Understanding concepts in graph signal processing for neurophysiological signal analysis

Stephan Goerttler[1,2], Min Wu[2], Fei He[1]

1. Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry, United Kingdom

2. Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore

Abstract. Multivariate signals measured simultaneously over time by sensor networks are becoming increasingly common. The emerging field of graph signal processing (GSP) promises to analyse spectral characteristics of these multivariate signals, while also taking the spatial structure between the time signals into account. A core idea in GSP is the graph Fourier transform, which projects a multivariate signal onto frequency-ordered graph Fourier modes and can be regarded as a spatial analogue of the classical Fourier transform. This chapter derives and discusses key concepts in GSP, with a specific focus on understanding the differences between parallel formulations and the interconnections between the various concepts. The experimental section focuses on the role of graph frequency in data classification, with applications to neuroimaging. To shed light on graph frequencies individually, sample sizes larger than those of relevant empirical datasets are needed. We therefore introduce a minimalist simulation to generate sufficiently many signals, which share key characteristics with neurophysiological signals. Using this artificial data, we find that higher graph frequency signals are more suitable for classification as compared to lower graph frequency signals, and propose GSP mechanisms to explain our findings. Finally, we present a baseline testing framework for GSP. Using this framework, our results suggest that GSP may be applicable for dimensionality reduction in neurophysiological signals.

1 Introduction

Multivariate signals are composite signals that consist of multiple simultaneous signals, typically time signals. They can be either directly acquired using arrays of spatially separated sensors [1], or inferred using more complex recording techniques, such as magnetic resonance imaging [2]. The lowering cost of sensors has made multivariate signals ubiquitous, which highlights the importance to develop or improve tools to process these signals [3–5]. Examples of multivariate signals are as diverse as temperature measurements at different geographic locations [6], pixels in images or digital movies [7], or the measurement of process variables in nuclear reactors [8]. Multivariate signals are also common in various biomedical imaging applications, such as electroencephalography (EEG) [9], electrocorticography [10], magnetoencephalography (MEG) [11], or functional magnetic resonance imaging (fMRI) [12]. This book chapter will focus primarily on multivariate EEG signals.

Unlike time or spatial signals, multivariate signals can capture the characteristics of two dimensions of the measured system. In EEG, sensors are placed on various locations on the scalp of the brain [9]. Each sensor, also called *channel* in this context, measures the electric field potential in time at its location, giving information about the dynamics of the electric activity. On the other hand, the electric activity of all sensors at one point in time provides information about the spatial characteristics of the system.

EEG recordings are most commonly analysed as time signals using conventional signal processing [13]. Historically, one fruitful approach has been the use of *power spectral density* estimates to extract spectral features for each channel. These features can be used to quantify the power of widely recognised *brain waves*, such as alpha (8-12 Hz), beta (12-35 Hz), gamma (>35 Hz), delta (0.5-4 Hz), and theta (4-8 Hz) waves [14]. The extent of these brain waves, together with their position on the scalp, have been shown to be useful biomarkers for Alzheimer's disease [15], sleep apnea syndrome [16], social anxiety disorders [17], or even performance in sport [18]. However, numerous other features can be extracted from the time signals, which can be both linear and non-linear. For example, non-linear summary metrics such as the Lyapunov exponent may predict the onset of epileptic seizures [19]. A second, traditional method to analyse EEG recordings in time is to relate the electrical activity to specific events such as task stimuli [20]. The electric potential can be averaged to yield the so-called *event-related potential* (ERP). The amplitude of the ERP at specific time delays for relevant tasks has been shown to be affected by disorders such as alcoholism [21] or schizophrenia [22].

Alternatively, multivariate signals can be analysed along their spatial dimension. This type of analysis rests on the assumption that the measured system has a *connectivity structure*, either static or changing with time, which causes some of the signals to be pairwise correlated. For example, Peters *et al.* analysed the connectivity structures retrieved from EEG recordings with tools from *graph theory*, allowing them to characterise autism [23]. Similarly, Atasoy *et al.* analysed the connectivity structure in fMRI recordings using *spectral graph theory* and were able to link the resulting harmonic waves to connectome networks. A comprehensive review of graph analysis methods for neuroimaging applications is given by Fallani *et al.* [24].

However, both the temporal and the spatial analysis of multivariate signals neglect possible spatio-temporal features in these signals, i.d., features that are both present in the temporal and spatial domain. Due to this shortcoming, *graph signal processing* (GSP) has emerged as a new promising framework that analyses the signals in time in dependence on their graph structure [25, 26], thereby taking both domains into account simultaneously. One core idea behind GSP is to transform the data across space based on the connectivity structure, before processing the signals with conventional tools. This spatial transformation is also called the *graph Fourier transform* (GFT), following its classical counterpart. GSP has been successfully employed for *neurophysiological imaging*, such as fMRI [27–30] or EEG [31–33], which will be discussed in more detail in section 3. Comprehensive review articles of GSP are those of Shuman *et al.* [34] and Ortega *et al.* [25], while Stanković *et al.* [5] may be a good resource for readers looking for a more intuitive introduction into GSP.

Currently, the ambiguity of the connectivity structure [35] and how this structure is used may be a significant limitation of GSP, given that the connectivity structure is an essential building block in this theory. Addressing this challenge will be a central theme in this book chapter. To this end, the theoretical section aims to derive GSP concepts from basic principles, juxtapose the alternative formulations in GSP, and explore links between them. On the other hand, the experimental section showcases a strong validation procedure using a novel baseline framework, which is needed in light of these ambiguities. It further shows how simulations can help to understand mechanisms in GSP by overcoming the data scarcity commonly observed in empirical datasets.

The experiment itself evaluates the role of *graph frequencies* of GFT-transformed signals, which may aid in GSP-based dimensionality reduction for neurophysiological signals. Our work complements a study by Ouahidi *et al.*, which aimed to select the most valuable graph frequencies for fMRI decoding [36]. In our study, we classified each graph frequency signal separately using a *support vector machine* (SVM), allowing us to determine the classification accuracy with respect to graph frequency. Our analysis of the results, aided by the carefully designed baseline framework, finally revealed that the superior performance of high-frequency transformed signals can be attributed to the graph structure.

The book chapter is organised as follows: Section 2 gives a mathematical introduction into GSP by firstly covering graph terminology and basic principles, before deriving key concepts in GSP. Section 3 relates these theoretical concepts to neuroimaging. While the first three subsections cover the background for neuroimaging with GSP, the following subsections illustrate several applications of GSP in this field. Section 4 draws important links between the graph retrieval methods and graph representations, both generally as well as applied to neurophysiological signals. Section 5 commences the experimental part of this book chapter and covers the methodology of our experiment, comprising the simulation of the data, the classification model, the testing and the baseline framework. The subsequent section 6 present the results of our study, while section 7 discusses these results. The last section 8 summarises the book chapter and proposes future directions.

2 Concepts in Graph Signal Processing

In this chapter, GSP concepts are developed in analogy to classical signal processing, following work by Sandryhaila *et al.* [37]. Figure 2 gives a comprehensive overview of the relations between the relevant concepts in classical signal processing and how they are extended to GSP. Importantly, this extension is ambiguous and leads to two separate definitions of the GFT, from which some of the other concepts are derived.

2.1 Graph terminology

Formally, networks can be represented by a *weighted graph* $\mathcal{G} := (\mathcal{V}, \mathbf{A})$, which consists of N vertices $\mathcal{V} = \{1, 2, 3, ..., N\}$ and the *weighted adjacency matrix* $\mathbf{A} \in \mathbb{R}^{N \times N}$. The entries a_{ij} of $\mathbf{A}$ represent the strength of the connectivity between node i and node j. If the connectivities between each pair of nodes are symmetric, or $a_{ij} = a_{ji}$ for any i and j,

then $\mathbf{A}$ is symmetric and the graph is said to be undirected; in the other case, the graph is said to be directed. Furthermore, if the diagonal elements a_{ii} are all zero, meaning that the nodes are not connected to themselves with loops, the graph is called a *simple graph*. Figure 1(A) depicts a weighted directed simple graph with $N = 6$ vertices.

