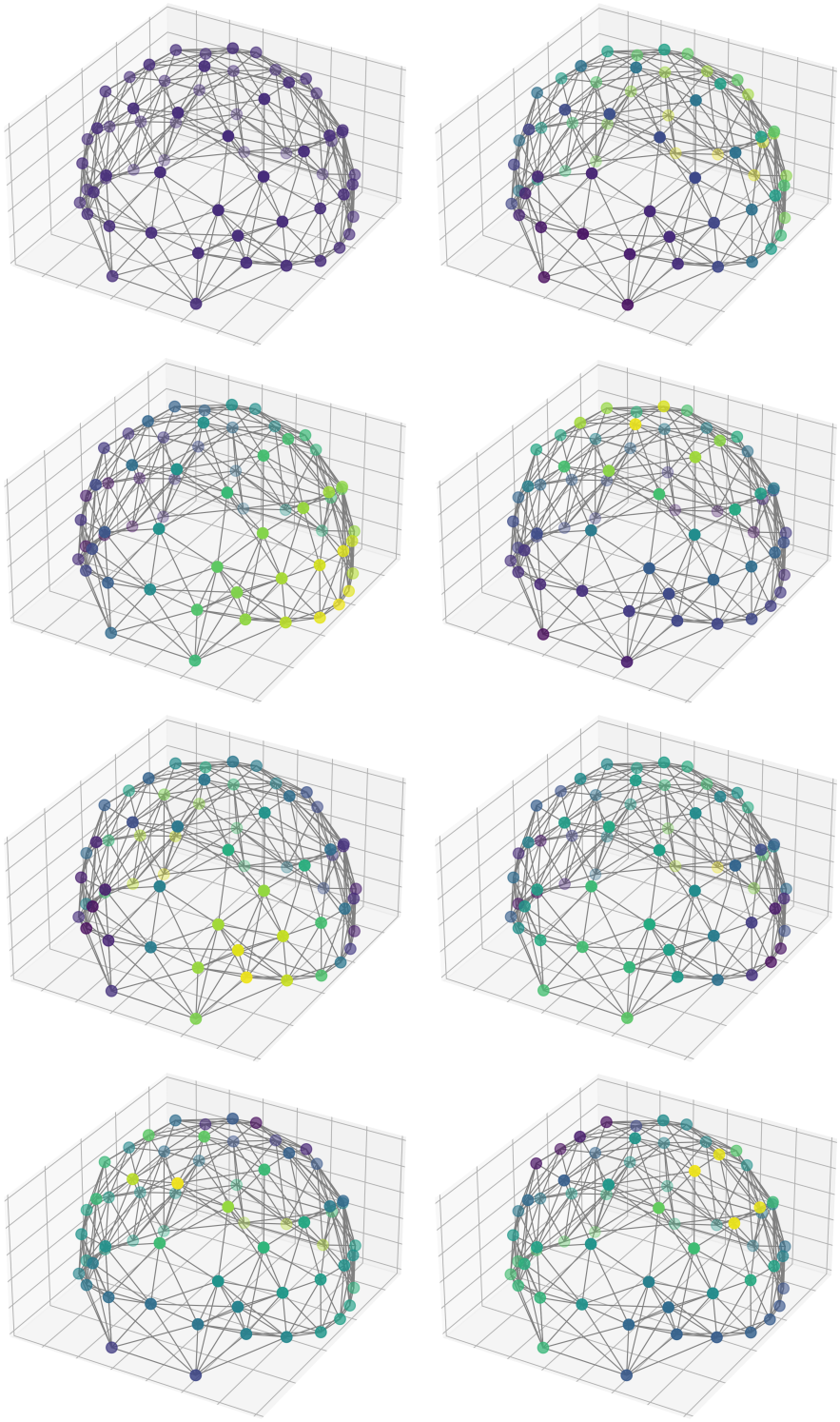
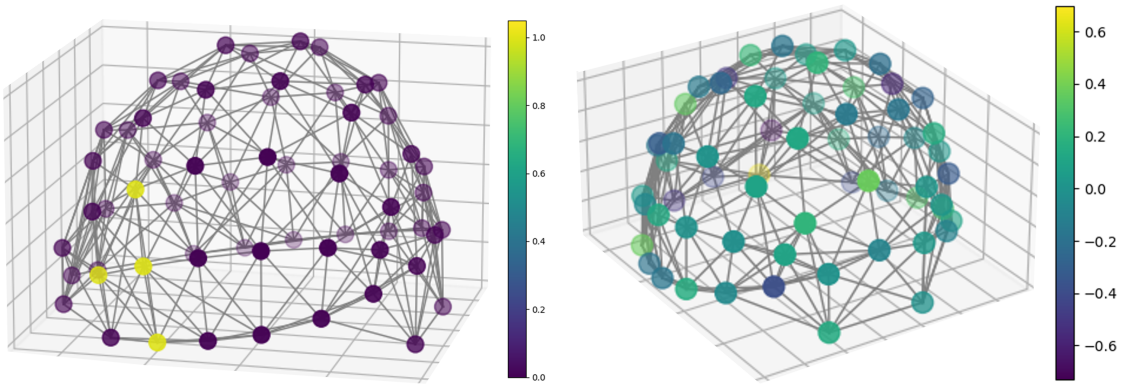
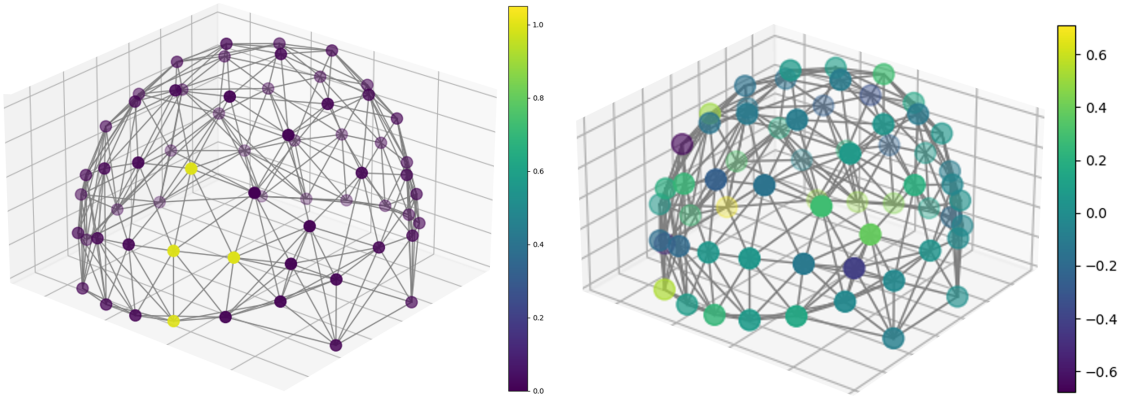


Figure 3.4: First eight eigenvectors (0 to 7) of the k-nn graph with k=8 for the first subject of the database [66]. There are 62 channels and eigenvectors are ordered from eigenvector 0 to eigenvector 61. First eigenvector shows the DC component where all the values are the same and equal to $1/\sqrt{N}$ and variance between the adjacent nodes increases with higher graph frequencies (figure created by the author using PyGSP package for Python [122]).



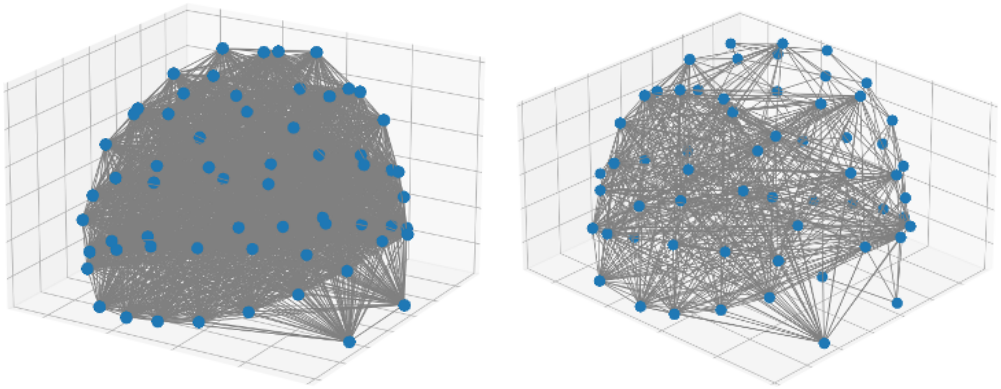
(a) (b)

Figure 3.5: (a) shows a graph signal on Broca's area. The sensors on top of Broca's area of the brain have the value one and the other sensors have the value zero. (b) shows the graph Fourier transform of the signal. Although the signal itself is localized at a small area and all zeros elsewhere, its graph Fourier transform counterpart is not localized and has a range of values which corresponds to reflections onto the eigenvectors of the graph (figure created by the author using [122]).



(a) (b)

Figure 3.6: (a) shows a graph signal on Wernicke's area. The sensors on top of Wernicke's area of the brain have the value one and the other sensors have the value zero. (b) shows the graph Fourier transform of the signal. Although the signal itself is localized at a small area and all zeros elsewhere, its graph Fourier transform counterpart is not localized and has a range of values which corresponds to reflections onto the eigenvectors of the graph (figure created by the author using [122]).



(a) (b)

Figure 3.7: (a) One of the two graphs for 2-cluster dynamic graph for the first subject of database [66]. As we assign different weights for all the combinations between different nodes, the graph is dense. (b) shows the same graph only with the connections with weights greater than 0.75 to show the strong connections (figure created by the author using [122]).

any cardinality. We also avoid using the term *"multigraph"* as it is a term reserved for graphs that can have multiple edges to connect the same two vertices instead of one edge to represent the pair-wise relationship [5].

The modeling of an EEG signal using K graphs requires the identification of K distinct correlation patterns within the graph signal. To this end, we use a sliding window to slice the multi-channel EEG signal into M temporal segments $\tau_m, m = 0, ..., M-1$. For the EEG matrix $\boldsymbol{S} \in \mathbb{R}^{N \times L}$, where N is the number of EEG channels and L is the number of time samples, the m^{th} temporal segment is defined as $\tau_m = [t_1^{(m)}, t_2^{(m)}]$. The m^{th} signal segment from channel i is $\boldsymbol{z}_i^{(m)} \in \mathbb{R}^n$, with $\boldsymbol{z}_i^{(m)} = (\boldsymbol{S}_{i,\tau_m})^\mathsf{T}$, where $n = t_2^{(m)} - t_1^{(m)} + 1$.

Subsequently, for each temporal segment, we calculate correlations between channel pairs. In our graph modeling, the correlation values between channels are directly used as weights in adjacency matrices. If $\boldsymbol{z}_i$, $\boldsymbol{z}_j$ represent signals from two channels, the (i, j) element of the correlation matrix $\boldsymbol{P}^{(m)}$ for the m^{th} segment is calculated as

$$\rho_{ij}^{(m)} = corr(\boldsymbol{z}_i, \boldsymbol{z}_j) = \frac{1}{2}\left(\frac{\boldsymbol{z}_i \boldsymbol{z}_j}{|\boldsymbol{z}_i||\boldsymbol{z}_j|} + 1\right) \tag{3.5}$$

and represents the correlation between channel i and channel j for temporal segment τ_m. As each N-channel signal is divided into M segments, M correlation matrices of size $N \times N$ are obtained from each multidimensional signal, representing cross correlations between different channels for each of the temporal segments. Note that we define the correlation between two signals as (3.5) to keep the range of the values between 0 and 1 as we later use those values as weights in adjacency matrices, and we want to avoid having complex eigenvalues and eigenvectors (Figures 3.7, 3.8).

The varying correlations between the channels of a multidimensional signal (e.g., an

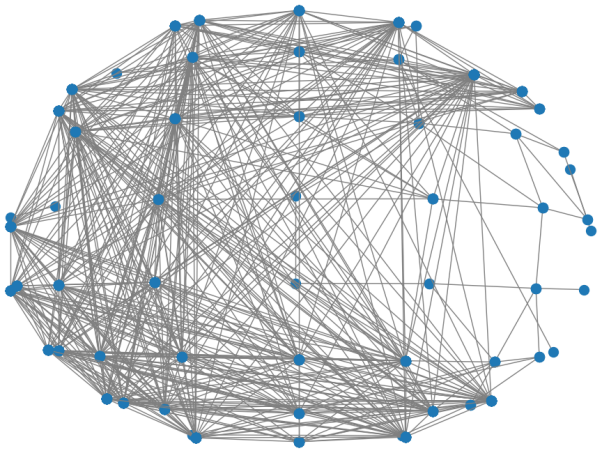


Figure 3.8: Two-dimensional version of the Figure 3.7 for one of the two graphs for 2-cluster dynamic graph for the first subject of database [66]. Only connections with weight greater than 0.75 are kept and the plot shows the top view of the scalp (figure created by the author using [122]).

EEG signal) indicate that multiple adjacencies may need to be taken into account for the graph modeling of EEG signals. To capture the varying correlation between channels over time, we cluster the correlation matrices from different temporal segments using the k-means algorithm [124] and calculate K correlation centers $\boldsymbol{C}^{(k)}$, $k = 0, \ldots, K-1$. For each experiment, we use all the available training data (all subjects and trials) to create K global graphs. In particular, in intra-subject experiments, we use all the training data from the subject, and in cross-subject experiments, we use all the training data from all subjects to create K global graphs based on correlation patterns.

For creating multiple graphs, we use the correlation centers to define K adjacency matrices $\boldsymbol{A}^{(k)}, k = 0, ..., K-1$, as

$$\boldsymbol{A}^{(k)} = \boldsymbol{C}^{(k)} - \boldsymbol{I} \quad \text{for } k = 0, \ldots, K-1 \tag{3.6}$$

where diagonal elements were set to zero to prevent self-loops in the graphs. To analyze the graph spectrum of different graphs, we use the Laplacian matrices

$$\boldsymbol{L}^{(k)} = \boldsymbol{D}^{(k)} - \boldsymbol{A}^{(k)} \quad \text{for } k = 0, \ldots, K-1 \tag{3.7}$$

where the diagonal matrix $\boldsymbol{D}^{(k)}$ is the degree matrix for correlation cluster k, the elements of which are calculated as in (2.6).

For each graph, we get the eigendecomposition

$$\boldsymbol{L}^{(k)} = \boldsymbol{U}^{(k)} \boldsymbol{\Lambda}^{(k)} \left(\boldsymbol{U}^{(k)}\right)^{H} \quad \text{for } k = 0, \ldots, K-1 \tag{3.8}$$

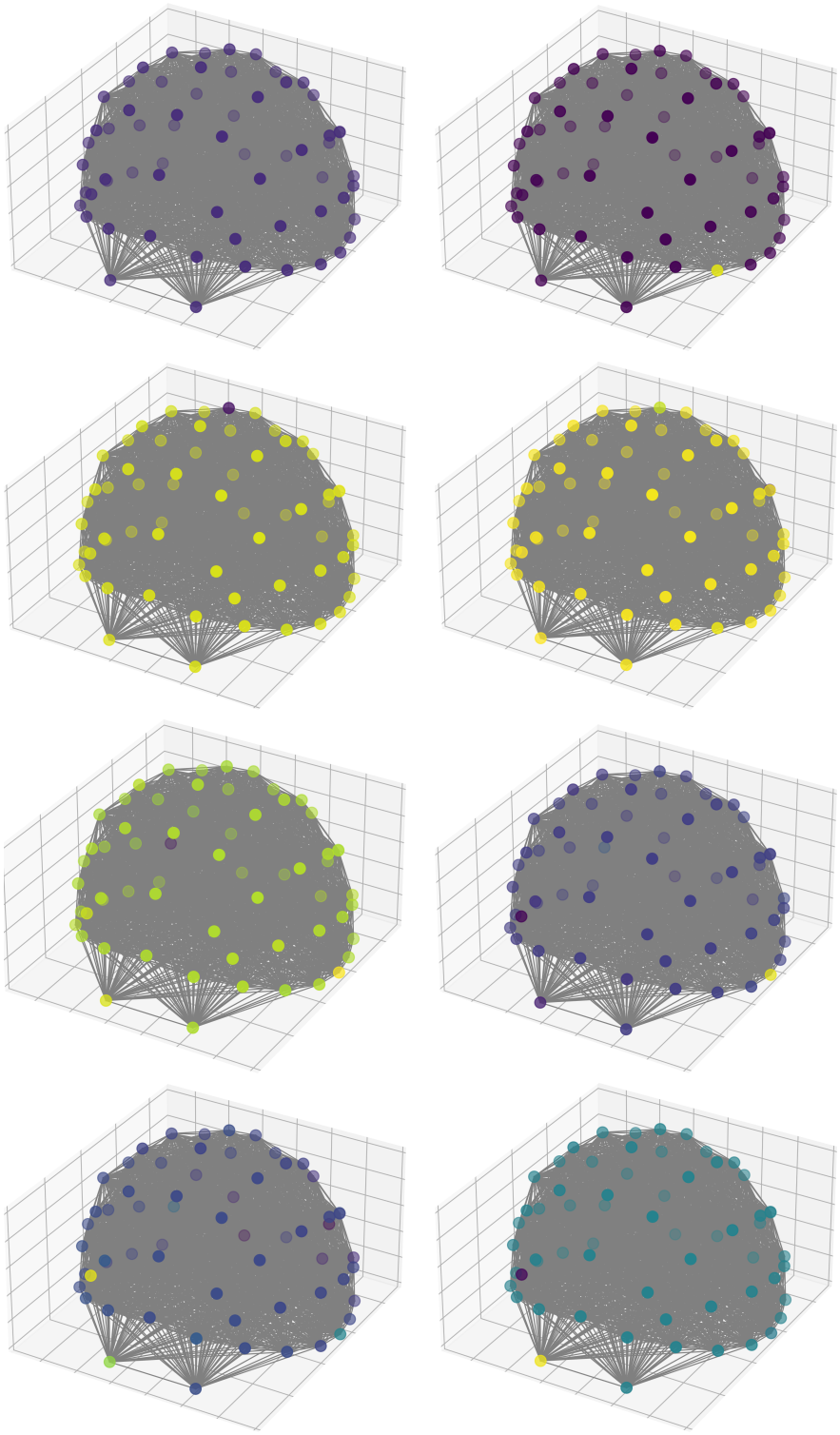


Figure 3.9: First eight eigenvectors (0 to 7) of the first cluster of the dynamic graph for the first subject of the database [66]. There are 62 channels and eigenvectors are ordered from eigenvector 0 to eigenvector 61. Since the connections are not based on proximities and may include links between nonadjacent nodes, the resulting graph structures and their visual representations are not as intuitive as those of proximity-based graphs (figure created by the author using PyGSP package for Python [122]).

with eigenvectors

$$\boldsymbol{U}^{(k)} = \begin{bmatrix} \mathbf{u}_0^{(k)} & \mathbf{u}_1^{(k)} & \dots & \mathbf{u}_{N-1}^{(k)} \end{bmatrix} \quad \text{for } k = 0, \dots, K-1 \tag{3.9}$$

In general, our dynamic graph modeling involves $K(N-1)+1$ eigenvectors. This is because each of the K different graphs $\boldsymbol{\mathcal{G}}^{(k)}$ produces N eigenvectors. However, we discard the constant vectors from all graphs except the first one resulting in $K(N-1)+1$ eigenvectors overall. The larger set of eigenvectors can capture a wider set of graph frequencies (spatial patterns) because the dynamic representation takes into account the varying channel-wise correlations over time (Figure 3.9).