A second, useful representation of a graph $\mathcal{G}$ is its *Laplacian matrix*. The Laplacian can be directly computed from the adjacency matrix $\mathbf{A}$ as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Here, $\mathbf{D} = \mathrm{diag}(\mathbf{A} \cdot \mathbf{1})$ denotes the *degree matrix*. Alternatively, the symmetric normalised Laplacian can be used instead of the Laplacian, which is computed as $\mathbf{L}_{\mathrm{norm}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. While the weighted adjacency matrix can be viewed as a *graph shift operator* (GSO, section 2.2), the Laplacian can be associated with the negative *difference operator* (section 2.3).

A multivariate signal $\mathbf{X} \in \mathbb{R}^{N \times N_t}$ is a data matrix consisting of N univariate signals each of length N_t, typically temporal signals. We denote the univariate signal at node i by $\mathbf{x}_{i*}$. The rows of $\mathbf{X}$, on the other hand, correspond to *graph signals* and contain one value for each node on the graph. We denote the graph signals as either $\mathbf{x}$ or more explicitly as $\mathbf{x}_{*j}$, where j is the column index.

2.2 Graph shift operator (GSO)

The weighted adjacency matrix is the canonical algebraic representation of a graph. This subsection further explores the role of the adjacency matrix as a GSO and its implications for graph dynamics. In particular, these graph dynamics are capitalised for generating the artificial multivariate signals in subsection 5.1.

In *discrete signal processing* (DSP), a cyclic time signal s with N samples can be represented algebraically by a time-ordered vector $\mathbf{s} = (s_0, ..., s_{N-1})^\top$, where s_0 and s_{N-1} are the time samples at the first and the last time step, respectively. The *shift operator* T_1 shifts each time sample s_n to the subsequent time step,

$$T_1 : (s_0, ..., s_{N-1})^\top \mapsto (s_{N-1}, s_0, ..., s_{N-2})^\top,$$

and can be algebraically represented by the cyclic shift matrix

$$\mathbf{A}_c = \begin{bmatrix} 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & \dots & 1 & 0 \end{bmatrix}. \tag{1}$$

Note that the last time sample s_{N-1} is shifted to the first time step, as we assume the time signal to be cyclic.

Importantly, $\mathbf{A}_c$ can be interpreted as the adjacency matrix of a graph with a linear topology with N vertices, as shown in Figure 1(B). The interpretation of the discretised time dimension as a linear time graph allows to generalise the shift operator to more arbitrary graph topologies, such as the one shown in Figure 1(A). Consequently, the GSO of a graph is simply given by its weighted adjacency matrix $\mathbf{A}$. Applied to a graph signal $\mathbf{x}$, the adjacency matrix spatially shifts a signal at node i to its neighbouring nodes,

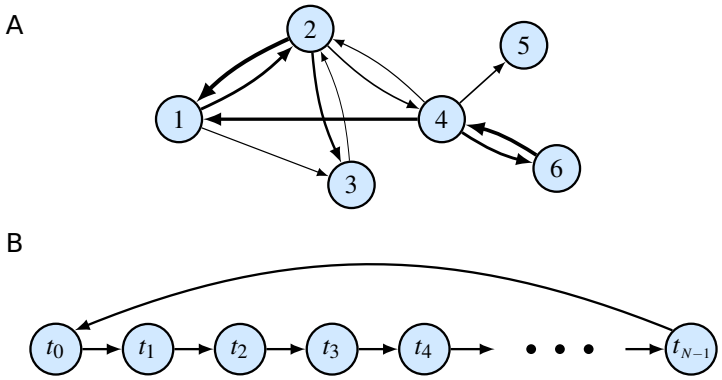

Fig. 1: (A) Graph with six nodes, which can be represented by an adjacency matrix $\mathbf{A}$. The displayed connections are directed, meaning that they can go in both directions. They are also weighted, which is indicated by the line width of the arrows. The graph can represent an arbitrary spatial graph topology of a multivariate signal acquired from a measurement system with six sensors. (B) Time graph with linear topology. The graph connects each time step t_i to the subsequent time step t_{i+1}, thereby shifting a signal in time. For periodic signals, the last time step is connected to the first time step. The graph can be represented by the cyclic shift matrix $\mathbf{A}_c$

weighted by the strength of the connection. The GSO may also be used to model graph dynamics. This follows the assumption that the time evolution from one time step to the next corresponds to a scaled graph shift, that is, the GSO is taken to be a "time evolution operator". However, this crude approximation only describes the dynamics of the signals which are due to the connectivity structure. Nevertheless, this characterisation of the GSO proves to be useful to simulate a multivariate signal in subsection 5.1. A more sophisticated way to ascribe the graph dynamics to the connectivity structure is to model the dynamics as a heat diffusion process [35], which involves the Laplacian operator. The Laplacian operator itself is based on the GSO, as shown in the following section 2.3.

2.3 Laplacian as negative difference operator

In discrete calculus, the operator Δ defines a *forward difference* in terms of the shift operator T_1 and the identity operator I:

$$\Delta = T_1 - I.$$

In the case of a finite, cyclic signal, the shift operator can be represented algebraically by the cyclic shift matrix $\mathbf{A}_c$ as defined in equation (1), while the identity operator can be represented by the degree matrix $\mathbf{D}_c = \mathbb{1}$. The negative forward difference operator $-\Delta$ can then be represented by the Laplacian matrix $\mathbf{L}_c$:

$$\mathbf{L}_c = \mathbf{D}_c - \mathbf{A}_c.$$

Likewise, the second-order central difference operator Δ^2 is given in terms of the right shift operator T_{-1}:

$$\Delta^2 = T_1 - 2I + T_{-1}$$
$$T_{-1} : (s_0, ..., s_{N-1})^\top \mapsto (s_1, ..., s_{N-1}, s_0)^\top.$$

The right shift operator T_{-1} can be represented algebraically by the transposed adjacency matrix $\mathbf{A}_c^\top$, such that the operator $-2\Delta^2$ can be represented by the symmetrised Laplacian matrix

$$\mathbf{L}_{c,\text{sym}} = \mathbf{D}_c - \frac{\left(\mathbf{A}_c + \mathbf{A}_c^\top\right)}{2}.$$

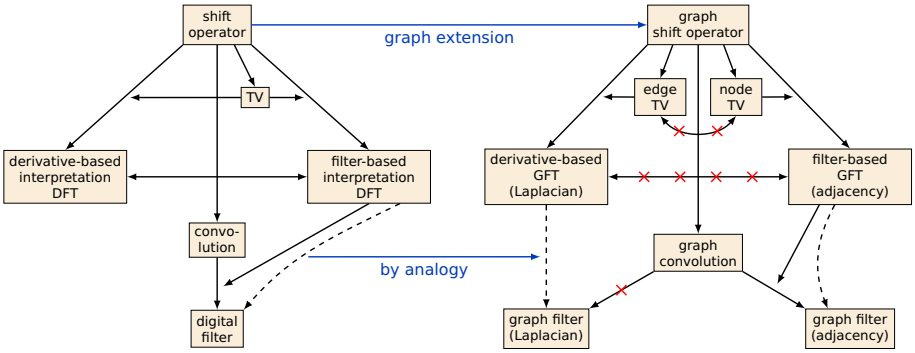

Fig. 2: Interconnections between discrete signal processing (DSP) concepts and its graph extensions. In DSP, the discrete Fourier transform (DFT) can be traced back to the shift operator. Crucially, a derivative-based interpretation and a filter-based interpretation of the DFT are equivalent. The total variation (TV) increases with increasing Fourier frequency, thereby linking the two concepts. Convolution and digital filters are built on the concept of the shift operator, whereby the digital filter can be expressed using the DFT. Extending the shift operator to graphs allows to build analogous concepts for graphs. Importantly, the derivative- and the filter-based graph Fourier transform (GFT) are not equivalent. The graph Fourier modes in both GFT variants can be ordered by their frequency using either the edge-based or the node-based TV. The graph convolution can be used to define an adjacency matrix-based graph filter. The similar, Laplacian matrix-based graph filter can only be constructed by analogy and is not directly linked to the graph convolution

2.4 Graph Fourier Transform (GFT)

The GFT is an extension of the Fourier transform to graphs and is at the heart of GSP. Here, we show that it can be derived either via graph filtering (*filter-based GFT*) [25,

38] or via discrete derivatives on graphs (*derivative-based GFT*) [38, 39]. Both these approaches are built on the extension of the shift operator to graphs, i.d., the GSO given by the adjacency matrix $\mathbf{A}$. While both approaches yield the same result in DSP, their extensions to GSP turn out to be slightly different: The derivative-based GFT is composed of the eigenvectors of the adjacency matrix, whereas the filter-based GFT is composed of the eigenvectors of the Laplacian matrix.

The two approaches mirror the two definitions for the total variation in section 2.5. Section 4.2 shows theoretically why the two approaches can nevertheless be similar to each other in some practical applications.

Filter-based Graph Fourier Transform The derivation in this subsection follows Sandryhaila *et al.* [40] and Ortega *et al.* [25]. Let s be a time signal represented by a time-ordered vector $\mathbf{s} = (s_0, ..., s_{N-1})^\top$ and T_1 a shift operator represented algebraically by $\mathbf{A}_c$, as defined in equation (1). A *finite impulse response filter* h of order N can be characterised as a sum of the N samples $s_{in}[n-i \mod N]$ preceding the time step n, weighted by p_i:

$$s_{out}[n] = (h \cdot s_{in})[n] = \sum_{i=0}^{N-1} p_i s_{in}[n-i \mod N].$$

Importantly, this means that the filter h can be represented as a N-th order polynomial in the time shift operator T_1, as defined in subsection 2.2:

$$h = \sum_{i=0}^{N-1} p_i T_1^i.$$

Algebraically, the filter is given by the matrix $\mathbf{H}$ as a polynomial in the cyclic shift matrix $\mathbf{A}_c$ as defined in equation (1):

$$\mathbf{H} = \sum_{i=0}^{N-1} p_i \mathbf{A}_c^i.$$

The eigenvectors of $\mathbf{A}_c$ are the complex exponentials $\mathbf{v}_k = (\omega^0, \omega^k, ..., \omega^{(N-1)k})^\top / \sqrt{N}$, with $\omega^k = e^{2\pi jk/N}$ and the imaginary unit $j = \sqrt{-1}$. The eigenvalue for each complex exponential $\mathbf{v}_k$ is ω^k. Consequently, the eigendecomposition of $\mathbf{A}_c$ is given by:

$$\mathbf{A}_c = \mathbf{Q}_c \Lambda_c \mathbf{Q}_c^{-1} = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N) \begin{bmatrix} \omega^0 & & & \\ & \omega^1 & & \\ & & \ddots & \\ & & & \omega^{N-1} \end{bmatrix} (\mathbf{v}_1^*, \mathbf{v}_2^*, ..., \mathbf{v}_N^*)^\top. \tag{2}$$

Crucially, the matrix $\mathbf{Q}_c^{-1}$, comprising the complex conjugated exponentials $\mathbf{v}_k$, can be identified as the discrete Fourier transform (DFT) matrix.

The eigenvectors $\mathbf{v}_k$ of $\mathbf{A}_c$ are also eigenvectors of the filter matrix $\mathbf{H}$,

$$\mathbf{H}\mathbf{v}_k = \sum_{i=0}^{N-1} p_i \mathbf{A}_c^i \mathbf{v}_k = \left(\sum_{i=0}^{N-1} p_i \left(\omega^k \right)^i \right) \mathbf{v}_k, \tag{3}$$

which of course is the well-known fact in DSP that filters can change the magnitude or the phase of a sinusoidal signal, but not its frequency.

The central role that the shift operator plays in constructing the filter and its relation to the DFT justifies to define the GFT in terms of the GSO, which can be represented algebraically by the adjacency matrix $\mathbf{A}$ (see subsection 2.2). In analogy to DSP, the filter-based GFT for a spatial signal $\mathbf{x} \in \mathbb{R}^N$ on a graph $\mathbf{A}$ is defined using the eigendecomposition $\mathbf{A} = \mathbf{Q}_\mathbf{A} \Lambda_\mathbf{A} \mathbf{Q}_\mathbf{A}^{-1}$:

$$\tilde{\mathbf{x}} := \mathbf{Q}_\mathbf{A}^{-1} \mathbf{x}, \tag{4}$$

where $\tilde{\mathbf{x}}$ is the transformed spatial signal. The eigenvalues λ_k of $\mathbf{A}$ are the diagonal elements of $\Lambda_\mathbf{A}$. If $\mathbf{A}$ is symmetric, they are real-valued and can be sorted in ascending order, and the eigenvectors $\mathbf{v}_k$ become orthogonal.

However, other matrices, such as the Laplacian matrix $\mathbf{L}_c = \mathbf{D}_c - \mathbf{A}_c$, have the same eigenvectors as $\mathbf{A}_c$ and can be similarly decomposed using the DFT matrix. Importantly, this degeneracy in the DSP case is lifted when the cyclic graph is extended to more complex graph topologies, as also illustrated in the overview in Figure 2. This yields the alternative, derivative-based definition of the GFT, which is derived in the following subsection.

Derivative-based Graph Fourier Transform In the classical Fourier transform, *Fourier modes* are complex exponentials $e^{-j\omega t}$. These exponentials are trivially eigenfunctions of the partial time-derivative $\partial/\partial t$ [38, 39], as well as the second partial time-derivative $\partial^2/\partial t^2$:

$$\begin{aligned} \frac{\partial}{\partial t} e^{-j\omega t} &= -j\omega e^{-j\omega t} \\ \frac{\partial^2}{\partial t^2} e^{-j\omega t} &= -\omega^2 e^{-j\omega t}, \end{aligned}$$

with eigenvalues of $-j\omega$ and $-\omega^2$, respectively.

As shown in subsection 2.3, in DSP the directed Laplacian matrix $\mathbf{L}_c$ of a cyclic shift graph is the algebraic representation of the negated forward difference operator $-\Delta$, whereas the symmetrised Laplacian matrix $\mathbf{L}_{c,\mathrm{sym}}$ is the algebraic representation of the operator $-2\Delta^2$. Crucially, the eigenvectors $\mathbf{v}_k$ of $\mathbf{A}_c$, which constitute the discrete Fourier modes, are also eigenvectors of $\mathbf{L}_c$ with eigenvalues $\lambda'_k = (1 - \lambda_k)$:

$$\mathbf{L}_c \mathbf{v}_k = (\mathbf{D}_c - \mathbf{A}_c)\, \mathbf{v}_k = 1 \cdot \mathbf{v}_k - \lambda_k \mathbf{v}_k = \lambda'_k \mathbf{v}_k.$$

In other words, the discrete Fourier modes are eigenvectors of the negated forward difference operator, in the same way that continuous Fourier modes are eigenfunctions of the time-derivative. As a consequence, the eigendecomposition of $\mathbf{L}_c$ is given by:

$$\mathbf{L}_c = \mathbf{Q}_c^{-1}(\mathbb{1} - \Lambda_c)\,\mathbf{Q}_c =: \mathbf{Q}_c^{-1}\Lambda_c'\mathbf{Q}_c,$$

where Λ_c and $\mathbf{Q}_c$ are the eigenvalue and the eigenvector matrix of $\mathbf{A}_c$, respectively, as defined by equation (2).