To simplify the above representation, from each correlation cluster we select Δ eigenvectors, indexed $e_0^{(k)}, e_1^{(k)}, \dots, e_{\Delta-1}^{(k)}$. Then we use the selected eigenvectors from each cluster to form the dynamic graph eigenvector matrix $\boldsymbol{\Omega}$.

$$\boldsymbol{\Omega} = \Big[\underbrace{\boldsymbol{u}_{e_0}^{(0)} \boldsymbol{u}_{e_1}^{(0)} \dots \boldsymbol{u}_{e_{\Delta-1}}^{(0)}}_{\text{from first cluster}} \dots \underbrace{\boldsymbol{u}_{e_1}^{(K-1)} \dots \boldsymbol{u}_{e_{\Delta-1}}^{(K-1)}}_{\text{from last cluster}} \Big] \tag{3.10}$$

The eigenvectors in $\boldsymbol{\Omega}$ represent different graph frequencies from all graphs within a dynamic graph representation. These graphs capture the dynamic relationships between vertices. By projecting each original N-dimensional signal $\boldsymbol{s}$ onto the eigenvectors of the dynamic graph eigenvector matrix $\boldsymbol{\Omega}$, we obtain the new representation $\hat{\boldsymbol{s}} = \boldsymbol{\Omega}^H \boldsymbol{s}$. For the entire original EEG signal (N-dimensional time series), the new representation is

$$\hat{\boldsymbol{S}} = \boldsymbol{\Omega}^H \boldsymbol{S} \tag{3.11}$$

where $\hat{\boldsymbol{S}}$ is a $K\Delta$-dimensional series of graph frequency coefficients. The resultant representation is passed as input to the neural network architecture, which is thoroughly explained in Section 4.2.

We presented different adjacency components for different setups from Figure 3.10 through Figure 3.13. As illustrated in the figures, the differences between the adjacency components decrease. In general, using a larger window size increases the similarity among adjacency components, undermining the objective of having a variety in adjacency components. Additionally, we observed that using overlap does not affect the results but increases computational cost; therefore, we avoid using overlaps while setting the window size to 16 ms.

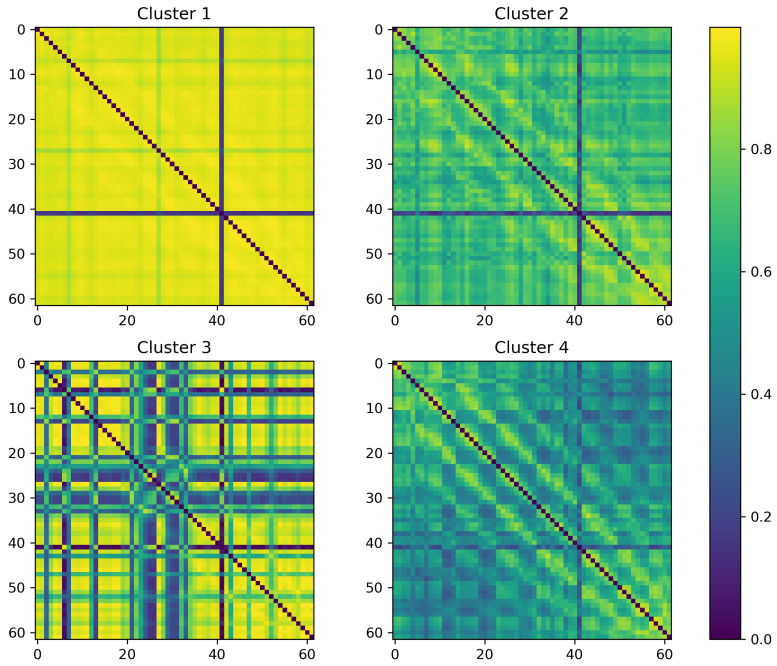


Figure 3.10: Dynamic graph adjacency matrices for four clusters when the window size is set to 16ms. Using a smaller window increases the difference between different clusters compared to larger windows (figure created by the author).

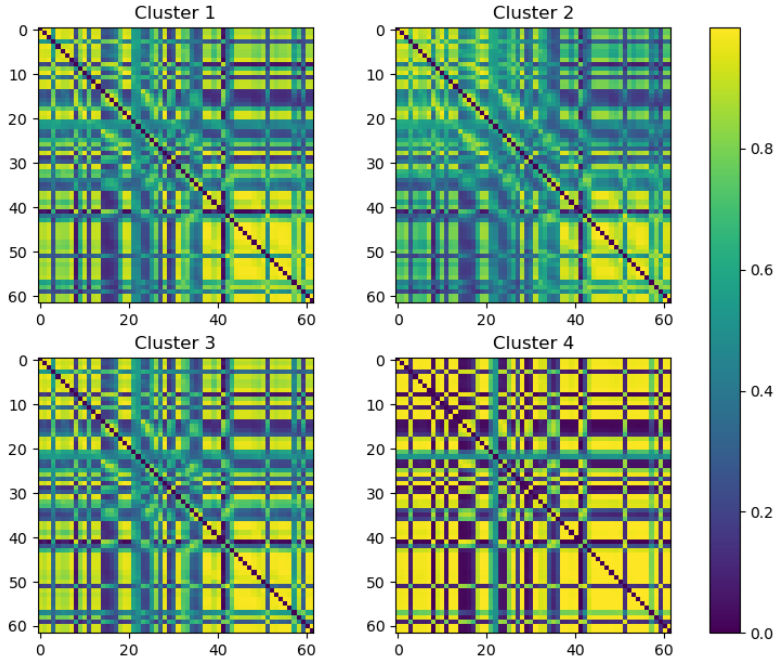


Figure 3.11: Dynamic graph adjacency matrices with four clusters when the window size is set to 100ms (figure created by the author).

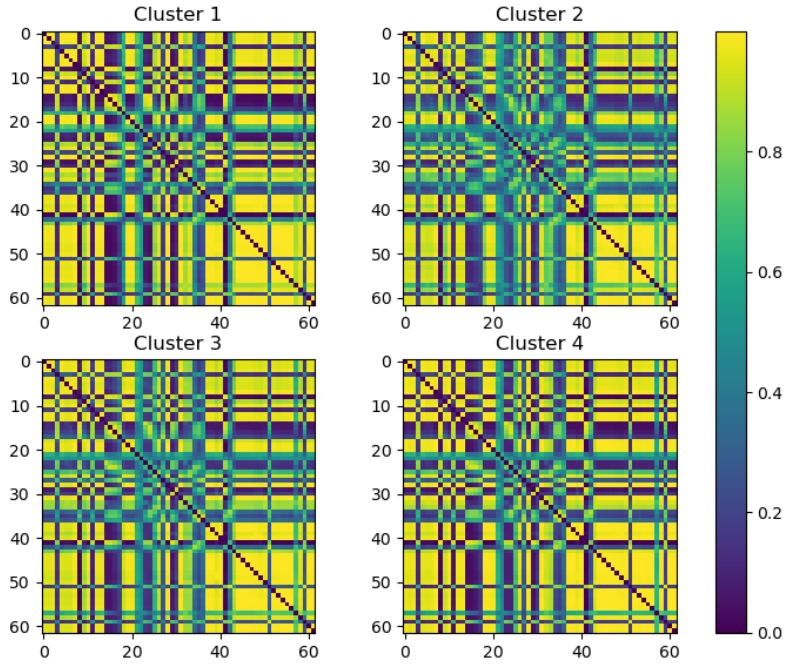


Figure 3.12: Dynamic graph adjacency matrices with four clusters when the window size is set to 4800ms. A very large window size increases the similarity of the clusters compared to smaller window sizes (figure created by the author).

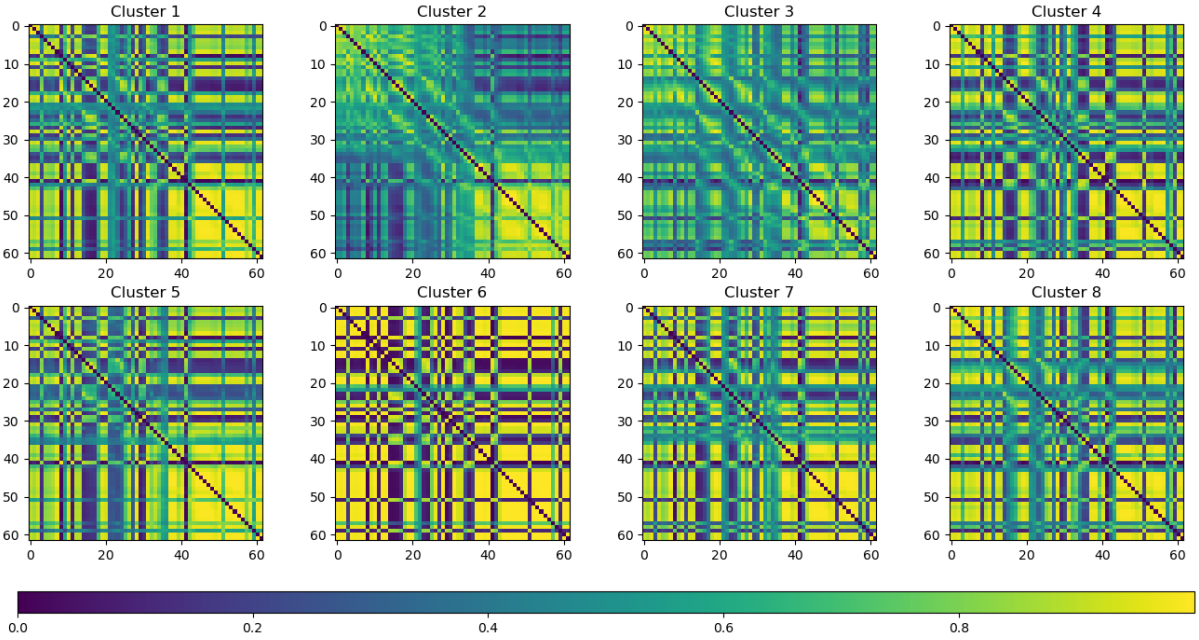


Figure 3.13: Dynamic graph adjacency matrices for eight correlation clusters (figure created by the author).

3.4 Spatiotemporal Representation of the Graph EEG Signals

The graph-based modeling defined in Section 3.3 produces time series of coefficients representing spatial frequencies (spatial patterns) over time. To account for temporal variations in the spatial frequency components, we calculate the STFT magnitude of each graph frequency coefficient time series separately as shown in Figure 3.14. The resultant *Dynamic Multiple Graph Fourier* (DMGF) representation expresses the original multidimensional signal with respect to how its spatial patterns change over time.

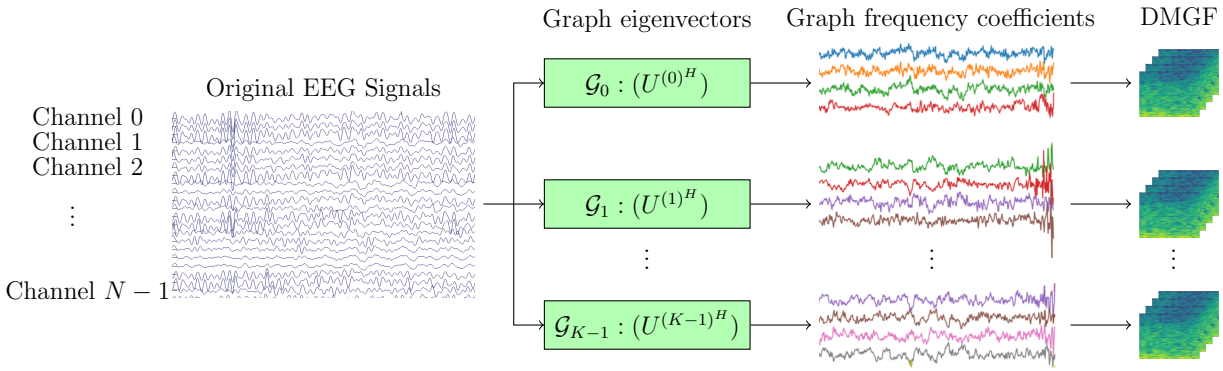


Figure 3.14: Spatiotemporal representation of EEG signals. EEG signals are projected onto the eigenvectors of different graphs and spatial frequency components obtained from each graph. We calculate the STFT of each component separately to get the Dynamic Multiple Graph Fourier (DMGF) representation that captures frequencies both in graph space and time (figure created by the author).

In order to prepare the DMGF for neural network classification, we offset STFT magnitude values by 1 to make the smallest magnitude equal to 1. We then apply log-normalization to ensure that the STFT magnitude values remain within a numerically stable range for the neural network. In this way, the minimum log-magnitude is 0 while small magnitudes, such as 10^{-20}, are represented as approximately 0. The resultant representation is graphically shown in Figure 3.15.

3.5 Conclusion

In the first part of this chapter, we introduced the graph representations of EEG signals by constructing an 8-nearest-neighbor (k-nn) graph from EEG sensor positions, similar to the grid graph presented in Section 2.1. Section 3.2 included illustrations of the graph itself and its eigenvectors, analogous to those provided in Section 2.1. Additional visualizations were provided to demonstrate how spatial variance changes with increasing

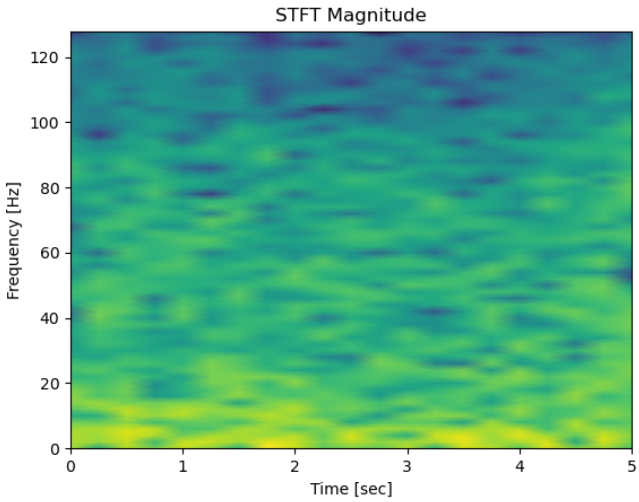


Figure 3.15: A sample DMGF (Dynamic Multiple Graph Fourier) representation that is input to the neural network. Multiple such representations are input simultaneously, one per graph frequency coefficient sequence (figure created by the author).

spatial frequencies. This simple representation constitures the main method used in the literature to model the network of sensors with one static graph based on Euclidean distances between sensor pairs, and to process signals defined on such graphs. This causes all the EEG applications to have almost the same graph structure for calculating the graph frequencies. Creating a single graph using Euclidean distances of sensors on the scalp based on the international 10-20 or 10-10 placement systems, results in almost identical graphs structures for all the applications, i.e., imagined speech classification, emotion recognition, biometrics, motor imagery. We address the shortcoming of this approach and how the graph structure of the sensors on top of different regions of the brain changes through time and task and why this is problematic. Depending on the individual, the type of activity, and its duration on the activity, the graph structure may vary, as it represents the pairwise relationships between different regions of the brain. Although there are variations of the Euclidean distance based static graphs, i.e., using two neighbors instead of eight, they remain fundamentally similar, as all are static and derived from sensor proximity.