The Fourier modes can be interpreted as eigenmodes of the derivative, or in the discrete case as eigenmodes of the forward difference. Leveraging this notion leads to the second extension of DFT to graphs. Accordingly, the derivative-based GFT for a spatial signal $\mathbf{x} \in \mathbb{R}^{N_c}$ on a graph with adjacency matrix $\mathbf{A}$ is defined as follows:

$$\begin{aligned} \mathbf{L} &= \mathbf{D} - \mathbf{A} = \mathbf{Q}_\mathbf{L}\Lambda_\mathbf{L}\mathbf{Q}_\mathbf{L}^{-1} \\ \tilde{\mathbf{x}} &:= \mathbf{Q}_\mathbf{L}^{-1}\mathbf{x}. \end{aligned} \tag{5}$$

In the literature, the symmetrically normalised Laplacian $\mathbf{L}_\text{norm}$ is sometimes used instead of the Laplacian [25, 38, 41]. In the case of a graph $\mathbf{A}_c$ with linear topology, the degree matrix $\mathbf{D}_c$ is given by the identity matrix, which means that the symmetrically normalised Laplacian $\mathbf{L}_{c,\text{norm}} = \mathbf{D}_c^{-1/2}\mathbf{L}_c\mathbf{D}_c^{-1/2}$ of this graph is equivalent to the Laplacian $\mathbf{L}_c$. Consequently, the eigendecomposition of $\mathbf{L}_{c,\text{norm}}$ is also given in terms of the DFT matrix.

Lastly, note that eigenvectors $\mathbf{v}_k$ of $\mathbf{L}_c$ with $k > 0$ are not the same as the eigenvectors of the symmetrised Laplacian $\mathbf{L}_{c,\text{sym}} = \mathbf{D}_c - \left(\mathbf{A}_c + \mathbf{A}_c^\top\right)/2$: While the eigenvectors $\mathbf{v}_k$ are complex-valued for $k > 0$, the eigenvectors of the $\mathbf{L}_{c,\text{sym}}$ are necessarily real-valued due to the symmetry of $\mathbf{L}_{c,\text{sym}}$. As a consequence, the eigendecomposition of $\mathbf{L}_{c,\text{sym}}$ is not given in terms of the DFT matrix; in other words, the eigendecomposition of the symmetrised Laplacian of a graph with linear topology does not reduce to the DFT. However, given the analogy of $\mathbf{L}_{c,\text{sym}}$ to the second partial time-derivative in the continuous case, of which the Fourier modes are also eigenfunctions, it may still be valid to use the symmetrised Laplacian $\mathbf{L}_\text{sym}$ of an arbitrary graph structure for the GFT in lieu of $\mathbf{L}$. This may be especially useful if the measurement of the graph structure is intrinsically symmetric, while the actual graph structure is not. For example, when assessing the structural connectivity by measuring white matter between brain regions, the directionality cannot be determined due to limitations in the methodology, and only the symmetric graph is retrieved.

In both variants, the GFT is a linear spatial transformation that is retrieved by computing the eigendecomposition of an algebraic graph representation. The rows of the GFT-matrix, which are the eigenvectors of the eigendecomposition, represent the *graph Fourier modes*. The computational complexity of this eigendecomposition is small as long as the number of graph nodes is sufficiently small, which is typically the case for neurophysiological applications. The GFT itself corresponds to a matrix multiplication

with matrices of size $N \times N$ and $N \times N_t$, thereby scaling linearly with the number of time samples N_t.

While equations (4) and (5) define the GFT for spatial signals, the GFT can be straightforwardly extended to multivariate signals $\mathbf{X} \in \mathbb{R}^{N \times N_t}$:

$$\tilde{\mathbf{X}} = \mathbf{Q}_{\mathbf{A}/\mathbf{L}}^{-1} \mathbf{X},$$

where the subscript $\mathbf{A}/\mathbf{L}$ indicates that the GFT can either be chosen to be filter-based ($\mathbf{A}$, computed using the adjacency matrix) or derivative-based ($\mathbf{L}$, computed using the Laplacian matrix). In the matrix multiplication on the right-hand side, each column $\mathbf{x}_{*j}$, corresponding to a spatial signal at time j, is transformed successively. The rows $\tilde{\mathbf{x}}_{i*}$ of the transformed multivariate signal $\tilde{\mathbf{X}}$ are the *graph frequency signals*, associated with the graph frequency i and the eigenvalue λ_i. Note that each graph frequency signal can also be thought of as a linear combination of the N time signals $\mathbf{x}_{i*}$.

2.5 Total variation (TV)

A useful statistic of a graph signal $\mathbf{x} \in \mathbb{R}^{N_c}$ is its *total variation* (TV), which is a measure of how smooth the signal is across the graph structure $\mathbf{A}$. The TV is based on the local variation, which in turn is based on the graph derivative of the graph signal. The graph derivative can be defined either on the node i or on the edge (i, j), leading to two different definitions of the TV, here referred to as the *node-based* [38] and the *edge-based* [34] TV.

The node derivative calculates the difference between the signal and the signal shifted by the GSO, yielding a derivative at node i:

$$\nabla_i(\mathbf{x}) \coloneqq [\mathbf{x} - \mathbf{A}\mathbf{x}]_i = x_i - \sum_j a_{ij} x_j.$$

The local variation is then given by the magnitude $\|\nabla_i(\mathbf{x})\|_1$ of the graph derivative.

The edge derivative, on the other hand, weighs the difference between the signal at node i and the signal at node j by their connectivity, yielding a derivative along the edge (i, j):

$$\nabla_{ij}(\mathbf{x}) \coloneqq \sqrt{a_{ij}}(x_i - x_j).$$

The local variation is then computed using the p-norm of the derivative vector $\nabla_{i*}(\mathbf{x})$:

$$\|\nabla_{i*}(\mathbf{x})\|_p = \left(\sum_j a_{ij}^{\frac{p}{2}} |x_i - x_j|^p \right)^{\frac{1}{p}}, \quad p \in \mathbb{N}.$$

In both cases, the TV of a graph signal is computed from the sum of the local variation across all nodes. Given a graph structure $\mathbf{A}$, the node-based [38] and the edge-based [34] TV of a graph signal $\mathbf{x}$ are then defined as follows:

$$\begin{aligned} \mathrm{TV}_{\mathbf{A}}^{(n)}(\mathbf{x}) &\coloneqq \sum_i \|\nabla_i(\mathbf{x})\|_1 = \sum_i \left| x_i - \sum_j a_{ij} x_j \right| = \|\mathbf{x} - \mathbf{A}\mathbf{x}\|_1 \\ \mathrm{TV}_{\mathbf{A}}^{(e)}(\mathbf{x}) &\coloneqq \frac{1}{2} \sum_i \|\nabla_{i*}(\mathbf{x})\|_2^2 = \frac{1}{2} \sum_i \sum_j a_{ij} (x_i - x_j)^2. \end{aligned} \tag{6}$$

Note that the node-based TV can also be computed from the normalised adjacency matrix $\mathbf{A}_{\text{norm}} = \mathbf{A}/|\lambda_{\max}|$, where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{A}$, which ensures numerical stability. While the node-based TV is always either positive or zero, the edge-based TV can become negative if we allow negative weights $a_{ij} < 0$ in the adjacency matrix. Section 3.1 discusses the validity of negative weights and possible implications.