This chapter provided the theoretical foundations for the following chapters, which are the applications of the theory represented in this chapter. One of our aims was to develop a dynamic-graph-based modeling strategy for EEG signals that captures evolving spatial and temporal relationships among different brain regions. To address those issues of using static graphs for the representation of EEG signals, in Section 3.3, we proposed a novel method for constructing dynamic graph representations of EEG data that model the time-varying, task-specific, and subject-specific relationships among different brain regions, independent of their physical proximity. Our proposed approach can model the

dynamic nature of the EEG signals to capture different network structures that change throughout the activities, subjects and duration.

We address the shortcoming of GSP as it gives the spatial variation for a fixed time instant, and we presented all the mathematical operations of EEG signal to explicitly show how the GSP represents the spatial frequencies only for a fixed time instant and how the multi-channel signals turn into a sequence of spatial-frequencies. This lacks the temporal variations. We address that we cannot take advantage of GSP by itself as it lacks the spatiotemporal variations by itself. In Section 3.4, we provided our proposed spatiotemporal representation of graph signals to extract and analyze spatiotemporal frequency features from the constructed dynamic graph representations of EEG signals.

In the ensuing chapters, we used this dynamic graph representation of the EEG signals for imagined speech classification and biometric identification applications.

Chapter 4

Machine Learning Architectures and Data Augmentation for EEG Signal Classification

4.1 Introduction

In Chapter 3, we presented the dynamic graph representation of graph signals and how to represent the spatiotemporal variations of the graph signals to capture both spatial and temporal variations of the graph signals. To use the resulting representations for any kind of applications like *imagined speech* classification or *brain computer interface* (BCI) applications, we need to feed the spatiotemporal representation of graph signals as inputs to a neural network architecture. However, using a machine learning based classifier itself is not straightforward as the performance and accuracy of neural network architectures depend on the task and data type. Furthermore, depending on the chosen number of clusters to represent the graph signals, the classification accuracy will change. The main points that we aim to investigate are optimizing the number of clusters and eigenvector selection process, choosing the suitable neural network architecture for the classification task, and data augmentation to increase the number of samples to make the training phase more robust and increase the classification accuracy.

As we stated in previous chapters, there are various applications [20, 21] of electroencephalography (EEG) signals. We stated some of them as classifying diseases and neurological disorders, sleep analysis, and brain-computer interfaces (BCI), or in general human-machine interfaces (HMI) [23]. A particular EEG application, which has received relatively less attention, is the recognition of imagined speech, i.e., the interpretation of the specific words *imagined* by a subject. This application can have important applications, such as communication for coma patients or other brain-computer interfaces. Although some methods focusing on this application have been proposed in the past,

these are mostly based on signal analysis approaches that extract features from each EEG channel separately. Such analysis neglects the structure of the graph of sensors as well as the spatial relationships between them. To jointly assess EEG signals, one would need a different representation, based on Graphs.

We used our proposed dynamic graph representation of EEG signals that we described in Chapter 3 for the imagined speech classification and used a machine learning based classification method to classify different imagined words. We presented the details of the experiments and the results in Section 4.3 in detail, and we present the overview of the method in Figure 4.1. As we mentioned the importance of the graph structure

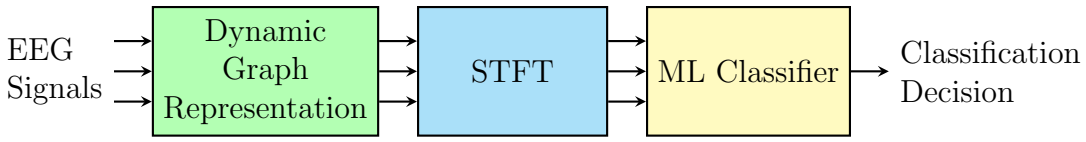


Figure 4.1: Method overview (figure created by the author).

and the channel-wise correlations between different sensors and how important it is to incorporate the spatial and temporal variations of graph signals; we want to point out the gap in literature about creating augmented EEG data without modifying the graph structure and channel-wise relationships. For this reason, we also introduced a novel EEG data augmentation method to increase the number of samples without changing the channel-wise relationships of EEG sensors.

The main contributions of the present chapter are:

- The application of the theory of dynamic graph representations for EEG signals, as proposed in Chapter 3, to imagined speech recognition.
- An LSTM-based machine learning architecture for the classification of imagined words.
- An investigation of the impact of different neural network architectures on the performance of imagined speech classification.
- An investigation of the effect of different clustering and eigenvector selection strategies on the performance of dynamic graph modeling of EEG signals.
- A novel graph-based method for generating artificial EEG data that preserves channel-wise relationships to increase the number of training samples.
- A thorough comparison of the resulting system with current state-of-the-art systems in imagined speech recognition.

and the structure of the chapter is as follows:

- Section 4.2 presents the neural network architecture that we used for our method.
- Section 4.3 presents extensive ML experiments conducted to evaluate the effectiveness of the dynamic graph representation of EEG signals. Various strategies for selecting the properties of the dynamic graph representation, such as the number of clusters and the indices of selected eigenvectors, are explored. The section details the databases employed in the experiments and describes the experimental procedures used for the assessment of our method. The effect of different numbers of clusters on the classification accuracies is analyzed, alongside an evaluation of different eigenvector selection strategies. Several neural network architecture variants for the classification task are introduced, and their respective advantages and disadvantages are compared. Results obtained on the dataset from [98] are reported and compared with the benchmark studies. Results from both intra-subject and cross-subject classification tasks are presented, demonstrating improved performance of the proposed model in a subject-independent setup. Additional results are provided using the Kara One database [66] to further evaluate the effectiveness of the method.
- Section 4.4 introduces a novel method to generate artificial EEG data without altering the graph structure or the channel-wise relationships. The section presents the experiment results with our proposed EEG data augmentation method and compares them with the results without augmented EEG data.
- Finally, conclusions are drawn in Section 4.5.

4.2 Neural Network Architecture for Imagined Speech Classification

For the classification task we used a three-layer LSTM network with attention mechanism as shown in Figure 4.2. The input signal is projected onto the eigenvectors of the dynamic graph representation, yielding a set of graph frequencies for each of the different graphs. The subsequent STFT of each graph frequency time series captures the variations of dynamic graph frequencies through time. The resultant DMGF coefficients are used as inputs to LSTM network. To simplify the representation and reduce the number of features input to the LSTM network, we calculate the average of every four temporal frequency component of each graph frequency. The number of features input to the LSTM network is the number of graph frequencies multiplied by the number of temporal frequencies.

As seen in Figure 4.2, we used three bidirectional LSTM layers. We chose the LSTM for such modeling as LSTMs are robust at learning long-term and short-term depen-

dencies in long temportal sequences, like the ones used in the present work. The small number of LSTM layers limits the number of parameters in the neural network, reduces the computational cost, and prevents overfitting. We used 256 parameters for each layer and did not use any dropout to regulate overfitting, as the model itself converges well and dropout induces oscillations in the loss function on small validation sets. Each spatiotemporal graph frequency in the DMGF representation is used as a different channel/feature which is input to the first LSTM layer, while the cascaded LSTM layer learns how those features evolve through time. The attention mechanism at the end of the last LSTM layer to ensure that the neural network focuses on the most relevant features at the end of the cascaded LSTM layers.

4.3 Experimental Assessment

4.3.1 Databases

This work focuses on publicly available EEG datasets to make a thorough comparison with the current state-of-the-art methods. In some cases results might be high because of the way the data is collected [98, 125]. There are publicly available EEG datasets for imagined speech [82] for benchmarking and comparing different methods. It is significant to use those datasets to see how a method performs compared to the performance of other methods that use the same datasets. The works [66, 98, 126–135] share EEG datasets publicly. However, they are not all suitable for this work. In this work we focused on EEG datasets [98] and [66] for a variety of reasons. First, we limit the language prompts to English. [126], [127] and [128] have Spanish prompts. [132] has prompts in Dutch. [130] has English and Chinese prompts. Another concern is the number of channels for the EEG recordings. [129] has only 14 channels. [133] has 16 channels, and [134] has 14 channels. To apply GSP, we focused on datasets with more than 60 channels and discarded datasets with a small number of channels. [131] has 64 channels but only has four subjects, which is not favorable for cross-subject evaluations. For these reasons, for the experimental evaluation of our method, we focused on [66] and [98] for the imagined speech recognition experiments as [66] has 12 subjects and 62 channels and [98] has 15 subjects and 64 channels.

The database in [98] includes four different experiments, namely, *short word* classification, *long word* classification, *vowel* classification and *short word vs long word* classification. In short word vs long word experiment, the subject sees the short word *"in"* or the long word *"cooperate"* as visual cues on a computer monitor and the subject imagines speaking those words without moving their lips or any other muscles. The highest accuracy reported in [98] came from the short vs long word classification, which we used for comparison, included only the highest results from [98] in our comparison tables. As only

a subset of the subjects contributed trials for the short word vs long word experiment, we represent them as subjects A, B, C, D, E, F instead of their IDs, for clarity.

The second database that we used is the Kara One database [66]. Although the database also includes video and audio data, we only used the EEG signals collected while subjects were performing speech imagery. There are 14 subjects performing seven phonemic/syllabic prompts which are /iy/, /uw/, /piy/, /tiy/, /diy/, /m/, /n/ and four different words *"pat"*, *"pot"*, *"knew"* and *"gnaw"*, with around 12 samples per class. For comparison, we perform the same five binary classification experiments as [66, 99–101], and in Section 4.3 we compared the results and presented them in Table 4.9.

4.3.2 Experiments

We applied the dynamic graph representation methodology explained in Chapter 3 to the databases described in Section 4.3.1. To model brain signal activity during imagined speech tasks, we created multiple graphs by clustering correlation matrices, calculating multiple adjacencies, and creating a separate graph for each adjacency matrix. For each EEG trial, we calculated the respective graph frequencies and we applied STFT on the respective temporal sequences of graph frequencies, producing a Dynamic Multiple Graph Fourier (DMGF) representation. An example of our DMGF representation is shown in Figure 3.15. We used this representation as an input to a three-layer LSTM network [73,75] with attention to classify brain signal activity during imagined speech. A summary of our main scheme is shown in Figure 4.2.

We concatenated the spatial-frequency and temporal-frequency dimensions of the DMGF data structure and used the resultant structure as input to the LSTM layers. The 4D output of the STFT is in the form of (trials $\times$ graph frequencies $\times$ time frequencies $\times$ time) and after concatenation it acquires a 3D shape (trials $\times$ graph/time frequency combinations $\times$ time). For intra-subject (subject-dependent, per-subject) experiments, we used a 10-fold cross validation [85] scheme to validate our results. For the cross-subject (subject-independent, combined data) experiments, we used each subject separately as test set. Results were averaged across subjects.

4.3.3 Exploration of Different Number of Correlation Clusters

For fine-tuning our dynamic graph representation, we explored the impact of the number of clusters on classification accuracy. To this end, we used the database [98] and ran different experiments using one, two, and four clusters. We repeated the experiments on Subject A and Subject D as Subject A performs worse than other subjects and Subject D performs better. We present the results in Table 4.1. As seen, using a higher number of clusters generally improves performance, while using only one cluster performance is approximately 3% lower.

Table 4.1: Classification accuracies using different numbers of correlation clusters using the database [98]. As seen, using more clusters help with classification accuracy. Results are averaged for all subjects.

Classification accuracies using different number of correlation clusters.				
Correlation clusters	Long words	Short vs long words	Vowels	Short words
1 cluster	71.92±5.27%	85.77±5.87%	52.54±3.44%	55.06±3.32%
2 clusters	73.08±5.43%	86.83±4.51%	53.50±3.71%	56.28±2.83%
4 clusters	**74.75**±5.36%	**88.58**±3.91%	**55.08**±2.59%	**57.33**±2.52%
[98]	66.18±5.78%	80.05±5.80%	48.96±6.21%	50.07±7.60%
[103]	62.99±4.78%	N/A	N/A	N/A
[104]	71%	70%	46%	44.2%
Chance level	50%	50%	33%	33%

4.3.4 Exploration of Different Eigenvector Selection Strategies

For the purpose of assessing the discrimination capacity of graph frequencies in imagined speech recognition, we used subsets of graph frequencies. We present the results in Table 4.2, where the representation using the lowest n eigenvectors is denoted as GSP_n, i.e., GSP_{20} is a representation using the first/lowest 20 eigenvectors (out of a total of 64). As seen in Table 4.2, using a subset of eigenvectors yields comparable results to using all eigenvectors. This means that the coefficients representing low spatial frequencies carry most discriminatory capacity and can be used reliably for classification of imagined speech. However, the best performance is achieved when high graph frequencies are also taken into account, which suggests that high graph frequencies are also useful in imagined speech.

Table 4.2: Classification accuracies using a subset of the eigenvectors for the short word vs long word experiment using database [98].

Classification accuracies using a subset of the eigenvectors		
Method	**Subject A**	**Subject D**
Raw signal + STFT	61.50 ± 3.20%	70.63 ± 4.00%
Raw signal + ICA + STFT	63.00 ± 4.00%	72.50 ± 5.00%
Dynamic GSP_5 + STFT	68.00 ± 3.32%	76.88 ± 4.88%
Dynamic GSP_{10} + STFT	71.50 ± 5.02%	80.00 ± 3.75%
Dynamic GSP_{20} + STFT	75.00 ± 3.87%	86.25 ± 3.75%
Dynamic GSP_{40} + STFT	78.50 ± 3.20%	91.88 ± 4.00%
Dynamic GSP_{64} + STFT	**81.50** ± 3.91%	**93.75** ± 4.84%
Benchmark Accuracy [98]	70.3 ± 5.5%	88.0 ± 6.4%

To confirm that the dynamic graph representation leads to improved performance, we deployed an experimental setup without any graph representations, which served as

a baseline system (shown in Figure 4.3(a)). The baseline system uses raw EEG signals as input to the STFT, i.e., no graph representation and no graph frequencies are used. In addition, we repeated the experiment (on raw EEG signals) using Independent Component Analysis (ICA) by using the algorithm defined in [136], and applied the STFT to produce the data to be fed into the neural network (Figure 4.3(b)). We did not use any graph representations.

Results using the two baseline methods are presented in Table 4.2. As seen, even using only five eigenvectors with our 4-cluster dynamic graph representation, it is possible to outperform the two baseline methods that do not use graph signal representations. This confirms the suitability and effectiveness of our graph representation for imagined speech classification.

We later repeated the experiments with high spatial frequencies to see if using only high spatial frequencies can give comparable results. We present the results in Table 4.3 and show how different eigenvector selections result in different classification accuracies. In general, using lower eigenvectors give higher results with lower variance, but it is still possible to reach those accuracies with high spatial frequencies too. This again shows that the information is spread through all the spatial frequencies. Using a subset of eigenvectors with low or high spatial frequencies gives comparable results, however, we lose information when we do not use all the eigenvectors.

Table 4.3: Classification accuracies using a subset of the eigenvectors for the short word vs long word experiment for subjects A and D using the database [98].

Classification accuracies using a subset of the eigenvectors		
Method	**Subject A**	**Subject D**
Lowest 5 eigenvectors (0, 1, 2, 3, 4)	68.00 ± 3.32%	76.88 ± 4.88%
Highest 5 eigenvectors (59, 60, 61, 62, 63)	66.50 ± 3.20%	74.38 ± 5.90%
Lowest 10 eigenvectors (0, 1, ..., 9)	71.50 ± 5.02%	80.00 ± 3.75%
Highest 10 eigenvectors (54, 55, ..., 63)	70.00 ± 4.47%	78.75 ± 3.06%
Lowest 20 eigenvectors (0, 1, ..., 19)	75.00 ± 3.87%	86.25 ± 3.75%
Highest 20 eigenvectors (44, 45, ..., 63)	73.00 ± 4.58%	83.13 ± 2.86%
Lowest 40 eigenvectors (0, 1, ..., 39)	78.50 ± 3.20%	91.88 ± 4.00%
Highest 40 eigenvectors (24, 25, ..., 63)	77.50 ± 2.50%	90.63 ± 3.13%
All eigenvectors (0, 1, ..., 63)	**81.50** ± 3.91%	**93.75** ± 4.84%
Benchmark Accuracy [98]	70.3 ± 5.5%	88.0 ± 6.4%

4.3.5 Exploration of Different Neural Network Variations

We repeated the experiments using three alternative neural network architectures for classification. The main reason of choosing an LSTM-based neural network as our main architecture was their robustness in learning long-term and short-term dependencies in

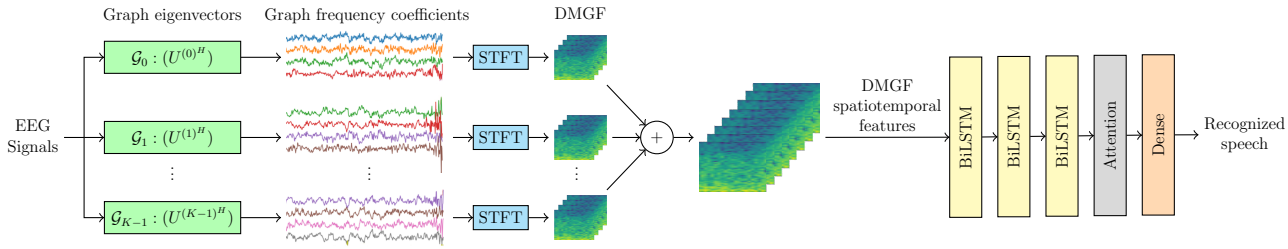


Figure 4.2: Main architecture for classification of imagined speech. Multi-channel EEG signals are projected onto the eigenvectors of each graph and yield timeseries of graph (spatial) frequency coefficients. These are subjected STFT, producing the proposed DMGF representation. The resulting DMGF coefficients are fed into an attention-based BiLSTM network for the recognition of imagined speech (figure created by the author).

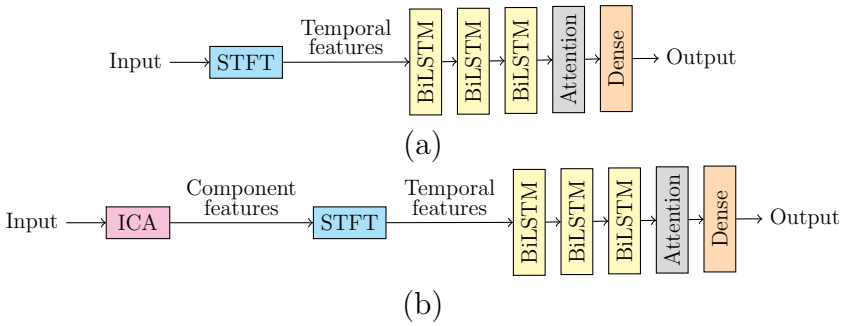


Figure 4.3: (a) Baseline method without any graph operators. (b) Baseline method with ICA (figure created by the author).

long temporal sequences, which is what we use in this work. However, convolutional neural networks are also used in clasification of frequency features of imagined speech [102] and multi-channel convolutional neural networks [137] are used in classification of EEG signals in some works [28, 102]. As we explained in Section 2.3.3, we avoided using transformers because of the size of the EEG datasets. For these reasons, we compared our LSTM-based architecture with a variety of combination of LSTM, convolutions, and multi-channel architectures to assess how using convolutions and multi-channel architectures perform compared to our single-channel LSTM-based neural network. These are explained below.

Multi-channel 3-BiLSTM with attention (Figure 4.4(a)) We tested our methodology using two separate graphs and separate graph eigenvector sets. We calculated the STFT of the resulting graph frequencies and provided them as inputs to different LSTM layers. This approach is similar to the main method, showed in Figure 4.2, but processes the features of different graphs separately. The attention layers at the end of the LSTM layers facilitate the selection of the most relevant features for the classification task. The concatenation of the resulting outputs was fed into a dense layer with a softmax function

that calculates the probability of each class.

Multi-channel 5-Conv2D with attention (Figure 4.4(b)) This architecture uses five 2D convolution layers instead of LSTM layers. The low number of LSTM layers and convolution layers helps keep computational cost low and avoid overfitting.

1-channel 5-Conv2D with attention (Figure 4.4(c)) This architecture uses 2D convolutional layers on a single channel, taking as input the same graph representation used by the main architecture.

Results obtained using the above architectures are presented in Table 4.4. As seen in Table 4.4, the main scheme (Figure 4.2) outperforms the alternative architectures, providing evidence that the dynamic graph EEG representation combined with a single-channel LSTM-based neural network is the architecture that yields the best recognition results.

Table 4.4: Classification accuracy using alternative neural networks for short word vs long word classification using database [98].

Classification accuracy using alternative neural networks.		
Neural network variation	**Subject A (Low performer)**	**Subject D (High performer)**
1-channel 3-BiLSTM + attention	**81.50** ± 3.91%	**93.75** ± 4.84%
2-channel 3-BiLSTM + attention	79.00 ± 4.36%	91.25 ± 4.15%
2-channel 5-Conv2D + attention	80.00 ± 3.87%	92.50 ± 5.45%
1-channel 5-Conv2D + attention	78.00 ± 5.10%	89.38 ± 5.63%
Benchmark method [98]	70.3 ± 5.5%	88.0 ± 6.4%

Table 4.5: Performance of our 4-cluster method with 3-layer LSTM network and attention mechanism for short word vs long word classification in comparison to the benchmark accuracies [98] using the database [98].

Classification accuracy for short word vs long word on database [98].		
Subject	**Proposed method**	**Benchmark method** [98]
Subject A	**81.50** ± 3.91%	70.3 ± 5.5%
Subject B	**89.50** ± 3.50%	71.5 ± 5.0%
Subject C	**91.25** ± 5.00%	81.9 ± 6.5%
Subject D	**93.75** ± 4.84%	88.0 ± 6.4%
Subject E	**81.50** ± 3.20%	79.3 ± 7.9%
Subject F	**94.00** ± 3.00%	89.3 ± 3.5%
Average	**88.58** ± 3.91%	80.1 ± 5.8%

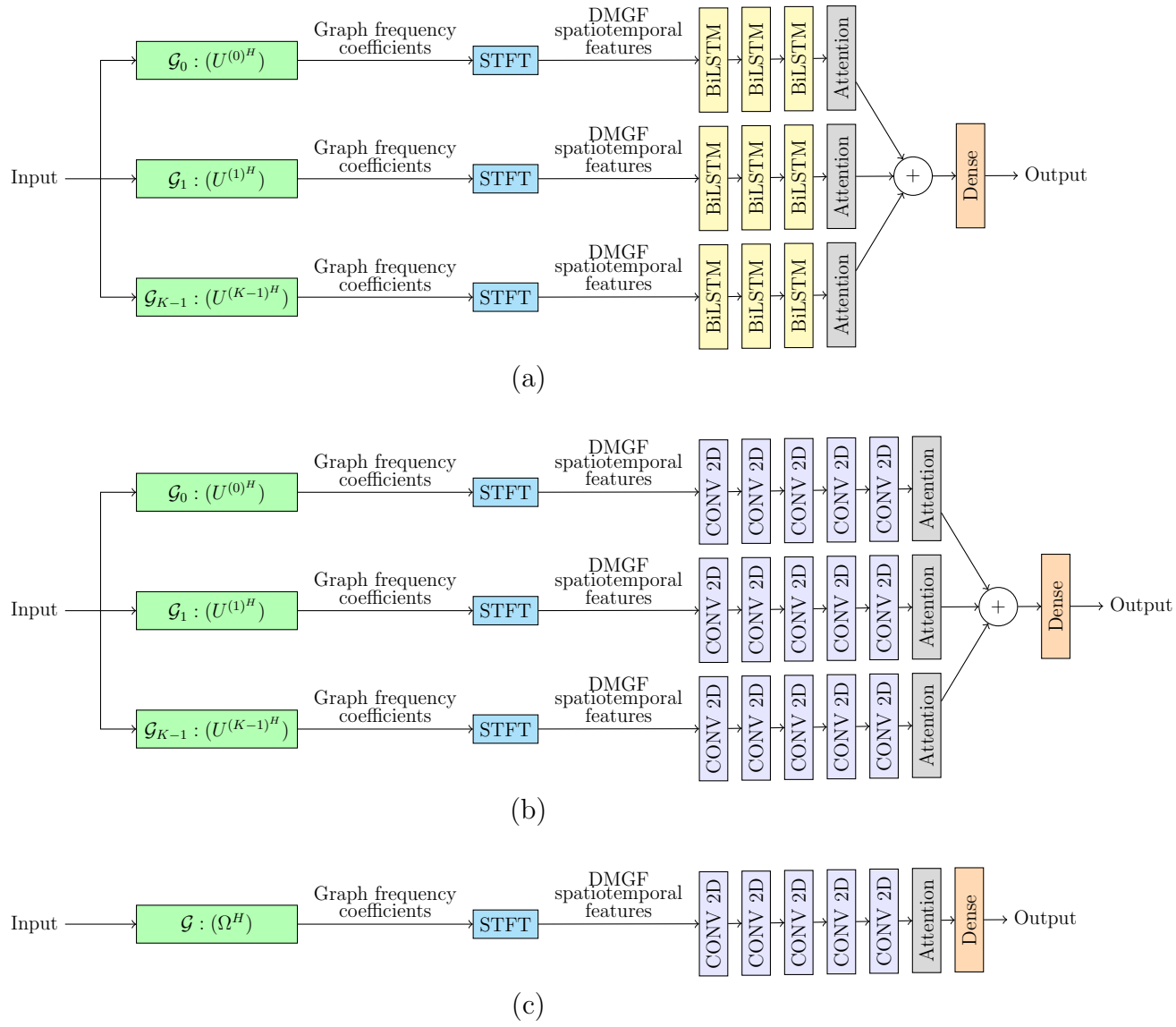


Figure 4.4: Neural network architecture variants used with the proposed method. (a) Multi-channel 3-BiLSTM with attention, (b) Multi-channel 5-Conv2D with attention and (c) 1-channel 5-Conv2D with attention (figure created by the author).

4.3.6 Results on Dataset [98]

We tested our method for imagined speech classification involving the short word *"in"* and long word *"cooperate"*. Results are presented in Table 4.5, with alternative metrics provided in Table 4.6 and Table 4.7. As seen, our method not only outperforms the benchmark method, but also exhibits lower variance across subjects, making it more suitable for subject-independent applications.

Further, we assessed our system in cross-subject (subject-independent) classification where data from all subjects are used for system training. In general, EEG classification is more difficult when signals from different subjects are used for training and testing [65], as opposed to using signals from one subject. Our method is inherently

Table 4.6: Confusion matrix for each subject for short vs long word experiment. Subjects A, B, E and F have 200 samples, and Subject C and D had 160 samples.

Subject A		
	Predicted Class 1	**Predicted Class 2**
Class 1 - Short word (*"in"*)	TP = 80	FN = 20
Class 2 - Long word (*"cooperate"*)	FP = 17	TN = 83

Subject B		
	Predicted Class 1	**Predicted Class 2**
Class 1 - Short word (*"in"*)	TP = 92	FN = 8
Class 2 - Long word (*"cooperate"*)	FP = 13	TN = 87

Subject C		
	Predicted Class 1	**Predicted Class 2**
Class 1 - Short word (*"in"*)	TP = 75	FN = 5
Class 2 - Long word (*"cooperate"*)	FP = 9	TN = 71

Subject D		
	Predicted Class 1	**Predicted Class 2**
Class 1 - Short word (*"in"*)	TP = 73	FN = 7
Class 2 - Long word (*"cooperate"*)	FP = 3	TN = 77

Subject E		
	Predicted Class 1	**Predicted Class 2**
Class 1 - Short word (*"in"*)	TP = 79	FN = 21
Class 2 - Long word (*"cooperate"*)	FP = 16	TN = 84

Subject F		
	Predicted Class 1	**Predicted Class 2**
Class 1 - Short word (*"in"*)	TP = 92	FN = 8
Class 2 - Long word (*"cooperate"*)	FP = 4	TN = 96

robust in subject-independent classification tasks because the clustering of correlation matrices from different subjects models task-dependent relationships between different graph vertices instead of subject-specific traits. Therefore, to emphasize features relevant to imagined speech, we create dynamic graphs by combining data from all subjects and imagined words. Results are presented in Table 4.8 and show that our method sustains a 75% accuracy in the cross-subject scenario, which confirms the effectiveness of our approach.

Table 4.7: Performance metrics for each subject for short vs long word experiment. Subjects A, B, E and F have 200 samples, and Subject C and D had 160 samples. The model demonstrates consistent precision, recall, F1 score and accuracy across all subjects, indicating the robustness of its classification performance.

Performance metrics for short vs long word experiment.				
Subject	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Subject A	0.82	0.80	0.81	0.82
Subject B	0.88	0.92	0.90	0.90
Subject C	0.89	0.94	0.91	0.91
Subject D	0.96	0.91	0.94	0.94
Subject E	0.83	0.79	0.81	0.82
Subject F	0.96	0.92	0.94	0.94

Table 4.8: Performance comparison of intra-subject and cross-subject classification in the short-word vs. long-word experiment using the database in [98].

Classification accuracies for short word vs long word classification.		
Task	**Proposed**	**Benchmark** [98]
Intra-subject classification	**88.58** ± 3.91%	80.1 ± 5.8%
Cross-subject classification	**75.89** ± 2.11%	N/A

Table 4.9: Cross-subject classification accuracies on KARA One database [66].

Classification accuracies on KARA One database [66]					
Method	**C/V**	**±Nasal**	**±Bilabial**	**±/iy/**	**±/uw/**
[66]	18.08%	63.50%	56.64%	59.60%	79.16%
[99]	25.00%	47.00%	53.00%	53.00%	74.00%
[100]	85.23%	73.45%	**75.55%**	73.30%	81.99%
[101]	86.52%	72.10%	69.08%	75.27%	83.98%
Proposed	**90.06**±1.98	**76.73**±1.48	71.71±1.60	**89.56**±1.47	**91.08**±1.81

4.3.7 Results on Kara One Database [66]

In addition to the experiments conducted on database [98], we ran five different binary classification tasks using the Kara One database [66]. These classifications tasks are: *vowel-only vs consonant (C/V)*, *presence of nasal (±Nasal)*, *presence of bilabial (±Bilabial)*, *presence of high-front vowel (±/iy/)*, and *presence of high-back vowel(±/uw/)*. We tested our system in subject-independent classification, where one subject is kept aside for testing, while the remaining subjects are used for training. The results are averaged over all subjects and presented in Table 4.9 in comparison to results from relevant benchmarks. As seen, our results exceed the benchmark accuracies for most experiments. This shows that our method is robust for subject-independent applications and can be reliably used for imagined speech classification.

4.4 Data Augmentation

EEG datasets have very limited sample sizes (less than 200 trials per subject) compared to datasets on other fields. Most of the EEG datasets have less than 50 trials per subjects and a lot of them have less than 20 subjects. For example, one of the popular EEG datasets Kara One database [66] is used a lot as a benchmark to compare different methods on EEG data and it has only 14 subjects with around 12 samples per class. Another popular EEG database PhysioNet EEG database [114] has 109 subjects, yet each subject has 15 samples per session with 45 samples in total of 3 sessions. Although EEG epoching is used to increase the number of samples, it is still limited. For example, one sample in Kara One database is approximately 5 seconds, and if you extract 9 epochs with 1 second of recording and 50% overlap, we can increase the number of samples per class from 12 to 108 for each subject. However, even with EEG epoching, the number of trials that we have is still a small fraction of what we have for image datasets. For cross-subject experiments, the data from different subjects are combined and that increases the number of trials per class, however, cross-subject experiments are more difficult by their nature as the signals vary too much for different subjects. For intra-subject experiments, which uses data from only one subject alone, that makes the assessment more problematic as less data means more bias on the results.

Generative Adversarial Networks (GANs) [94, 138, 139], could have been a solution to address this issue by creating artificial samples. However, since the EEG datasets themselves are very limited, this can cause overfitting [140]. Artificial trials created by GANs will not have new information on top of the available brain data captured through EEG. If the artificial trials are very similar to the existing trials, this can cause overfitting as the generative network and classification network will not generalize well enough. If the artificial trials are very different than the original samples, this will not carry any information not available the original dataset, and it might be just introducing noise.

EEG datasets are not as large as image datasets or natural language datasets. If a dataset is large enough, there can be enough data to learn the both general features and individual characteristics. Then, a generative model can create artificial trials with generalized features with variations of individual characteristics through different combinations [141]. This can help increasing the amount of data and help with training. However, EEG datasets are relatively small [82], and it is challenging to create artificial trials through GANs, and it would be a different research topic by itself.

There are examples of using GANs for EEG dataset augmentation in literature, however, published results are not convincing. For example, [142] uses GANs to increase the sample size of a motor imagery dataset which has only three electrodes. Authors mention "thousands of" epochs for training, and report that their data augmentation improved accuracy from 83% to 86%. They also use only one subject for binary classification.

Similarly, [143] uses GANs for augmenting a motor task EEG dataset. They use only one channel for one subject and they do not report the accuracies with and without data augmentation. There are other use cases for GANs to generate artificial EEG data for motor imagery, motor tasks, and emotion recognition tasks [140]. However, some of them do not report classification accuracies with and without augmented data, which makes it difficult to assess the performance of data augmentation, and some of them have only one subject or less than three channels for their experimental setups.

Creating bigger EEG datasets would be helpful but it is still costly and have other obstacles. Recording EEG signals for long periods of time is not practical. Although EEG is much more practical and costs much less than other data collection methods like fMRI, it is still not practical to collect hundreds of thousands of samples from a variety of different subjects. For this reason, when we compare the size of EEG datasets with image datasets, there is an enormous difference. For example, in computer vision, datasets like cifar-10 [144] are considered small even though it has 60000 samples. If we think about the amount of images available on the internet, where millions of users upload image data voluntarily, the amount of data available to collect is enormous and the cost is very low. On the other hand, recruiting subjects for EEG data collection, arranging laboratories and personnel to collect data, preparing subjects and placing EEG sensors on top of each subject's scalp is very costly, not that practical and consumes a lot of time.

For these reasons, it is very essential to have data augmentation methods to enhance the number of trials we have in the current and future EEG datasets. Increasing the number of trials in an EEG dataset not only can enhance the number of trials, but also can decrease the cost of recording process as it can be possible to have less recordings for each subject during the recording process. There are different methods to augment EEG data, i.e., adding noise, combining different EEG sequences, delaying an EEG sequence and combining it with different EEG sequences. However, those techniques have limitations [140].

Data augmentation for EEG signals has one important challenge. When we augment the data of different channels to create new samples, we lose the channel-wise relationships between different channels. It is important to preserve the correlations and channelwise relationships between different channels, otherwise we would change the network structure of the EEG signals.