The TV allows to understand the ordering of the graph eigenvectors as frequencies in terms of their eigenvalue. In the case of the node-based TV with $\mathbf{A}_{\text{norm}}$, the TV for a normalised eigenvector $\mathbf{v}_k$ with real eigenvalue λ_k is given by:

$$\text{TV}^{(n)}_{\mathbf{A}_{\text{norm}}}(\mathbf{v}_k) = \left|1 - \frac{\lambda_k}{\lambda_{\max}}\right|.$$

It can easily be seen that the following holds for two eigenvectors $\mathbf{v}_k$ and $\mathbf{v}_l$ of $\mathbf{A}_{\text{norm}}$ with respective real eigenvalues λ_k and λ_l:

$$\begin{aligned} &\lambda_k < \lambda_l \\ \Rightarrow &\text{TV}^{(n)}_{\mathbf{A}_{\text{norm}}}(\mathbf{v}_k) > \text{TV}^{(n)}_{\mathbf{A}_{\text{norm}}}(\mathbf{v}_l). \end{aligned}$$

In other words, the ordering of the eigenvalues from highest to lowest orders the graph eigenvectors from lowest to highest frequency. Notice that any eigenvector $\mathbf{v}_k$ of $\mathbf{A}_{\text{norm}}$ is also an eigenvector of $\mathbf{A}$.

A similar relationship between eigenvalue and frequency can be established for the edge-based TV if we assume the adjacency matrix to be symmetric, i.d., $a_{ij} = a_{ji}$. To this end, the TV is firstly expressed in terms of the Laplacian:

$$\begin{aligned} \text{TV}^{(e)}_{\mathbf{A}}(\mathbf{x}) &= \frac{1}{2}\sum_{i,j} a_{ij}(x_i - x_j)^2 \\ &\overset{a_{ij}=a_{ji}}{=} 2\left(\frac{1}{2}\sum_{i,j} a_{ij}x_i^2\right) - \frac{1}{2}\sum_{i,j} a_{ij}2x_i x_j \\ &= \mathbf{x}^\top \begin{bmatrix} \sum_j a_{1j} & & \\ & \ddots & \\ & & \sum_j a_{N_c j} \end{bmatrix} \mathbf{x} - \mathbf{x}^\top \mathbf{A}\,\mathbf{x} = \mathbf{x}^\top \mathbf{L}\,\mathbf{x}. \end{aligned}$$

This representation of the TV allows to link the eigenvalues to graph frequencies. For a normalised eigenvector $\mathbf{v}_k$ of $\mathbf{L}$ with eigenvalue λ_k, the TV is given by:

$$\text{TV}^{(e)}_{\mathbf{A}}(\mathbf{v}_k) = \mathbf{v}_k^\top \mathbf{L}\,\mathbf{v}_k = \lambda_k \|\mathbf{v}_k\|_2 = \lambda_k.$$

Hence, the frequency of the eigenvector $\mathbf{v}_k$ of $\mathbf{L}$, which is given in terms of its TV, is directly linked to its eigenvalue. Importantly, here the eigenvectors of the Laplacian matrix are linked to the graph frequencies, while in the case of the node-based TV the ones of the adjacency matrix are linked to the graph frequencies. This explains why here higher eigenvalues correspond to higher graph frequencies, while in the case of the node-based TV the relationship is inversed.

If the Laplacian matrix is symmetric, the eigendecomposition is given by $\mathbf{L} = \mathbf{Q_L} \Lambda_\mathbf{L} \mathbf{Q_L}^\top$, and the expression can be further rewritten as follows:

$$\begin{aligned} \mathrm{TV}_\mathbf{A}^{(e)}(\mathbf{x}) &= \mathbf{x}^\top \mathbf{L}\, \mathbf{x} = \mathbf{x}^\top \mathbf{Q_L} \Lambda_\mathbf{L} \mathbf{Q_L}^\top \mathbf{x} \\ &= \tilde{\mathbf{x}}^\top \Lambda_\mathbf{L}\, \tilde{\mathbf{x}} = \sum_k \lambda_k \tilde{x}_k^2. \end{aligned}$$

This identity of the edge-based TV allows to understand how the TV of a graph signal can be decomposed in terms of its graph frequency components $\tilde{x}_k$.

2.6 Graph convolution

The definition of the *graph convolution* is built on the graph shift operator defined in subsection 2.2. In the classical case, the convolution for two period time signals s and r, represented by the time-ordered vectors $\mathbf{s} = (s_0, ..., s_{N-1})^\top$ and $\mathbf{r} = (r_0, ..., r_{N-1})^\top$, is given by the vector

$$\Big((s * r)[n]\Big)_{1 \le n \le N}^\top = \left(\sum_i r_i \mathbf{A}_c^i \right) \mathbf{s}.$$

The following two identities can be directly derived from the eigendecomposition of $\mathbf{A}_c$ in equation (2):

$$\begin{aligned} \mathbf{A}_c^i &= \mathbf{Q}_c \Lambda_c^i \mathbf{Q}_c^{-1} \\ \sum_i r_i \Lambda_c^i &= \mathrm{diag}(\mathbf{Q}_c^{-1} \mathbf{r} / \sqrt{N}). \end{aligned} \tag{7}$$

Using these two identities, the vector of the convoluted signal $s * r$ can be rewritten as follows:

$$\begin{aligned} \Big((s * r)[n]\Big)_{1 \le n \le N}^\top &= \sum_i r_i\, \mathbf{Q}_c \Lambda_c^i \mathbf{Q}_c^{-1} \mathbf{s} \\ &= \mathbf{Q}_c\, \mathrm{diag}\left(\mathbf{Q}_c^{-1} \mathbf{r} / \sqrt{N} \right) \mathbf{Q}_c^{-1} \mathbf{s} \\ &= \mathbf{Q}_c \frac{1}{\sqrt{N}} (\mathbf{Q}_c^{-1} \mathbf{r}) \circ (\mathbf{Q}_c^{-1} \mathbf{s}). \end{aligned}$$

The analogy between the classical shift operator and the GSO can be used to define the graph convolution on two spatial signals $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbf{x} * \mathbf{y} = \left(\sum_i y_i \mathbf{A}^i \right) \mathbf{x}. \tag{8}$$

Using the filter-based GFT, we can use the identity

$$\mathbf{A}^i = \mathbf{Q_A} \Lambda_\mathbf{A}^i \mathbf{Q_A}^{-1}, \tag{9}$$

which can be trivially derived from the eigendecomposition of $\mathbf{A}$, to rewrite the graph convolution:

$$\begin{aligned}\mathbf{x} * \mathbf{y} &= \sum_i y_i \mathbf{Q}_\mathbf{A} \Lambda_\mathbf{A}^i \mathbf{Q}_\mathbf{A}^{-1} \mathbf{x} \\ &= \mathbf{Q}_\mathbf{A} \operatorname{diag}\left(\sum_i y_i \lambda_0^i, ..., \sum_i y_i \lambda_{N-1}^i\right) \mathbf{Q}_\mathbf{A}^{-1} \mathbf{x}.\end{aligned}$$

The crucial difference between the classical convolution and the graph convolution is that the identity (7) does not have a comparable equivalent in GSP.

We can also implicitly define the graph convolution for the case of the derivative-based GFT by analogy to the classical convolution:

$$\mathbf{x} * \mathbf{y} = \mathbf{Q}_\mathbf{L} \left(\sum_i y_i \Lambda_\mathbf{L}^i\right) \mathbf{Q}_\mathbf{L}^{-1} \mathbf{x} = \left(\sum_i y_i \mathbf{L}^i\right) \mathbf{x}. \tag{10}$$

This definition of the graph convolution has been commonly used in the literature [42–45]. Note, however, that this definition is implicitly based on the inadequate assumption that the Laplacian matrix is a GSO, which can be seen by comparing the last equation (10) to the GSO-based convolution in equation (8).