To preserve the network structure of the EEG signals, we propose a novel graph-based EEG data augmentation technique to create augmented EEG signals while we conserve the channelwise relationships between different channels, without creating an overfitting problem. Instead of randomly combining different channels with randomly selected patches of sequences, we use graph operators to combine the channel data with the neighboring nodes.

If $\boldsymbol{S}$ is a multi-channel signal in the form of a matrix with rows representing channels,

we create an artificial sample as

$$\boldsymbol{S}^{'} = d^{k}\left((\alpha \boldsymbol{I} + \beta \boldsymbol{Q})\boldsymbol{S} + \theta \boldsymbol{\mathcal{L}} \boldsymbol{N}\right) \tag{4.1}$$

where $\boldsymbol{Q}$ is the *random walk matrix* defined as $\boldsymbol{Q} = \boldsymbol{D}^{-1}\boldsymbol{A}$, $\boldsymbol{\mathcal{L}}$ is the *symmetric normalized Laplacian* defined as $\boldsymbol{\mathcal{L}} = \boldsymbol{D}^{-1}\boldsymbol{L}\boldsymbol{D}^{-1}$, $\boldsymbol{N}$ is a white noise matrix, d is the delay operator, and k is the amount of delay. The product $\boldsymbol{QS}$ provides the weighted average of the neighboring vertices using the weights for each vertex pairs [6]. Constants α, β, θ and k are parameters to tune the augmentation with α as the weight of the channel itself, β as the weight of the average of the neighboring channels, θ as the weight of the noise. We multiply the white noise with the *symmetric normalized Laplacian* operator $\boldsymbol{\mathcal{L}}$ to amplify the noise by the average variance between the node and its neighbors. We did this to have more noise when there is more variance between neighboring nodes. We did not use the *combinatorial Laplacian* operator $\boldsymbol{L}$ as we did not want to amplify the noise by the degree of the node. For example, a node with 10 neighbors will have five times greater value compared to a node with 2 nodes if we use the Laplacian operator. On the other hand, if we use the symmetric normalized Laplacian as an operator, we will get the normalized variance between the nodes and their neighbors. This is a better metric to use to amplify the noise. Lastly, we introduce the delay operator d^{k} with k as the constant specifying the number of samples to shift. Note that d^{k} is at the outermost part of the parenthesis as the last operation, since we shift the channels and the neighboring nodes together after applying the operators to preserve the network structure. We do not shift the sequences from different channels independently. Figure 4.5 presents the creation of an artificial sample based on the pair-wise relationships of vertices of a graph.

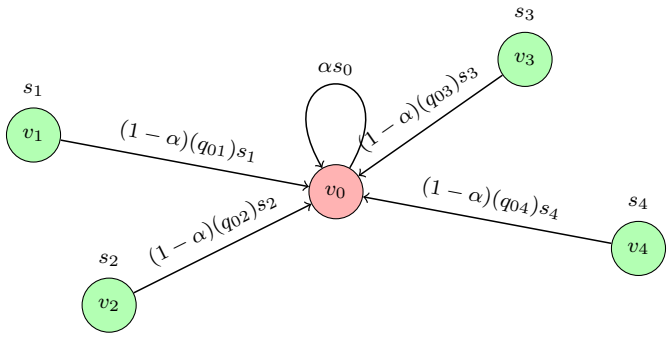


Figure 4.5: Creation of an artificial EEG signal by using graph operators as formulized in (4.1). Signal $\boldsymbol{s} = [s_0, s1, s2, s3, s4]^{\mathsf{T}}$ is shown on a graph with vertex set $\mathcal{V} = \{v_0, v_1, v_2, v_3, v_4\}$. For each vertex i, neighbor j contributes with a value of $(1-\alpha)(q_{ij})s_j$ alongside αs_i, which is a portion of the original signal value s_i, for vertex i (figure created by the author).

We want to emphasize the fact that to get the augmented version of one channel, we need to use all the channels to preserve the channelwise correlations. So, to get the

augmented version of one channel, one needs to use the multi-channel signal first to create the augmented multi-channel signal and then get the associated channel from the augmented signal. This is done to preserve the channelwise correlations. Still a subset of the multi-channel signal can be used to get the augmented channel, as long as we have the n-hop neighborhood channels, and the degree of the operators are not greater than n. So, for an augmentation where we have all the operators in first degree, we can only use the one-hop neighborhood of the channel that we are interested in and calculate the augmented version of the original signal. This assures the preservation of the relationships between neighboring channels.

4.4.1 Data Augmentation Experiment Results

Table 4.10: Intra-subject classification accuracies for short word vs long word experiment with and without data augmentation. Performance is reported for our proposed 4-cluster method with 3-layer LSTM network and attention mechanism.

Classification accuracies for short word vs long word on database [98].		
Subject	**No augmentation**	**With augmentation**
Subject A	81.50 ± 3.91%	**82.50** ± 3.35%
Subject B	89.50 ± 3.50%	**90.00** ± 3.16%
Subject C	91.25 ± 5.00%	**91.88** ± 4.00%
Subject D	93.75 ± 4.84%	**94.38** ± 4.38%
Subject E	81.50 ± 3.20%	**82.00** ± 2.45%
Subject F	94.00 ± 3.00%	94.00 ± 2.00%
Average	88.58 ± 3.91%	**89.13** ± 3.22%

Table 4.11: Intra-subject and cross-subject classification accuracies for the short word vs long word experiment with and without augmented EEG data.

Classification accuracies for short word vs long word classification.		
Task	**No augmentation**	**With augmentation**
Intra-subject classification	88.58 ± 3.91%	**89.13** ± 3.22%
Cross-subject classification	75.89 ± 2.11%	**76.52** ± 1.65%

Due to the scarcity of EEG data relevant to the taks of imagined speech analysis and recognition, we introduced artificial trials using the novel methodology presented in Section 4.4. We used the same experimental setup for imagined speech classification and compared how the classification accuracies change when we use augmented data. To generate artificial trials, we introduced $15ms$ delay with using 90% of the signal by setting $\alpha = 0.9$ and $\beta = 0.10$ to combine the signal from the proximity of each node. We used $\theta = 0.05$ to introduce white noise based on the signals within the proximity of

each node. We used the augmented data only for training and did not use it on test data to only measure the performance on the original test samples. To assess performance, we utilized the original test samples. To evaluate the impact of data augmentation, we applied the same experimental setup for imagined speech classification and compared the classification accuracies with and without using augmented data. Results are reported in Table 4.10. As seen, training data augmentation leads to a small increase in recognition performance accompanied by a decrease in variability. We also conducted cross-subject experiments using augmented data and observed a similar pattern in performance, with a small increase in recognition accuracy and a decrease in variability, as shown in Table 4.11. These experiments suggest that data augmentation is a viable strategy for addressing the problem of lack of training data.

4.5 Conclusion

To model a dynamic system like brain, where the network structure is not static and changes through time and task, using a static graph with fixed connections based on Euclidean distances is a very limiting approach.

One of our aims was to evaluate the effectiveness of dynamic-graph-based EEG representations in imagined speech classification and biometric identification. To apply the proposed method to imagined speech classification and evaluate its performance against state-of-the-art methods and conventional static graph approaches, we used the dynamic graph representation of the EEG signals, which we proposed in Chapter 3, to represent the EEG signals for the imagined speech experiments and showed that our proposed method is more effective and experiment results surpasses the current state-of-the-art methods for all our experiments except one. We also showed that our method is more robust for subject-independent classification tasks. We presented results for another experiment setup that we combined the data from different subjects for cross-subject experiments, which was not reported on the benchmark method [98]. We showed that even when we combine data from different subjects, we can still get similar results to intra-subject experiments.

To investigate the effect of different clustering and eigenvector selection strategies on the performance of dynamic graph modeling of EEG signals, we obtained the results for different setups where we investigated the effect of using different number of clusters, different neural network variations, different eigenvector selection strategies, and we also compared them with baseline methods where we do not use any graph operators.

Lastly, to investigate the impact of different neural network architectures on imagined speech classification performance, and to propose a method for generating artificial EEG data while preserving channel-wise relationships, we proposed a novel graph-based EEG data augmentation method to augment multi-channel EEG signals and increase the num-

ber of trials for training. We presented the results with and without data augmentation to show that the proposed method helps with training, increases the classification accuracy and reduces the bias. We observed that the increase in accuracy is not substantial. The main reason for this is the fact that augmented data does not provide new information that is not already available in the dataset. It increases the number of samples. This makes the training more robust and decreases the variation between different folds as we use 10-fold cross validation. Althought the increase in the classification accuracy is not substantial, less variation between different folds results as an increase in classification accuracy and a decrease in variation. This means that the augmentation is actually helping with the robustness of the system rather than providing new information.

Chapter 5

Graph Signal Processing for EEG-based Biometric Identification

5.1 Introduction

In Chapters 1 and 2, we mentioned various electroencephalography (EEG) applications. One such research area involves EEG-based biometrics, where EEG can be used to identify an individual with a unique signal generated by the subject's cognitive activity [145]. There are a variety of biometric identification and authentication methods [146, 147]. Popular biometric authentication and identification methods such as fingerprint and iris scanning have downsides, as they can be stolen and used against the subject's will or without the subject's consent. Another limitation of using fingerprint or iris scanning is the inability to verify whether the subject is alive or conscious during biometric authentication or identification [148]. EEG-based biometric systems provide greater security against identification without consent, and make it difficult for third parties to replicate the original signal [148]. This approach can be further enhanced by incorporating a variety of cognitive tasks, as each individual can perform a different imagery task while their EEG signal is acquired. In this case, the cognitive task becomes part of the biometric signature as if it is part of a passkey, which is implemented as a *"pass-thought"* in [119].

Most EEG applications use hand-crafted statistical features [18,149–151] without considering spatial relationships among EEG signals. For these reasons, it is important to develope a more robust system that extracts spatiotemporal features from EEG signals and uses the most discriminative features for biometric identification. Using our methodology for *dynamic graph representations of EEG signals*, presented in Chapter 3, we propose a novel method to incorporate both graph features and time variations of spatial frequencies to analyze the multi-channel EEG signals and use in biometric identification. As in Chapter 4, we adopt the strategy of employing eigenvectors of different graphs to represent EEG signals for identification. Our approach relies on individual differences of

projections of EEG signals onto the eigenvectors of the dynamic graph representations. We use those differences as discriminatory information in identification.

There are two main scenarios where EEG signals can be used as biometric information. One is *biometric identification* and the other is *biometric authentication*. Biometric identification is using a biometric input to identify the subject, i.e., using fingerprints in a crime scene to compare it with a database to identify the subject [112]. On the other hand, biometric authentication uses the biometric data to verify a user's identity, i.e., face recognition on smartphones, fingerprint recognition in borders to verify a person's identity [152]. In this chapter, we focus on biometric identification as an application, that means we use data from a pool of users to identify the identity of the user.

5.2 Dynamic Graph Representations of EEG Signals for Biometric Identification

In previous chapters, we presented dynamic graph representations of EEG signals and their application on imagined speech classification. Using dynamic graph for imagined speech classification was effective as it allowed us to model changes in the graph structure during imagined speech. During the imagination of speech, graph structure changes based on subject and imagined word. Dynamic graph representations address those issues by representing a graph signal using different graphs. Likewise, using EEG signals for biometric identification requires capturing the subject-dependent nature of those changes in spatiotemporal features. When EEG signals are projected onto the eigenvectors of a dynamic graph, different subjects have different representations, i.e., graph frequencies which correspond to spatial frequencies. When we analyze the temporal variations of those spatial frequencies to get the spatiotemporal representations of signals, it creates different patterns for different subjects, and they provide discriminatory information for the neural network to classify the subject's identity.

5.3 Experimental Assessment

5.3.1 Databases

Although our focus in previous chapters was imagined speech, we used the Physionet [114] dataset alongside the Kara One [66] dataset even though the Physionet is a motor imagery dataset. To assess the performance of our method, and compare it with other benchmark methods, it is essential to use benchmark datasets that other methods use. If a dataset does not have enough subjects, the quality of the recorded data, task or other properties would not have any significance since the main goal is identifying individuals rather than

Table 5.1: Experimental procedure for the PhysioNet database. The column *"Imagery"* explains which experiments use the imagery data (since we are not interested in actual physical movements). We use imagery data only where the subject merely imagines moving their limbs without making any muscle movements.

Experimental procedure during the recording for the PhysioNet database			
Run	**Task**	**Task definition**	**Imagery**
Run 1	-	Baseline, eyes open	
Run 2	-	Baseline, eyes closed	
Run 3	Task 1	Open and close left or right fist	
Run 4	Task 2	Imagine opening and closing left or right fist	Imagery
Run 5	Task 3	Open and close both fists or both feet	
Run 6	Task 4	Imagine opening and closing both fists or both feet	Imagery
Run 7	Task 1	Open and close left or right fist	
Run 8	Task 2	Imagine opening and closing left or right fist	Imagery
Run 9	Task 3	Open and close both fists or both feet	
Run 10	Task 4	Imagine opening and closing both fists or both feet	Imagery
Run 11	Task 1	Open and close left or right fist	
Run 12	Task 2	Imagine opening and closing left or right fist	Imagery
Run 13	Task 3	Open and close both fists or both feet	
Run 14	Task 4	Imagine opening and closing both fists or both feet	Imagery

the tasks undertaken. Additionally, for a biometric identification experimental setup, it is important to have a greater number of individuals. This makes the Physionet dataset a better option to use for benchmarking purposes to see how our method performs relative to other methods in literature. As listed in Table 5.3, for biometric identification tasks, [115–119] and [112] use their own datasets. This makes it difficult assess how much of the performance is coming from the dataset itself. Dataset and the recording method itself can cause an increase in performance [98]. [109] and [110] use the DEAP dataset [106] for emotion recognition, while [111, 112] and [113] uses Physionet database [114]. As the Physionet database has 109 subjects, and already used by other methods, it is a better benchmark dataset for comparing our method with other methods in literature. Also, since the Kara One dataset includes only 14 subjects, it would not be sufficient for the assessment of the performance of the method. Therefore, for the application of EEG-based biometric identification, we also include the PhysioNet dataset alongside the Kara One database as it is more suitable for testing biometric applications.

PhysioNet database includes four different tasks for each subject recorded through three different sessions. Each recording starts with two baseline runs with one of them eyes open and the other one with eyes closed. Since there are 14 different runs, four different tasks, two different baseline recordings and some recordings have actual movements and some of them have motor imagery, it becomes difficult to follow. So, we presented the details in the Table 5.1 to give a quick summary. As presented in Table 5.1, *"Run 4"*, *"Run*

8" and *"Run 12"*, which are highlighted with magenta, includes imagery data where the subject imagines opening and closing left or right fist. We combined those three sessions for one experiment for the biometric identification. It is important to emphasize that using data from different sessions makes classification tasks more difficult. In a real-life scenario, any biometric identification or authentication application should work after a period of time passed from the collection of the original recording. For example, if we had EEG data for biometric authentication, the subject should be able to verify their identity even if three months passed from the time the EEG data collected from the subject. Most of the datasets include only one session. When there is only one session with around 20 subjects, obviously any kind of classification task becomes easier, however, that might not reflect the actual performance of the system as the results will be biased.

Additionally, as presented in Table 5.1, *"Run 6"*, *"Run 10"* and *"Run 14"*, which are highlighted with cyan, includes imagery data where the subject imagines opening and closing both fists or both feet. We combined the data from those sessions for another experiment. We did not use the remaining runs as they correspond to actual physical movements. We used two different imagery tasks separately to investigate how they differ in classification accuracy.

We presented the details of the Kara One database in Section 4.3.1. As we stated before in Section 4.3.1, again we discard the video and audio data and we only use the EEG data from the 14 subjects where they imagine speaking 11 different phonemes and words.

5.3.2 Preprocessing of EEG Signals

We used the PhysioNet dataset described in Section 5.3.1. As the published data is unprocessed EEG data, we applied a notch filter to remove the 60Hz power line and later applied a 5^{th} order butterworth bandpass filter to keep the frequencies between 0.5Hz and 50Hz. We later multiplied the signal by 10^6 to change the values from mV to V which would otherwise be problematic as it would be difficult for the neural network to handle very small numbers. We did not apply any normalization to the raw data before or after the preprocessing as we wanted to get the short-time Fourier transform without changing the amplitudes of the signals.

As we described in Chapter 3, in order to prepare the DMGF for neural network classification, we offset STFT magnitude values by 1 to make the smallest magnitude equal to 1. We then apply log-normalization to keep the STFT magnitude values within a range that can be handled by the neural network. This ensures that the minimum log-magnitude is 0 while small magnitudes, such as 10^{-20}, are represented as approximately 0.

For the Kara One database, the authors shared both processed and raw EEG data.

To make sure that we use the same input signals, we used their processed signals instead of processing the raw signals to make sure the preprocessing is not a factor in difference of classification accuracy. However, as in the case of PhysioNet database, we performed the log-normalization as another preprocessing step before feeding the input into the neural network to keep the values in a range where the neural network can handle.