2.7 Graph signal filtering

In classical signal processing, filters can be described both by their impulse response in the time domain as well as by their frequency response in the Fourier frequency domain; the two descriptions are linked by the DFT. *Graph filters* can be similarly defined by their impulse and by their frequency response; here the descriptions are linked by the GFT [46]. The former, impulse response description was previously encountered in subsection 2.4 and is closely related to the concept of convolution as introduced in the previous subsection 2.6. The frequency response description is more convenient to define graph spectral band-pass filters, which is essential for tasks such as graph denoising. For example, a low-pass filter can be used to reduce the total variation of the spatial signal. Alternatively, graph impulse response filters can also be used to minimise the node-based TV together with a similarity term, as shown in Chen *et al.* [47]. However, this approach is less intuitive, as it requires to explicitly compute the filter parameters.

In the time domain, a filter $\mathbf{H}$ can be built by accessing previous signal values through shifting the signal $\mathbf{x}$ and weighing each of these values by p_i, as previously shown in equation (3):

$$\mathbf{H}\mathbf{x} = \left(\sum_{i=0}^{N-1} p_i \mathbf{A}_c^i\right) \mathbf{x} = h(\mathbf{A}_c)\mathbf{x} = \tilde{\mathbf{x}}, \tag{11}$$

where $\tilde{\mathbf{x}}$ is the filtered graph signal and h is the polynomial filter function.

Using the identity (9) with $\mathbf{A} = \mathbf{A}_c$,

$$\mathbf{A}_c^i = \mathbf{Q}_c \Lambda_c^i \mathbf{Q}_c^{-1},$$

we can rewrite equation (11):

$$\begin{aligned}\mathbf{H}\mathbf{x} &= \mathbf{Q}_c \left(\sum_{i=0}^{N-1} p_i \Lambda_c^i \right) \mathbf{Q}_c^{-1}\mathbf{x} = \mathbf{Q}_c\, h(\Lambda_c)\, \mathbf{Q}_c^{-1}\mathbf{x} \\ &=: \mathbf{Q}_c \,\mathrm{diag}\,(h_0, ..., h_{N-1})\, \mathbf{Q}_c^{-1}\mathbf{x}.\end{aligned}$$

In other words, to define the filter $\mathbf{H}$ we can either use the N parameters p_i, which define the filter by its weights in the time domain, or equivalenty the parameters h_i, which define it by its weights in the Fourier domain.

In analogy to this classical case, we can define a graph filter $\mathbf{H}_{\mathbf{A/L}}$ in the spatial domain for the two definitions of the GFT as follows:

$$\begin{aligned}\mathbf{H}_\mathbf{A}\mathbf{x} &= \left(\sum_{i=0}^{N-1} p_i \mathbf{A}^i \right) \mathbf{x} \\ \mathbf{H}_\mathbf{L}\mathbf{x} &= \left(\sum_{i=0}^{N-1} p_i \mathbf{L}^i \right) \mathbf{x}.\end{aligned}$$

Note that the Laplacian-based GFT filter is not built on the graph convolution, unlike the adjacency matrix-based GFT filter.

As in the classical case, the same filters can be described in the graph spectral domain using the eigenvalue matrix $\Lambda_{\mathbf{A/L}}$:

$$\begin{aligned}\mathbf{H}_{\mathbf{A/L}}\mathbf{x} &= \mathbf{Q}_{\mathbf{A/L}}\, h(\Lambda_{\mathbf{A/L}})\, \mathbf{Q}_{\mathbf{A/L}}^{-1}\mathbf{x} \\ &=: \mathbf{Q}_{\mathbf{A/L}} \,\mathrm{diag}\left(h_0^{(A/L)}, ..., h_{N-1}^{(A/L)}\right) \mathbf{Q}_{\mathbf{A/L}}^{-1}\mathbf{x}.\end{aligned}$$

The parameters p_i or h_i can either be designed or learned using machine learning-approaches. Sometimes, the spatial domain description of the filter is used to avoid computing the eigendecomposition of the adjacency matrix or the Laplacian [42]. To further limit computations or the number of parameters, only the first k parameters p_i are used, such that $p_i = 0$ for $i >= k$.

3 Graph Signal Processing in Biomedical Imaging

Biomedical imaging is one of many applications where multivariate signals are acquired. Brain imaging techniques, such as fMRI, EEG, or MEG, record multiple signals simultaneously in time at different spatial locations. In fMRI, the signal locations are called *voxels*, whereas in MEG and EEG they are called *channels*. An EEG measurement setup with N_c channels which record the brain over a time period T at a sampling rate f yields a data matrix $\mathbf{X} \in \mathbb{R}^{N_c \times N_t}$, where $N_t = \lfloor f \cdot T \rfloor$ is the number of time samples. The following three sections cover graphs and graph Fourier modes in biomedical imaging more generally, before discussing applications of GSP in this field.

3.1 Graph retrieval

The GFT of the multivariate signals relies on the graph, or spatial structure, of the data. The retrieval of the graph from neuroimaging data is not unique, but can be based on the structural, or anatomical, connectivity, the functional connectivity [48], or the geometric location of the sensors [27].

Firstly, the graph can be based on the *functional connectivity* between the signals. This method is purely data-driven and can therefore be used on all data sets. Common choices to build the graph include computing pairwise Pearson correlations or covariances, but nonlinear measures such as mutual information, the phase lage index, or the phase locking value can be used as well [49]. As shown in subsection 4.3, there is a link between GFT using functionally retrieved graphs and PCA. The experimental section of this chapter uses the Pearson correlation to construct the graph (see subsection 5.2). Secondly, a graph can be constructed using the *structural connectivity* between nodes, which can be determined with secondary measurements. For neurophysiological signals, for example, diffusion tensor imaging can be employed [50]. Lastly, the connectivity between the nodes can be determined by their *geometric* properties, such as the pairwise distances between them. The distances can then be mapped to the connectivity strengths. The method only requires knowledge about the data acquisition system.

Often, these different graph retrieval methods can lead to similar graphs, in part because the methods are aiming to estimate the same connectivity structure underlying the neural substrate. As an example of this, the structural and the functional connectivity are generally related to each other [50]. In an attempt to make this relation more explicit, Li *et al.* learned the mapping between structural and functional connectivity, giving insight into how functional connectivity arises from structural connectivity [51]. On a more cautious note, Wang *et al.* give examples why the structural connectivity is not sufficient to explain the dynamics of neural activity [52], and thus the functional connectivity.

Despite some similarities, Horwitz argued that there is no single underlying connectivity structure, but that connectivity should be thought of as "forming a class of concepts with multiple members" [53]. Generally, the choice of the graph retrieval method can alter the interpretation of the graph [54]. For example, Mortaheb *et al.* used a geometric graph to calculate the edge-based total variation, which allowed them to interpret the results as effects of local communication [31]. Nevertheless, the ambiguity of the retrieved graph remains a challenge for GSP. To illustrate this, Ménoret *et al.* observed vastly different results when testing seven different graphs for their GSP-based methodology [27]. An interesting solution to reduce this ambiguity is the construction of a multilayer graph, as was done by Cattai *et al.* with EEG data [55].

One important difference between the graph retrieval methods lies in whether they allow negative connections between nodes or not. For example, some functional connectivity measures, such as the pairwise Pearson correlation, can yield negative connections. On the other hand, structural connectivity measures looking at physical structures in the brain, such as white matter projections, typically cannot discern whether a connection

is positive or negative [56]. The implications of using negative weights in the adjacency matrix are further explored in subsection 3.2.

3.2 Negative graph weights

Some graph retrieval methods, such as the pairwise Pearson correlation, can lead to negative weights in the adjacency matrix. While negative weights may be less intuitive, they can extend the applicability of graphs [57]. In neuroimaging, for example, negative weights may be used to model inhibitory pathways in the brain [58]. This section will discuss some of the mathematical implications of negative weights for GSP.