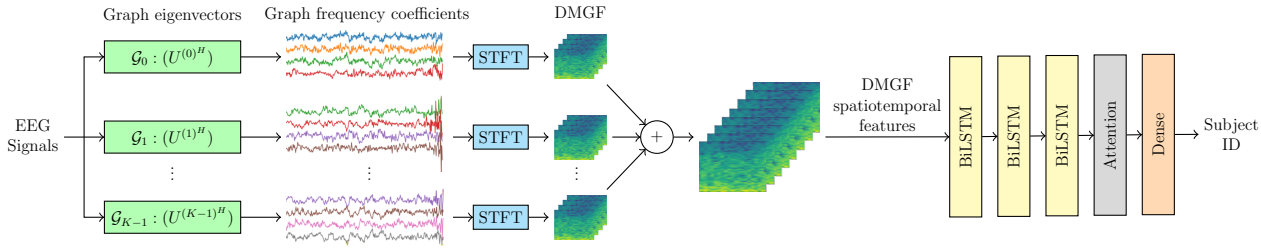


Figure 5.1: Main architecture for the recognition of the subject's identity. Multi-channel EEG signals are projected onto the eigenvectors of each graph and yield timeseries of graph (spatial) frequency coefficients. These are subjected STFT, producing the proposed DMGF representation. The resulting DMGF coefficients are fed into an attention-based BiLSTM network for the recognition of subjet's identity (figure created by the author).

5.3.3 Intra-session vs Cross Session

One important challenge when EEG signals are used for biometric identification is the variation between signals from different sessions [119], as recordings taken in different times can vary too much. However, EEG datasets generally include data from one continuous session of recordings [119]. This makes it difficult to assess whether the system is robust enough to achieve identification/authentication when the signals come from different sessions. For example, in a scenario where the user's EEG data is recorded in a visa application center, the recording should not vary too much with time for the user to provide similar EEG data at the border while visa control. In Section 5.3.1, we explained the details of the two databases used, which are PhysioNet [114] and Kara One [66]. The PhysioNet database has recordings from three different sessions. We combined data from three different sessions on PhysioNet database to take session-wise variation into account and ensure that our results are not session-specific. On the other hand, the Kara One database includes a continuous recording for each subject, so the respective results do not reflect session-wise variations.

5.3.4 Experiments

We run three experiments in total, two of them with the PhysioNet database and one of them with the Kara One database. As we stated in Section 5.3.1, we combined *"Run 4"*,

"Run 8" and *"Run 12"*, which includes imagery data of the subjects imagining opening and closing left or right fist. This part of the data is used for the first experiment, and we refer to it as *PhysioNet left vs right fist*. We combined *"Run 6"*, *"Run 10"* and *"Run 14"*, which includes imagery data of the subjects imagining opening and closing both fists or both feet. This data is used for the second experiment, and we refer to this data as *PhysioNet both fists vs both feet*. Finally, we used the Kara One database for third experiment where subjects imagine speaking 7 phonemic/syllabic prompts and 4 words.

Contrary to other variations of cross-validations, we used 10-fold nested cross validation scheme, which we detail in Section 2.4.3. We divide the data from all the users into training and test sets and we do not use test data in any way for calculating the dynamic graph representation or normalization. Instead of dividing the data into training and test sets before feeding them into neural network, we divide the data at the start of the experiment and make all the calculations for the dynamic graph representations based on the training set. This increases the amount of calculations, computational cost and amount of time the experiment takes. However, to get reliable results this approach is necessary. So, instead of calculating the dynamic graph representations once and running the machine learning part of the experiment 10 times, we repeat the entire process 10 times.

After dividing the data into training and test sets, we use all the training data for calculating the correlations between different channels and clustering the correlations to calculate the dynamic graph representations. We used two correlation centers for this application and used all the eigenvectors from both graphs. After projecting all the individuals' EEG signals onto the eigenvectors of the dynamic graph, we calculated the STFT of each graph frequency sequence, as we also did in Chapter 4. This gives the temporal variations of the spatial frequencies, which effectively provides the spatiotemporal variations of the signals on graph as we presented in Section 3.4.

We used the three-layer bidirectional LSTM network with attention for the classification presented in Figure 5.1, which was the main neural network architecture that we used in Chapter 4. We used the same architecture since we have the same DMGF features for the classification. As we mentioned in Section 4.3.5, after experimenting with different neural network architectures, we observed that we had the best classification performance with our neural network *1-channel 3-BiLSTM + attention* as shown in Figure 5.1, which is described in detail in Chapter 4. We kept the hyperparameters of the neural network the same with 256 parameters for each layer, which are doubled as they are bidirectional, and number of parameters on the last dense layer again equals to the number of classes, which becomes the number of subjects in this experimental setup. We did not use any dropout regularization as we kept the learning rate 10^{-5} and the learning curves converged smoothly for both training and test sets without making oscillations.

Table 5.2: Identity identification accuracies using two different EEG databases. PhysioNet database includes motor imagery signals, and the Kara One database includes imagined speech signals. We reached to 100% accuracy on both databases, which is on a par with the state-of-the-art methods as presented in Table 5.3.

Biometric identification accuracies using dynamic graph representation.			
Database	**Imagined task**	**Number of subjects**	**Identification accuracy**
PhysioNet [114]	Moving left or right fist	109	**100**%
PhysioNet [114]	Moving both fists or both feet	109	**99.9**%
Kara One [66]	Imagined phonemes and words	14	**100**%

Table 5.3: Comparison of identity identification accuracies of different methods.

Comparison of identification accuracies with different methods.			
Method	**Database**	**Number of subjects**	**Identification accuracy**
Palaniappan et al. (2007) [115]	Self [115]	102	98%
Abdullah et al. [116]	Self [116]	10	96%
Chuang et al. (2013) [117]	Self [117]	15	22%
Armstrong et al. (2015) [118]	Self [118]	45	97%
Ruiz-Blondet et al. (2016) [119]	Self [119]	50	**100**%
Wang et al. (2020) [112]	Self [112]	59	92.05%
Khalil et al. (2017) [109]	DEAP [106]	20	75%
Cai et al. (2023) [110]	DEAP [106]	31	99.21%
Seyfizadeh et al. (2024) [113]	[153]	10	99.33%
Seyfizadeh et al. (2024) [113]	PhysioNet [114]	109	99.73%
La Rocca et al. (2014) [111]	PhysioNet [114]	109	**100**%
Wang et al. (2020) [112]	PhysioNet [114]	109	99.65%
Proposed method	PhysioNet [114]	109	**100**%
Proposed method	Kara One [66]	14	**100**%

5.3.5 Results

We used our proposed method on PhysioNet dataset, described in Section 5.3.1, and also on the Kara One [66] database, described in Section 5.3.1. In Table 5.2, we present our results on PhysioNet and Kara One databases, and in Table 5.3, we present our results alongside other methods in literature for comparison. The Kara One database has 14 subjects and PhysioNet dataset has 109 subjects. Several works in literature use fewer than 25 subjects in their experiments. This number is very small for the reliable assessment of the performance of any method. For this reason, the identification accuracy in the PhysioNet dataset yields more reliable evaluation of the performance of the method. Our methodology reached 100% accuracy on both Kara One and PhysioNet datasets for the biometric identification tasks, which is on a par with the state-of-the-art methods.

5.4 Conclusion

One of the aims of the thesis was to evaluate the effectiveness of dynamic-graph-based EEG representations in biometric identification. Regarding this aim, our objectives was to evaluate whether individual differences in dynamic graph representations can be used for biometric identification in imagined speech tasks, and to compare the performance against state-of-the-art methods and conventional static graph approaches. Using the dynamic graph representation of EEG signals that we presented in Chapter 3, we reached to 100% accuracy of biometric identification of subjects. This includes two different EEG datasets with one of them including 109 subjects and their recordings from three different sessions. We reached the identification accuracy of the state-of-the-art methods and proved that different eigenvectors of the dynamic graph representation can be used for biometric identification.

However, to use this technology in real life scenarios like in airport borders, we need to make it reliable to work with less than 1 second of input. Also, we use all the data for the training, as of other methods present in the literature. Although current state-of-the-art methods include all the data from subjects for training, and our method also reaches up to 100% accuracy with the same strategy, in a real-life scenario it would not be possible to capture long sessions of EEG recordings from subjects for EEG based biometric identification. Just like collecting fingerprints of a subject, EEG-based biometric identification methods should work with a very short sequence of signals. Further, another important aspect is to be able to make prediction with a limited input, i.e., a partial fingerprint for biometric identification. To be able to use EEG-based biometrics, the system should also work with partial input data.

As future work, we can run variations of these experiments with different number of clusters and different eigenvector selection strategies as we did in Chapter 4 to analyze which eigenvectors are more discriminatory. We also need to assess those results with a database that includes siblings, family members and twins [154], so that we can assess whether the eigenvectors we use for discrimination can work on identifying twins and family members, and establish how the projections on eigenvectors change for twins.

Another aspect that needs to be investigated is the long-term performance of different eigenvector selection strategies. For example, a selection of eigenvectors from different clusters, i.e., lowest eight spatial frequencies from four-cluster dynamic graph, can work very efficiently and give 100% accuracy. However, this accuracy can change if we use data from different sessions that are three months apart. To check the robustness of the system, we need to have EEG datasets with different sessions with varying time intervals, i.e., one day, one week, one month, three months. This can help us investigate the robustness of different eigenvector selection strategies for biometric identification over a period of time. Those are very crucial questions that need to be answered in the future.

Chapter 6

Conclusions

6.1 Findings of the Thesis

Throughout this work, we investigated how to represent the dynamic nature of the EEG signals by considering that the network structure is not static. Correlations and connections between different vertices can change over time and across activities, and exhibit varying characteristic traits for different individuals. We conducted a systematic investigation to answer the research questions we listed in Section 1.6. For this investigation, we had two important aims as we described in Section 1.7, and to reach these aims we had six objectives as we described in Section 1.8. This thesis contributes to the literature by proposing a novel dynamic graph representation framework for EEG signals, utilizing a unified representation of spatial and temporal frequency features, and demonstrating applicability in imagined speech recognition and biometric identification tasks. This thesis also proposes a novel method to produce artificial EEG trials without altering the channel-wise correlations and preserving the correlations between different sensor pairs without altering the network structure.

The first aim was to develop a dynamic-graph-based modeling strategy for EEG signals that captures evolving spatial and temporal relationships among different brain regions. This aim corresponded to research questions one and two, and was addressed through objectives one and two.

The first objective was to propose a method for constructing dynamic graph representations of EEG data that model the time-varying, task-specific, and subject-specific relationships among different brain regions, independent of their physical proximity. In Chapter 3, we proposed dynamic graph representations of EEG signals to address the limitations of using static graphs for representing EEG signals. We provided a framework based on the eigenvectors of different graphs instead of a single static graph. This allows the signal to be projected onto different spatial frequency components derived from a variety of graphs. This addresses the problem of having a limited set of graph frequency

components for different applications, without considering the change in network structure and correlations between different brain regions across tasks. This answered the first research question of how the dynamic nature of EEG data can be modeled using graph-based representations that capture temporal and task-related variations in the relationships among different brain regions, and established the theoretical foundation for the applications presented in the following chapters.

The second objective was to extract and analyze spatiotemporal frequency features from the constructed dynamic graph representations of EEG signals. In Chapter 3, in Section 3.4, we provided our proposed spatiotemporal representation of graph signals. We addressed the limitation of traditional representations that provide spatial frequency components only at individual time instants, and we proposed a framework to represent spatial and temporal frequencies in a unified way. This framework provides the temporal frequency components of each spatial frequency component, representing how different spatial frequency components evolve over time. This provided the answer to the second research question regarding how spatial and temporal variations in EEG signals can be integrated to extract spatiotemporal frequency features from graph-based EEG representations.

The second aim was to evaluate the effectiveness of dynamic-graph-based EEG representations in imagined speech classification and biometric identification. This aim addressed research questions three, four, five, and six, as listed in Section 1.6. Objectives three through six were defined to achieve this aim and provide answers to those research questions, as described in Section 1.8. To meet these objectives, we conducted extensive experiments in Chapter 4 for the third, fourth, and fifth objectives, and in Chapter 5 for the sixth objective. These experiments demonstrated the effectiveness of the proposed approach in classification and identification tasks, as discussed in detail below.

The third objective was to apply the proposed method to imagined speech classification and evaluate its performance against state-of-the-art methods and conventional static graph approaches. In Chapter 4, we investigated the application of dynamic graph representations of EEG signals to imagined speech recognition by classifying different words imagined by subjects. We showed that our method surpasses current state-of-the-art approaches in most cases. We compared our results with both graph-based methods and other methods using the same datasets. This demonstrated the effectiveness of dynamic graph modeling in imagined speech classification and provided the answer to the third research question regarding whether dynamic-graph-based EEG representations enhance imagined speech classification performance in comparison to static graph models and current state-of-the-art methods.

The fourth objective was to investigate the effect of different clustering and eigenvector selection strategies on the performance of dynamic graph modeling of EEG signals. In Section 4.3.3 and Section 4.3.4, we conducted a thorough investigation using different

clustering and eigenvector selection scenarios to examine the effects of these strategies and to identify the most effective clustering and eigenvector selection strategies. We showed that using multiple clusters helps with classification accuracy and obtained the best results when using four clusters. We conducted experiments with different numbers and subsets of eigenvectors and presented the results of our investigation in detail. We found that using a subset of eigenvectors can still provide comparable results. Low spatial frequencies generally provided better classification accuracy than high spatial frequencies, although the difference was not substantial. We achieved the best classification results when using all the eigenvectors together, which suggests that the relevant information is distributed across different spatial frequency components. This provided the answer to the fourth research question regarding how clustering strategies and eigenvector selection methods affect the performance of dynamic-graph-based EEG representations in imagined speech classification tasks.

The fifth objective was to investigate the impact of different neural network architectures on imagined speech classification performance, and to propose a method for generating artificial EEG data while preserving channel-wise relationships. In Section 4.2, we used different neural network architectures to examine their effect on classification performance. We concluded that using both convolutional neural networks and long short-term memory networks yielded comparable results, although LSTMs were found to be more suitable. We therefore used a 1-channel 3-layer bidirectional LSTM network with attention for the rest of the experiments after conducting different experiments to compare the performance of different neural network architectures. We used the attention mechanism as it helps with selecting more relevant features for the classification task, allowing the neural network to focus on those features instead of factoring in all the information equally. At the end of Chapter 4, in Section 4.4, we proposed a graph-based EEG data augmentation method to create augmented EEG data by preserving the channel-wise relationships of the EEG signals. We provided different parameters to tune the augmentation process, and we presented experiment results with the augmented EEG data. We showed that data augmentation led to a modest increase in classification accuracy. We also showed that the variance between different folds decreased, so that the variation between results decreased with the augmented data, which means the training and testing phases are more robust. We observed that the augmented data improves the robustness of the training phase, as the number of samples in EEG datasets is limited and this increases the variance between different folds for validation and makes the training harder for the neural network. Augmented data addresses this issue to increase the number of samples and make the training more robust. However, we also observed that this does not result in a substantial increase in classification accuracy. We concluded that the information in the data is limited and augmentation does not address that issue but helps provide enough information between different folds to increase the robustness of the

training which results in a lower variation and higher classification. This provided the answer to the fifth research question regarding how different neural network architectures affect imagined speech classification performance, and how artificial EEG data can be generated while preserving channel-wise relationships among different brain regions.

The sixth objective was to evaluate whether individual differences in dynamic graph representations can be used for biometric identification in imagined speech tasks, and to compare the performance against state-of-the-art methods and conventional static graph approaches. In Chapter 5, we investigated whether the individual differences between the dynamic graph representations of EEG signals can be used to discriminate different subjects. We applied our proposed method for the biometric identification of subjects by using two different EEG databases where one of them has 109 subjects and recordings from three different sessions. We achieved 100% identification accuracy from the classification experiments that took place. We did not run the experiments for authentication. However, it is important to note that authentication is an easier task than identification. For authentication, if we ignore the weights/number of different subjects, we have a 50% chance of giving the correct answer, since the task is to either verify the subject's identity or deny access. So, in that scenario, we have a one-versus-all case and the output is binary. On the other hand, for the identification task, the chance level accuracy is $1/N$ where N is the number of subjects. For this reason, we did not run any experiments for the authentication case as it is a much simpler case. Ultimately, we showed that dynamic graph representations of EEG signals contain discriminative subject-specific features for biometric identification. When we reflect the EEG signals onto the eigenvectors of the dynamic graph, different subjects produce distinct reflections. Those differences not only provide discriminatory information to classify what the subjects think, but also discriminatory enough to identify the identity of the subjects. We compared our results with other methods in the literature and showed that our method is on a par with the current state-of-the-art methods. This provided the answer to the sixth research question, whether individual differences in dynamic-graph-based EEG representations can be used for biometric identification in imagined speech tasks.

We concluded with a comparison and critique of current state-of-the-art methods and our proposed method, highlighting why these approaches are not yet feasible for deployment in practical biometric systems. Our method, like other approaches in the literature, requires the use of all the EEG data for training and testing. This requirement limits its applicability in a real-life scenario. In contrast to other methods like fingerprint scanning, EEG acquisition typically takes at least half an hour, and there is also considerable variance between different recording sessions. We conducted a similar experimental setup and achieved 100% classification accuracy, which is on par with current state-of-the-art methods. However, this level of accuracy does not imply that the approach is suitable for practical applications such as visa control or biometric identification systems. For a de-

ployment scenario to be feasible, the amount of data needed to train the neural network must be substantially reduced. Also, it is important to consider that it is possible to use a partial fingerprint to identify a subject. Therefore, for the training phase, to have a comparable system to the current state-of-the-art approaches, it is acceptable to use the whole EEG sequences for training. However, to have a comparable system to other biometric identification methods like fingerprint-based identification, the system should also be able to identify the subjects based on partial EEG recordings. All these details are important factors to make the EEG-based identification and authentication viable in practical settings.

6.2 Open Issues and Future Work

Unlike the graph theory itself, Graph Signal Processing (GSP) has a relatively short history and remains an emerging area. There are still some basic concepts that do not have definitive answers in GSP. For example, there are multiple definitions of frequency, which actually should be very straightforward, and they do not fit into our intuition from the conventional signal processing (SP). However, it still addresses several limitations of conventional signal processing even though it has its own shortcomings. Some works in the literature take those concepts for granted, as if the Laplacian-based frequency is the only way to define frequency and they consider it as if it is a well-established area as if all meaningful developments have already been achieved, which is not accurate. Graphs are not unique, and different applications and different data types require different modeling strategies. Depending on the task and application, we can create different graph models to represent a network. Since the sensor placement system is standardized, this modeling strategy results in nearly identical graphs across subjects, as sensor locations on top of their scalp will just vary slightly based on the size and shape of each subject's head. However, different people can have different connections between different regions of their brain and those relationships do not necessarily depend on their contiguity.

GSP provides various interesting and beautiful representations that can be elegant in some cases and chaotic in others. If we consider a ring graph and use the eigenvectors of the adjacency matrix, we can derive the whole conventional discrete signal processing formulas from scratch and we can see the conventional signal processing as a subset of GSP. In that case, the frequency definition also matches conventional SP definition because the ring graph becomes the time sequence. On the other hand, it makes life very complicated if we use adjacency-based frequency for other graphs as dealing with complex eigenvalues and eigenvectors will not be as simple as dealing with the SP formulas, frequencies would not be ordered, and many other inconveniences would be waiting for us. Even this by itself is an issue as almost all the research in GSP applications is based on symmetric graphs with Laplacian-based frequency definitions just to limit the calculations to real

values. This does not completely refer to the complex nature of the data we deal with. Hopefully, in the next decade, there will be much more interesting use cases of GSP and some other mathematical developments to address some of the shortcomings of the theory.

As we stated before, academic fields like *network science* and *network processes* focus on and analyze the graphs themselves, not the signals on graphs. GSP focuses on processing the signals represented on graphs and aims to carry the knowledge and intuition from the well-established domain of SP to process the signals defined on graphs. As a downside, GSP analyzes the signals at a time instant, not through different time-steps.

EEG signal processing and EEG based brain computer interface (BCI) applications will be popular in the next decades. Right now, smartphones are the mainstream devices that everybody uses around the world, but new BCI applications with wearable devices can be mainstream alongside smartphones or they can take their place altogether in the future. To have a mainstream, low-cost, intuitive BCI application, the interface should not be invasive. Asking every user to have medical operations to cut their skull and place an electronic device on top of their brain cortex or within the cortical tissue is not a feasible strategy. For these reasons, EEG can be a reliable, non-invasive alternative. Nevertheless, there remain many challenges to overcome for using EEG for these applications. The noise that is captured while having EEG recording is an issue. Also, the amount of data and the resolution of data need to increase. We need big datasets with more participants, recorded through different and longer sessions, including family members and twins as participants. In addition, further mathematical theory must be developed to better analyze the EEG recordings. When we get the EEG recordings, the signals related to basal metabolism, all the emotions of the participant, endocrinological components, sensory mechanisms, and numerous other physiological and psychological factors are included. We cannot simply record only the components that we are interested in. Methods like independent component analysis are too simple to address those issues. On the other hand, all those issues mean that there is a lot of room for improvement.

One of the main issues in using EEG for biometrics is the lack of appropriate datasets. To analyze which traits/features of the signals are feasible to be used for biometric identification and authentication applications, we need datasets that include twins and other family members. We need to investigate how the classification accuracies change among the family members. A particular application may discriminate unrelated individuals but might fail when twins, siblings, or other family members are present in the dataset. On the other hand, a setup to discriminate the relatives might fail in discriminating a large pool of users including non-relatives.

The number of different sessions is also an important subject in dealing with EEG data. To use EEG data for biometric identification and authentication, just like in many other EEG applications, we need to have a robust system to work after a period of time.

The signals created by users can vary after a period of time. If we have a recording from a user for biometric authentication, it should be able to verify the identity of the user after a few weeks or months. If the signal varies too much and the system cannot verify the identity, it would not be a reliable method to be used for biometric authentication. Having data from different sessions from different days, weeks and months can help us investigate those issues. On the other hand, if we modify the system to make it more robust for dealing with signals from different times, we need to check how this changes the accuracy of the system. A modification to the system to make it work after a long period of time can decrease the accuracy and it may grant access to unauthorized users when it should not.

All those issues depend on each other and there can be tradeoffs between the robustness of different aspects of the system that we mentioned above. For these reasons we need more complex datasets to investigate all those issues together. Currently available datasets are very limited. Most EEG datasets include only one session, and they do not have twins and other family members as different subjects. Also, most of them have less than 20 subjects, which is not a reliable way to test the robustness of an application. To have large datasets with the properties that we mentioned above, there needs to be investment, and those datasets should be public, so that researchers from different institutions can use those datasets to validate and compare their methods.

Currently, collecting EEG data as a PhD project would not be a feasible way to address those issues we mentioned above. An EEG dataset to address those issues would require much more workforce, time, participants and funding. Collecting EEG data independently also has disadvantages. The performance of a particular application may result from the dataset itself. To compare and validate different methods proposed by different researchers, EEG datasets must be public and available to those researchers, so that we can use them as benchmarks.

Furthermore, the variance of EEG signals based on the effect of emotional states is another issue. Currently, EEG applications about emotion recognition are based on binary classification of emotional states. Differentiating emotional states of a user in binary classification settings is already a difficult task, and it becomes even harder to do more complex classification tasks when the emotions vary. We need to consider how the performance of EEG-based biometric identification and authentication methods vary based on the emotional state of the users. Emotional states will affect the brain state and hence the EEG signals recorded from the subject. Therefore, the signals used for identification and authentication processes can vary too much. It is important to have a system that is robust against the variations of signals based on the change of users' emotional states, otherwise the system would not be reliable to be used in the first place. Users may be denied access to a system if they experience intense emotional states. For example, in a time-sensitive scenario where a user is under threat and needs immediate

access to their home or their car, the system may deny access because the user is frightened and had an adrenaline rush. Such failures would be a big problem and undermine the reliability of EEG-based biometric authentication systems. Therefore, it is essential to ensure that EEG-based identification and authentication systems can work reliably when the users are under stress or when they have different emotional states.

Numerous open issues remain in GSP, graph representations of EEG signals, EEG dataset development and EEG-based biometric identification and authentication. These areas are expected to remain active research fields in the coming decades. Continued contributions to their advancement will be of significant scientific and practical importance.

Bibliography

[1] A. Sandryhaila and J. M. Moura, "Big Data Analysis with Signal Processing on Graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sep. 2014.

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[3] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, "Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges," May 2021.

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[5] A. E. Bickle, *Fundamentals of Graph Theory*, ser. Pure and Applied Undergraduate Texts. Providence, Rhode Island: American Mathematical Society, 2020, no. 43.

[6] A. Ortega, *Introduction to Graph Signal Processing*, 1st ed. Cambridge University Press, Jun. 2022.

[7] A.-L. Barabási, "Network science," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1987, p. 20120375, Mar. 2013.

[8] M. Newman, *Networks*. Oxford University Press, Oct. 2018, vol. 1.

[9] J. M. F. Moura, "Chapter 8 - Graph Signal Processing," in *Cooperative and Graph Signal Processing*, P. M. Djurić and C. Richard, Eds. Academic Press, Jan. 2018, pp. 239–259.

[10] A. Ortega, P. Frossard, J. Kovacevic, J. M. F. Moura, and P. Vandergheynst, "Graph Signal Processing: Overview, Challenges, and Applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[11] G. Leus, A. G. Marques, J. M. Moura, A. Ortega, and D. I. Shuman, "Graph Signal Processing: History, development, impact, and outlook," *IEEE Signal Processing Magazine*, vol. 40, no. 4, pp. 49–60, Jun. 2023.

[12] M. Gardner, *Martin Gardner's Sixth Book of Mathematical Diversions from "Scientific American"*. Chicago: Univ of Chicago Pr, Jan. 1984.

[13] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[14] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[15] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs: Frequency Analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[16] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. Grave De Peralta, "EEG source imaging," *Clinical Neurophysiology*, vol. 115, no. 10, pp. 2195–2222, Oct. 2004.

[17] T. Kirschstein and R. Köhling, "What is the Source of the EEG?" *Clinical EEG and Neuroscience*, vol. 40, no. 3, pp. 146–149, Jul. 2009.

[18] A. Saibene, M. Caglioni, S. Corchs, and F. Gasparini, "EEG-Based BCIs on Motor Imagery Paradigm Using Wearable Technologies: A Systematic Review," *Sensors*, vol. 23, no. 5, p. 2798, Mar. 2023.

[19] Y. Rezaei Tabar and U. Halici, "Brain Computer Interfaces for Silent Speech," *European Review*, vol. 25, no. 2, pp. 208–230, May 2017.

[20] R. Abiri, S. Borhani, E. W. Sellers, Y. Jiang, and X. Zhao, "A comprehensive review of EEG-based brain–computer interface paradigms," *Journal of Neural Engineering*, vol. 16, no. 1, p. 011001, Feb. 2019.

[21] A. Khosla, P. Khandnor, and T. Chand, "A comparative analysis of signal processing and classification methods for different applications based on EEG signals," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 2, pp. 649–690, Apr. 2020.

[22] W. Lee, J. J. Seong, B. Ozlu, B. S. Shim, A. Marakhimov, and S. Lee, "Biosignal Sensors and Deep Learning-Based Speech Recognition: A Review," *Sensors*, vol. 21, no. 4, p. 1399, Feb. 2021.

[23] X. Gu, Z. Cao, A. Jolfaei, P. Xu, D. Wu, T.-P. Jung, and C.-T. Lin, "EEG-Based Brain-Computer Interfaces (BCIs): A Survey of Recent Studies on Signal Sensing Technologies and Computational Intelligence Approaches and Their Applications," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 5, pp. 1645–1666, Sep. 2021.

[24] N. Fitriah, H. Zakaria, and T. L. E. Rajab, "EEG-Based Silent Speech Interface and its Challenges: A Survey," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 11, 2022.

[25] S. Mortaheb, J. Annen, C. Chatelle, H. Cassol, G. Martens, A. Thibaut, O. Gosseries, and S. Laureys, "A Graph Signal Processing Approach to Study High Density EEG Signals in Patients with Disorders of Consciousness," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Berlin, Germany: IEEE, Jul. 2019, pp. 4549–4553.

[26] K. Georgiadis, N. Laskaris, S. Nikolopoulos, and I. Kompatsiaris, "Connectivity steered graph Fourier transform for motor imagery BCI decoding," *Journal of Neural Engineering*, vol. 16, no. 5, p. 056021, Aug. 2019.

[27] A. Einizade, M. Mozafari, M. Rezaei-Dastjerdehei, E. Aghdaei, A. M. Mijani, and S. Hajipour Sardouie, "Detecting ADHD children based on EEG signals using Graph Signal Processing techniques," in *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, Nov. 2020, pp. 264–270.

[28] S. Sartipi, M. Torkamani-Azar, and M. Cetin, "EEG Emotion Recognition via Graph-based Spatio-Temporal Attention Neural Networks," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. Mexico: IEEE, Nov. 2021, pp. 571–574.

[29] A. Einizade, M. Mozafari, S. Jalilpour, S. Bagheri, and S. Hajipour Sardouie, "Neural decoding of imagined speech from EEG signals using the fusion of graph signal processing and graph learning techniques," *Neuroscience Informatics*, vol. 2, no. 3, p. 100091, Sep. 2022.

[30] S. Sartipi, M. Torkamani-Azar, and M. Cetin, "A Hybrid End-to-End Spatio-Temporal Attention Neural Network with Graph-Smooth Signals for EEG Emotion Recognition," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2023.

[31] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*. Pearson Education, Limited, 1997.

[32] P. Prandoni and M. Vetterli, *Signal Processing for Communications*, ser. Communication and Information Sciences. Lausanne (Suisse) Boca Raton (États-Unis): EPFL Press CRC Press, 2008.

[33] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge University Press, Sep. 2014.

[34] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, Mar. 2016.

[35] L. Stanković and E. Sejdić, Eds., *Vertex-Frequency Analysis of Graph Signals*, ser. Signals and Communication Technology. Cham: Springer International Publishing, 2019.

[36] Euclid, *The Thirteen Books of Euclid's Elements*. Courier Corporation, 1956.

[37] H. Coxeter, *Non-Euclidean Geometry*, ser. Heritage. University of Toronto Press, Dec. 1965.

[38] D. Schattschneider and M. Emmer, Eds., *M.C. Escher's Legacy*. Berlin, Heidelberg: Springer, 2003.

[39] H. S. M. Coxeter, "The Trigonometry of Escher's Woodcut Circle Limit III," in *M.C. Escher's Legacy: A Centennial Celebration*, D. Schattschneider and M. Emmer, Eds. Berlin, Heidelberg: Springer, 2003, pp. 297–304.

[40] J. Rus, "An octant of the sphere, a spherical triangle with three right angles, in orthographic projection https://commons.wikimedia.org/wiki/File:Octant_of_a_sphere.png," Apr. 2025.

[41] W. J. Levelt, "Models of word production," *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 223–232, Jun. 1999.

[42] H. Ackermann, "Cerebellar contributions to speech production and speech perception: Psycholinguistic and neurobiological perspectives," *Trends in Neurosciences*, vol. 31, no. 6, pp. 265–272, Jun. 2008.

[43] D. Dash, P. Ferrari, and J. Wang, "Decoding Imagined and Spoken Phrases From Non-invasive Neural (MEG) Signals," *Frontiers in Neuroscience*, vol. 14, p. 290, Apr. 2020.

[44] H. H. Jasper, "Ten-twenty electrode system of the international federation," *Electroencephalography and Clinical Neurophysiology*, vol. 10, pp. 371–375, 1958.

[45] M. R. Nuwer, G. Comi, R. Emerson, A. Fuglsang-Frederiksen, J.-M. Guérit, H. Hinrichs, A. Ikeda, F. Jose C. Luccas, and P. Rappelsburger, "IFCN standards for digital recording of clinical EEG," *Electroencephalography and Clinical Neurophysiology*, vol. 106, no. 3, pp. 259–261, Mar. 1998.

[46] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph Signal Processing for Machine Learning: A Review and New Perspectives," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117–127, Nov. 2020.

[47] C. Bodnar, "Topological Deep Learning: Graphs, Complexes, Sheaves," Ph.D. dissertation, Apollo - University of Cambridge Repository, Dec. 2022.

[48] G. Strang, *Linear Algebra and Learning from Data*. Wellesley: Wellesley-Cambridge Press, Jan. 2019.

[49] L. Stankovic, D. Mandic, M. Dakovic, M. Brajovic, B. Scalzo, and T. Constantinides, "Graph Signal Processing – Part I: Graphs, Graph Spectra, and Spectral Clustering," Aug. 2019.

[50] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph fourier transform," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada: IEEE, May 2013, pp. 6167–6170.

[51] L. Stankovic, D. Mandic, M. Dakovic, M. Brajovic, B. Scalzo, and A. G. Constantinides, "Graph Signal Processing – Part II: Processing and Analyzing Signals on Graphs," Sep. 2019.

[52] L. Stanković, D. Mandic, M. Daković, B. Scalzo, M. Brajović, E. Sejdić, and A. G. Constantinides, "Vertex-frequency graph signal processing: A comprehensive review," *Digital Signal Processing*, vol. 107, p. 102802, Dec. 2020.

[53] L. Stankovic, D. P. Mandic, M. Dakovic, I. Kisil, E. Sejdic, and A. G. Constantinides, "Understanding the Basis of Graph Signal Processing via an Intuitive Example-Driven Approach [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 133–145, Nov. 2019.

[54] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada: IEEE, May 2013, pp. 6163–6166.

[55] A. Biasiucci, B. Franceschiello, and M. M. Murray, "Electroencephalography," *Current Biology*, vol. 29, no. 3, pp. R80–R85, Feb. 2019.

[56] A. Kawala-Sterniuk, N. Browarska, A. Al-Bakri, M. Pelc, J. Zygarlicki, M. Sidikova, R. Martinek, and E. J. Gorzelanczyk, "Summary of over Fifty Years with Brain-Computer Interfaces—A Review," *Brain Sciences*, vol. 11, no. 1, p. 43, Jan. 2021.

[57] B. Graimann, B. Z. Allison, and G. Pfurtscheller, *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction*. Springer Science & Business Media, Oct. 2010.

[58] P. Amaral, F. Sales, J. P. Cunha, P. Dias, and J. M. Fernandes, "Multimodal application for visualization and manipulation of Electrocorticography data," Jan. 2007.

[59] L. Luo, *Principles of Neurobiology*, second edition ed. Boca Raton, FL: CRC Press, Taylor & Francis Group, 2020.