A first consequence of using a graph with negative weights is that the Laplacian matrix is not necessarily diagonally dominant, which in turn means that the Laplacian can have negative eigenvalues. A downside of this is that the constant, or DC, graph Fourier mode is no longer the lowest Fourier mode, i.d., the mode with the lowest frequency. An example of this can be seen in Figure 3, where the graph Fourier modes for two graphs are given, namely the geometric graph with only positive weights and the Pearson correlation graph with both positive and negative weights. The DC graph Fourier mode is the lowest mode for the geometric graph, which is not the case for the Pearson correlation graph. Secondly, allowing negative weights means that the edge-based TV, as defined in equation (6), can become negative. This is in contrast with the notion of TV as a positive quantity in the field of image processing [59]. A third consequence of negative weights is that they can cause negative entries in the degree matrix $\mathbf{D}$, which means that the square root of the inverse of $\mathbf{D}$, $\mathbf{D}^{-1/2}$, is not real-valued. As a result of this, the symmetric normalised Laplacian $\mathbf{L}_{\text{norm}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, which may be required for certain applications, can not be computed.

3.3 Graph Fourier modes

In *spectral graph theory*, the graph is decomposed into graph Fourier modes. The GFT transforms a spatial signal $\mathbf{x} \in \mathbb{R}^{N_c}$ by projecting it onto the N_c graph Fourier modes, which can be thought of as eigenmodes of the graph. This section intends to shed more light on the meaning of these graph Fourier modes.

Figure 3 shows the graph Fourier modes for a real EEG data set. The dataset was recorded on 20 healthy participants and 20 Alzheimer's patients. The recordings were acquired using 23 bipolar channels at a sampling rate of 2048 Hz. For all participants, three 12 s-long segments were selected from a larger recording. For one participant, one segment is missing. A more detailed description of the data set can be found in Blackburn *et al.* [61]. The top row (A-E) in the figure shows graph Fourier modes computed from the geometric distance between the EEG sensors. The bottom row (F-J) shows modes computed from the functional connectivity, specifically the pairwise Pearson correlation between the sensors, averaged across all participants and segments. Here, the relation between the graph Fourier modes and their corresponding frequency becomes clearer: Lower frequency modes are "waves" with a global pattern spreading across the whole brain, whereas higher modes are highly localised and fast-varying waves. This

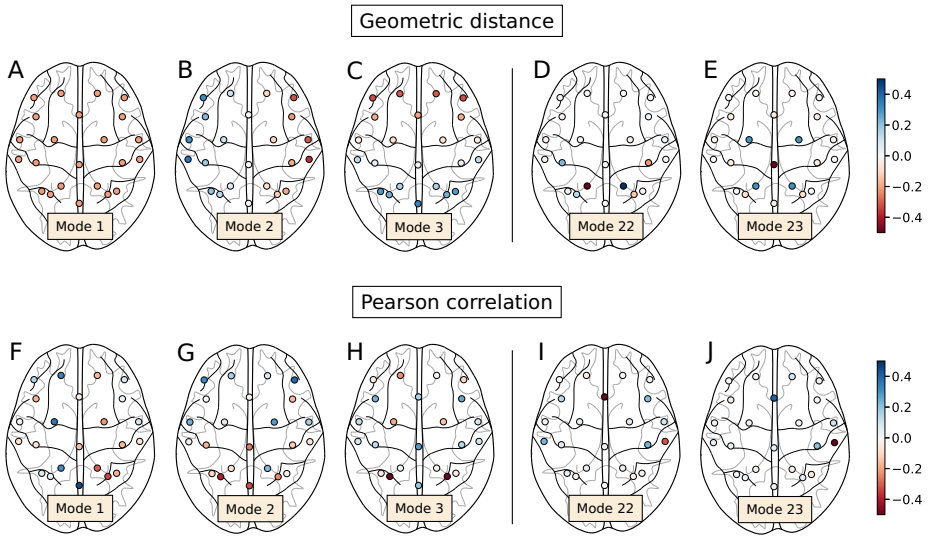

Fig. 3: Lowest and highest graph Fourier modes for a geometric distance-based graph and a functional connectivity-based graph, computed from a real EEG data set and visualised using the Nilearn package in Python [60]. (A-C) The lowest graph Fourier modes of the geometric graph capture the fundamental symmetries and comprise the DC mode, the lateral symmetry mode and the coronal symmetry mode. (D-C) The highest graph Fourier modes are more localised. (F-H) The lowest graph Fourier modes of a functional connectivity Pearson correlation graph, which is computed from a real-world EEG data set. The graph contains negative weights, such that the DC mode is not necessarily the lowest mode. Modes 1 and 2 vaguely reflect the lateral symmetry mode 2 and the coronal symmetry mode 3 of the geometric distance-based graph. (I-J) The highest graph Fourier modes of the Pearson correlation graph are localised and still exhibit a lateral symmetry

general description applies to the two graphs, even though they are retrieved with fundamentally different methods. One major difference between the two graphs is that the DC mode, i.d., the mode with constant entries, is the lowest frequency mode for the geometric distance graph, but not for the Pearson correlation graph. This is a consequence of the negative edge weights in the correlation-based graph. These induce eigenmodes with negative eigenvalues, and therefore have a frequency below that of the DC mode with eigenvalue zero.

The projections of a multivariate signal onto these graph Fourier modes, namely the *graph frequency signals*, may yield insight into the acquired data beyond the raw signal. For example, the projection of the multivariate signal onto the DC mode corresponds to the mean of the signal across the channels. The analysis of the graph frequency signals in terms of their performance on a classification task may shed more light on

the meaning of graph frequency and is the main topic in the experimental part of this book chapter (see sections 5-7).

One major difference between DFT Fourier modes and GFT graph Fourier modes is the magnitude of the eigenvalues associated with these modes, which can take on values other than one in the case of GFT modes. Therefore, when using the GSO as a time evolution operator, the eigenmode coefficients may either vanish or explode. This is in contrast to the shift operator in Euclidean space, which ensures that eigenmode coefficients stay constant in time. One example from physics are electromagnetic waves, which can travel distances of billions of light years due to this property. Previously, an energy-preserving GSO was introduced by Gavili *et al.* [41]; however, this GSO is constructed by simply replacing the eigenvalues of the decomposed GSO with one. While this changes the original GSO, it leaves the GFT and thereby the eigenmodes untouched. The new GSO preserves the energy of the eigenmodes, but it may no longer be an accurate representation of the graph structure.

3.4 Graph-based dimensionality reduction

The GFT projects a spatial or a multivariate signal onto graph Fourier modes, which were the topic of the previous section 3.3. These projections are called the graph frequency signals. Unlike the unaltered time signals, the graph frequency signals can be ordered. This ordering allows to reduce the dimensionality of the signal by selecting specific graph frequencies, such as the k lowest graph frequencies, as for example demonstrated by Rui *et al.* on an MEG dataset [62]. The same dimensionality reduction method can also be used for *graph-based image compression*, as shown by Fracastoro *et al.* [63]. The experimental section of this book chapter will further examine the role that the frequency ordering plays for the graph frequency signals.

Similar as in classical signal processing, graph bandlimited signals correspond to subsampling in the spatial domain [64]. However, unlike in the classical case, loss-free graph sampling for a given band requires sophisticated algorithms. Examples are the greedy algorithm by Anis *et al.* [64] and the random sampling scheme by Puy *et al.* [65]. Both the projection-based and the graph sampling-based dimensionality reduction were evaluated by Ménoret *et al.* on an fMRI classification task [27]. Their results indicate that improvements over graph-free dimensionality reduction methods depend on the graph used. We show in subsection 4.3 that GSP-based dimensionality reduction is equivalent to *principal component analysis* (PCA) for a certain class of graphs.