[60] J. G. Betts, K. Young, J. Wise, E. Johnson, B. Poe, D. Kruse, O. Korol, J. Johnson, M. Womble, and P. DeSaix, *Anatomy and Physiology 2e*, Oct. 2022.

[61] S. Marchesotti, M. Bassolino, A. Serino, H. Bleuler, and O. Blanke, "Quantifying the role of motor imagery in brain-machine interfaces," *Scientific Reports*, vol. 6, no. 1, p. 24076, Jul. 2016.

[62] Encyclopædia Britannica, "Functional areas of the human brain https://www.britannica.com/science/Broca-area#/media/1/135877/100577," Sep. 2024.

[63] I. C. Semanjski, "Chapter 5 - Data analytics," in *Smart Urban Mobility*. Elsevier, Jan. 2023, pp. 121–170.

[64] D. Lord, X. Qin, and S. R. Geedipally, "Chapter 5 - Exploratory analyses of safety data," in *Highway Safety Analytics and Modeling*, D. Lord, X. Qin, and S. R. Geedipally, Eds. Elsevier, Jan. 2021, pp. 135–177.

[65] G. Zhang, V. Davoodnia, A. Sepas-Moghaddam, Y. Zhang, and A. Etemad, "Classification of Hand Movements From EEG Using a Deep Attention-Based LSTM Network," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3113–3122, Mar. 2020.

[66] S. Zhao and F. Rudzicz, "Classifying phonological categories in imagined and articulated speech," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 992–996.

[67] A. Delorme and S. Makeig, "EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9–21, Mar. 2004.

[68] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[69] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Global Edition*. Harlow, UNITED KINGDOM: Pearson Education, Limited, 2021.

[70] A. Al-Saegh, S. A. Dawwd, and J. M. Abdul-Jabbar, "Deep learning for motor imagery EEG-based classification: A review," *Biomedical Signal Processing and Control*, vol. 63, p. 102172, Jan. 2021.

[71] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *Journal of Neural Engineering*, vol. 16, no. 5, p. 051001, Oct. 2019.

[72] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *Journal of Neural Engineering*, vol. 16, no. 3, p. 031001, Apr. 2019.

[73] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[74] Y. Le Cun, L. Jackel, B. Boser, J. Denker, H. Graf, I. Guyon, D. Henderson, R. Howard, and W. Hubbard, "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, Nov. 1989.

[75] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[76] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," Sep. 2015.

[77] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[78] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," Apr. 2016.

[79] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," 2014.

[80] M. Murgia, "Transformers: The Google scientists who pioneered an AI revolution," *Financial Times*, Jul. 2023.

[81] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," 2013.

[82] J. Tang, J. Chen, X. Xu, A. Liu, and X. Chen, "Imagined Speech Reconstruction From Neural Signals—An Overview of Sources and Methods," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–21, 2024.

[83] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, no. 1, pp. 1–8, Dec. 2006.

[84] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," *Journal of Cheminformatics*, vol. 6, no. 1, pp. 1–15, Dec. 2014.

[85] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," *PLOS ONE*, vol. 14, no. 11, p. e0224365, Nov. 2019.

[86] H. Yang, J. Han, and K. Min, "A Multi-Column CNN Model for Emotion Recognition from EEG Signals," *Sensors*, vol. 19, no. 21, p. 4736, Oct. 2019.

[87] S. S. Saboksayr, G. Mateos, and M. Cetin, "EEG-Based Emotion Classification Using Graph Signal Processing," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 1065–1069.

[88] P. Mathur and V. K. Chakka, "Graph Signal Processing Based Cross-Subject Mental Task Classification Using Multi-Channel EEG Signals," *IEEE Sensors Journal*, vol. 22, no. 8, pp. 7971–7978, Apr. 2022.

[89] M. Miri, V. Abootalebi, and H. Behjat, "Enhanced Motor Imagery-Based Eeg Classification Using A Discriminative Graph Fourier Subspace," in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, Mar. 2022, pp. 1–5.

[90] S. Itani and D. Thanou, "A Graph Signal Processing Framework for the Classification of Temporal Brain Data," in *2020 28th European Signal Processing Conference (EUSIPCO)*. Amsterdam, Netherlands: IEEE, Jan. 2021, pp. 1180–1184.

[91] M. Sun, E. Isufi, N. M. de Groot, and R. C. Hendriks, "Graph-time spectral analysis for atrial fibrillation," *Biomedical Signal Processing and Control*, vol. 59, p. 101915, May 2020.

[92] I. Kavasidis, S. Palazzo, C. Spampinato, D. Giordano, and M. Shah, "*Brain2Image*: Converting Brain Signals into Images," in *Proceedings of the 25th ACM International Conference on Multimedia*. Mountain View California USA: ACM, Oct. 2017, pp. 1809–1817.

[93] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, and M. Shah, "Deep Learning Human Mind for Automated Visual Classification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, Jul. 2017, pp. 4503–4511.

[94] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014.

[95] S. Alzahrani, H. Banjar, and R. Mirza, "Systematic Review of EEG-Based Imagined Speech Classification Methods," *Sensors*, vol. 24, no. 24, p. 8168, Dec. 2024.

[96] N. Rahman, D. M. Khan, K. Masroor, M. Arshad, A. Rafiq, and S. M. Fahim, "Advances in brain-computer interface for decoding speech imagery from EEG signals: A systematic review," *Cognitive Neurodynamics*, vol. 18, no. 6, pp. 3565–3583, Dec. 2024.

[97] H. M. V. and B. S. Begum, "Towards imagined speech: Identification of brain states from EEG signals for BCI-based communication systems," *Behavioural Brain Research*, vol. 477, p. 115295, Feb. 2025.

[98] C. H. Nguyen, G. K. Karavas, and P. Artemiadis, "Inferring imagined speech using EEG signals: A new approach using Riemannian manifold features," *Journal of Neural Engineering*, vol. 15, no. 1, p. 016002, Dec. 2017.

[99] P. Sun and J. Qin, "Neural networks based EEG-Speech Models," Mar. 2017.

[100] P. Saha, S. Fels, and M. Abdul-Mageed, "Deep Learning the EEG Manifold for Phonological Categorization from Active Thoughts," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 2762–2766.

[101] M. A. Bakhshali, M. Khademi, A. Ebrahimi-Moghadam, and S. Moghimi, "EEG signal classification of imagined speech based on Riemannian distance of correntropy spectral density," *Biomedical Signal Processing and Control*, vol. 59, p. 101899, May 2020.

[102] S. Datta and N. V. Boulgouris, "Recognition of grammatical class of imagined words from EEG signals using convolutional neural network," *Neurocomputing*, vol. 465, pp. 301–309, Nov. 2021.

[103] M. Jiménez-Guarneros and P. Gómez-Gil, "Standardization-refinement domain adaptation method for cross-subject EEG-based classification in imagined speech recognition," *Pattern Recognition Letters*, vol. 141, pp. 54–60, Jan. 2021.

[104] A. Mohan and R. Anand, "Classification of Imagined Speech Signals Using Functional Connectivity Graphs and Machine Learning Models." Aug. 2024.

[105] Y. F. Alharbi and Y. A. Alotaibi, "Imagined Speech Recognition and the Role of Brain Areas Based on Topographical Maps of EEG Signal," in *2024 47th International Conference on Telecommunications and Signal Processing (TSP)*. Prague, Czech Republic: IEEE, Jul. 2024, pp. 274–279.

[106] S. Koelstra, C. Muhl, M. Soleymani, Jong-Seok Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "DEAP: A Database for Emotion Analysis ;Using Physiological Signals," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 18–31, Jan. 2012.

[107] S. Nikolopoulos, P. C. Petrantonakis, K. Georgiadis, F. Kalaganis, G. Liaros, I. Lazarou, K. Adam, A. Papazoglou-Chalikias, E. Chatzilari, V. P. Oikonomou, C. Kumar, R. Menges, S. Staab, D. Müller, K. Sengupta, S. Bostantjopoulou, Z. Katsarou, G. Zeilig, M. Plotnik, A. Gotlieb, R. Kizoni, S. Fountoukidou, J. Ham, D. Athanasiou, A. Mariakaki, D. Comanducci, E. Sabatini, W. Nistico, M. Plank, and I. Kompatsiaris, "A multimodal dataset for authoring and editing multimedia content: The MAMEM project," *Data in Brief*, vol. 15, pp. 1048–1056, Dec. 2017.

[108] Z. Keirn and J. Aunon, "A new mode of communication between man and his surroundings," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 12, pp. 1209–1214, Dec. 1990.

[109] R. Khalil, A. Arasteh, and A. K. Sarkar, "EEG based biometrics using emotional stimulation data," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. Dhaka: IEEE, Dec. 2017, pp. 246–249.

[110] H. Cai, J. Jin, H. Wang, L. Li, Y. Huang, and J. Pan, "AITST—Affective EEG-based person identification via interrelated temporal–spatial transformer," *Pattern Recognition Letters*, vol. 174, pp. 32–38, Oct. 2023.

[111] D. L. Rocca, P. Campisi, B. Vegso, P. Cserti, G. Kozmann, F. Babiloni, and F. D. V. Fallani, "Human Brain Distinctiveness Based on EEG Spectral Coherence Connec-

tivity," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 9, pp. 2406–2412, Sep. 2014.

[112] M. Wang, J. Hu, and H. A. Abbass, "BrainPrint: EEG biometric identification based on analyzing brain connectivity graphs," *Pattern Recognition*, vol. 105, p. 107381, Sep. 2020.

[113] A. Seyfizadeh, R. L. Peach, P. Tovote, I. U. Isaias, J. Volkmann, and M. Muthuraman, "Enhancing security in brain–computer interface applications with deep learning: Electroencephalogram-based user identification," *Expert Systems with Applications*, vol. 253, p. 124218, Nov. 2024.

[114] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, "BCI2000: A general-purpose brain-computer interface (BCI) system," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004.

[115] R. Palaniappan and D. P. Mandic, "Biometrics from Brain Electrical Activity: A Machine Learning Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 738–742, Apr. 2007.

[116] M. K. Abdullah, K. S. Subari, J. L. C. Loong, and N. N. Ahmad, "Analysis of the EEG Signal for a Practical Biometric System," vol. 4, no. 8, 2010.

[117] J. Chuang, H. Nguyen, C. Wang, and B. Johnson, "I Think, Therefore I Am: Usability and Security of Authentication Using Brainwaves," in *Financial Cryptography and Data Security*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. A. Adams, M. Brenner, and M. Smith, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 7862, pp. 1–16.

[118] B. C. Armstrong, M. V. Ruiz-Blondet, N. Khalifian, K. J. Kurtz, Z. Jin, and S. Laszlo, "Brainprint: Assessing the uniqueness, collectability, and permanence of a novel method for ERP biometrics," *Neurocomputing*, vol. 166, pp. 59–67, Oct. 2015.

[119] M. V. Ruiz-Blondet, Z. Jin, and S. Laszlo, "CEREBRE: A Novel Method for Very High Accuracy Event-Related Potential Biometric Identification," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1618–1629, Jul. 2016.

[120] A. G. Marques, N. Kiyavash, J. M. Moura, D. Van De Ville, and R. Willett, "Graph Signal Processing: Foundations and Emerging Directions [From the Guest Editors]," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 11–13, Nov. 2020.

[121] M. R. Nuwer, "10-10 electrode system for EEG recording," *Clinical Neurophysiology*, vol. 129, no. 5, p. 1103, May 2018.

[122] M. Defferrard, L. Martin, R. Pena, and N. Perraudin, "PyGSP: Graph Signal Processing in Python," Zenodo, Oct. 2017.

[123] C. Berge, *Graphs and Hypergraphs*, [2d rev. ed.] translated by edward minieka ed. Amsterdam, New York: North-Holland Pub. Co., 1976.

[124] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[125] "EEG-BASED SPEECH RECOGNITION - Impact of Temporal Effects:," in *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing*. Porto, Portugal: SciTePress - Science and and Technology Publications, 2009, pp. 376–381.

[126] G. A. P. Coretto, I. E. Gareis, and H. L. Rufiner, "Open access database of EEG signals recorded during imagined speech," in *12th International Symposium on Medical Information Processing and Analysis*, vol. 10160. SPIE, Jan. 2017, p. 1016002.

[127] N. Nieto, V. Peterson, H. L. Rufiner, J. E. Kamienkowski, and R. Spies, "Thinking out loud, an open-access EEG-based BCI dataset for inner speech recognition," *Scientific Data*, vol. 9, p. 52, Feb. 2022.

[128] L. C. Sarmiento, S. Villamizar, O. López, A. C. Collazos, J. Sarmiento, and J. B. Rodríguez, "Recognition of EEG Signals from Imagined Vowels Using Deep Learning Methods," *Sensors*, vol. 21, no. 19, p. 6503, Sep. 2021.

[129] B. C. I. Committee, "2020 International BCI Competition," Apr. 2020.

[130] S. Wellington and J. Clayton, "Fourteen-channel EEG with Imagined Speech (FEIS) dataset," Nov. 2019.

[131] F. Simistira Liwicki, V. Gupta, R. Saini, K. De, N. Abid, S. Rakesh, S. Wellington, H. Wilson, M. Liwicki, and J. Eriksson, "Bimodal electroencephalography-functional magnetic resonance imaging dataset for inner-speech recognition," *Scientific Data*, vol. 10, no. 1, p. 378, Jun. 2023.

[132] B. Dekker, A. C. Schouten, and O. Scharenborg, "DAIS: The Delft Database of EEG Recordings of Dutch Articulated and Imagined Speech," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Rhodes Island, Greece: IEEE, Jun. 2023, pp. 1–5.

[133] J. LaRocco, Q. Tahmina, S. Lecian, J. Moore, C. Helbig, and S. Gupta, "Evaluation of an English language phoneme-based imagined speech brain computer interface with low-cost electroencephalography," *Frontiers in Neuroinformatics*, vol. 17, p. 1306277, Dec. 2023.

[134] P. Kumar, R. Saini, P. P. Roy, P. K. Sahu, and D. P. Dogra, "Envisioned speech recognition using EEG sensors," *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 185–199, Feb. 2018.

[135] A. Jahangiri and F. Sepulveda, "The Relative Contribution of High-Gamma Linguistic Processing Stages of Word Production, and Motor Imagery of Articulation in Class Separability of Covert Speech Tasks in EEG Data," *Journal of Medical Systems*, vol. 43, no. 2, p. 20, Feb. 2019.

[136] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, Jun. 2000.

[137] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 589–597.

[138] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.

[139] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[140] E. Lashgari, D. Liang, and U. Maoz, "Data augmentation for deep-learning-based electroencephalography," *Journal of Neuroscience Methods*, vol. 346, p. 108885, Dec. 2020.

[141] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems*, vol. 29. Curran Associates, Inc., 2016.

[142] Q. Zhang and Y. Liu, "Improving brain computer interface performance by data augmentation with conditional Deep Convolutional Generative Adversarial Networks."

[143] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, "EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals," Jun. 2018.

[144] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009.

[145] E. Maiorana, D. L. Rocca, and P. Campisi, "Eigenbrains and Eigentensorbrains: Parsimonious bases for EEG biometrics," *Neurocomputing*, vol. 171, pp. 638–648, Jan. 2016.

[146] N. Boulgouris, D. Hatzinakos, and K. Plataniotis, "Gait recognition: A challenging signal processing technology for biometric identification," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 78–90, Nov. 2005.

[147] N. V. Boulgouris, K. N. Plataniotis, and E. Micheli-Tzanakou, *Biometrics: Theory, Methods, and Applications*. Wiley-IEEE Press, 2009.

[148] Z. Akhtar, C. Micheloni, and G. L. Foresti, "Biometric Liveness Detection: Challenges and Research Opportunities," *IEEE Security & Privacy*, vol. 13, no. 5, pp. 63–72, Sep. 2015.

[149] A. E. Hramov, V. A. Maksimenko, and A. N. Pisarchik, "Physical principles of brain–computer interfaces and their applications for rehabilitation, robotics and control of human brain states," *Physics Reports*, vol. 918, pp. 1–133, Jun. 2021.

[150] J. T. Panachakel and A. G. Ramakrishnan, "Decoding Covert Speech From EEG-A Comprehensive Review," *Frontiers in Neuroscience*, vol. 15, p. 642251, Apr. 2021.

[151] U. Shah, M. Alzubaidi, F. Mohsen, A. Abd-Alrazaq, T. Alam, and M. Househ, "The Role of Artificial Intelligence in Decoding Speech from EEG Signals: A Scoping Review," *Sensors*, vol. 22, no. 18, p. 6975, Sep. 2022.

[152] M. Svetlakov, I. Kovalev, A. Konev, E. Kostyuchenko, and A. Mitsel, "Representation Learning for EEG-Based Biometrics Using Hilbert–Huang Transform," *Computers*, vol. 11, no. 3, p. 47, Mar. 2022.

[153] H. Cho, M. Ahn, S. Ahn, M. Kwon, and S. C. Jun, "EEG datasets for motor imagery brain–computer interface," *GigaScience*, vol. 6, no. 7, Jul. 2017.

[154] H.-L. Chan, P.-C. Kuo, C.-Y. Cheng, and Y.-S. Chen, "Challenges and Future Perspectives on Electroencephalogram-Based Biometrics in Person Recognition," *Frontiers in Neuroinformatics*, vol. 12, p. 66, Oct. 2018.