3.5 Graph filtering

Graph filtering, formally derived in subsection 2.7, is a transformation that can be used for tasks such as graph denoising of a multivariate signal [47]. Graph denoising can be based on the graph frequency response filter using either the adjacency or the Laplacian matrix. In analogy to band-pass Fourier filtering [66], a low-pass graph filter can be applied to the input signal for graph denoising. Intuitively, structurally connected sensors measure similar values, therefore, variations between highly connected sensors should indicate noise. More formally, it has been shown that graph filters minimise the TV [34].

Conversely, variations between connected sensors may carry important information as well, in which case a high-pass graph filter would be more suitable. The graph-filtered signal can then be used for further signal processing tasks. Huang *et al*. use both a graph low-pass and a graph high-pass filter on fMRI signals, and find that the high-pass filter is more useful to correlate signal norms with switch cost in the Navon switching task. Graph band-pass filtering is also the basis for the structural-decoupling index introduced by Preti *et al.*, which compares the norm of graph high-pass and low-pass filtered graph signals [67].

Graph impulse response filters can also be used for graph denoising, albeit less inuitively so. Chen *et al.* applied graph denoising on temperature sensor measurements and for opinion combining. A more sophisticated approach is to adaptively learn the graph filter, which has been used to denoise impulsive EEG signals [68].

Use in neural networks In machine learning applications, graph filtering allows to extend convolutional neural networks to data on graphs. Introduced by Defferrard *et al.* [42], graph neural networks have found widespread applications [69]. Here, a Laplacian-based graph impulse response filter as opposed to a graph frequency response filter is commonly used, which has the advantage that it can be localised on the graph. To decrease the range of suitable filter values, the polynomial of the filter is commonly replaced by a Chebyshev polynomial. The polynomial impulse filter can also be replaced by an auto-regressive moving average filter, as shown by Bianchi *et al.* [70]. In EEG, graph neural networks have been employed to detect emotions [71, 72], to classify error-related potentials [73], or to detect seizures [74].

3.6 Analysis of total variation (TV)

The TV, as introduced in subsection 2.5, is a statistic which measures how much a signal varies on the graph structure. While the previously examined spatial low-pass filter aims to reduce the TV of a signal on a graph, the TV in itself serves as a useful statistic of the graph signal or, when summed over time, the whole multivariate signal. For example, Mortaheb *et al*. found a correlation between the level of consciousness and the TV [31]. Specifically, the authors analysed alpha-band signals in EEGs on a geometric graph derived from the location of the sensors, giving insight into how disorders of consciousness affect short-range and long-range communication in the brain. The TV can also be used to extract features for classification tasks, such as motor imagery tasks from EEG [75].

3.7 Graph learning with Graph Signal Processing (GSP)

Similar to graph denoising, GSP-based *graph learning* exploits the link between the graph structure and the spatial variation of a signal, i.d., the notion that a signal does not vary much on the graph structure. However, instead of changing the signal to reduce the TV, graph learning inversely learns the graph structure to minimise the TV [76]. In other words, the TV is used as an optimisable objective function with the graph structure as its argument. Other terms can be added to this objective function to add further constraints

on the graph. For example, one can further require the graph to have positive weights and to be sparse, i.d., to have few non-zero weights. While this graph-learning method has connections to learning Markov random fields, GSP adds a new perspective in terms of signals by interpreting the graph learning as minimising the TV [77]. Kalofolias has introduced an algorithm in 2016 to solve this graph learning problem [78], which has since been picked up to retrieve graphs from neurophysiological signals [27, 79, 80]. Section 4.1 explores the link between this TV-based graph retrieval and functional connectivity-based graph retrieval.

4 Links in Graph Signal Processing (GSP)

The plethora of graph retrieval methods and graph representations results in a deep ambiguity in GSP. To begin with, there are several ways to retrieve the graph structure for the data (see subsection 3.1). For some of the retrieved raw graphs, further preprocessing steps may be possible or required depending on the application, such as setting negative weights to zero, taking the absolute value of the weights, or normalising the graph (see also subsection 3.2). Lastly, several graph representations can be computed from the preprocessed graphs, such as the adjacency matrix, the Laplacian matrix or the normalised Laplacian matrix (subsection 2.4). This multitude of graphs and representations thereof poses the question of whether there are similarities between them that could minimise the ambiguity in GSP. To this end, this subsection investigates the conditions for which some of the graph retrieval methods are equivalent, and explores the similarities between different graph representations.

4.1 Links between graph retrieval methods

This section specifically looks at the links between functional connectivity-based graphs and total variation-optimised graphs.

Kalofolias already showed such a link for a commonly used graph $\mathbf{A}_{\text{exp}} \in \mathbb{R}^{N\times N}$, which is constructed from the rows $\mathbf{x}_{i*}$ of the data matrix $\mathbf{X}$ as follows [78]:

$$a_{ij}^{(\text{exp})} = \exp\left(-\frac{\|\mathbf{x}_{i*} - \mathbf{x}_{j*}\|_2^2}{\sigma}\right).$$

Given the following regularisation term,

$$f^{(\log)}(\mathbf{A}) = \sigma^2 \sum_i \sum_j a_{ij} \left(\log\left(a_{ij}\right) - 1\right),$$

this functional connectivity-based connectivity matrix minimises the edge-based total variation:

$$\underset{\mathbf{A}\in\mathcal{A}}{\operatorname{argmin}}\ \mathrm{TV}_{\mathbf{A}}^{(e)}(\mathbf{X}) + f^{(\log)}(\mathbf{A}) = \mathbf{A}_{\text{exp}},$$

where $\mathcal{A}$ denotes the set of real-valued $N \times N$ adjacency matrices.

In the following, suppose we are given a normalised multivariate signal $\mathbf{X}_{\text{norm}}$, where each row $\mathbf{x}_{i*}$, corresponding to the time signal at sensor i, is standardised to mean zero and standard deviation one. We show that for such a signal a similar link between a total variation-optimised graph and a functional connectivity-based graph, whose entries in the adjacency matrix are the pairwise Pearson correlations, can be established. Note that due to the normalisation of the time signals, this Pearson correlation matrix is equivalent to the covariance matrix.

We firstly introduce the regularisation term

$$f^{(\mathrm{c})}(\mathbf{A}) = \frac{1}{2}\|\mathbf{J}_N - \mathbf{A}\|_F^2 = \frac{1}{2}\sum_i\sum_j (1-a_{ij})^2, \tag{12}$$

where $\mathbf{J}_N$ is an all-ones matrix of size $N \times N$, and $\|\cdot\|_F$ denotes the Frobenius norm. Specifically, this regularisation term aims to shift the entries a_{ij} of $\mathbf{A}$ closer to 1.

Then, the solution to the optimisation problem

$$\underset{\mathbf{A}\in\mathcal{A}}{\operatorname{argmin}}\, \mathrm{TV}^{(e)}_{\mathbf{A}}(\mathbf{X}_{\text{norm}}) + f^{(\mathrm{c})}(\mathbf{A}) \tag{13}$$

is the adjacency matrix given by the correlation matrix $\mathbf{A}_{\text{corr}}$, where the entries $a_{ij}^{(c)}$ are calculated as the Pearson correlation between the normalised time signals $\mathbf{x}_{i*}$ and $\mathbf{x}_{j*}$ of the sensors i and j, respectively. This can be shown by taking the derivative with respect to a_{ij}:

$$\begin{aligned}
&\frac{\partial}{\partial a_{ij}}\mathrm{TV}^{(e)}_{\mathbf{A}}(\mathbf{X}_{\text{norm}}) + f^{(\mathrm{c})}(\mathbf{A})\Big|_{\mathbf{A}=\mathbf{A}^{\max}} \\
&\qquad = \frac{\partial}{\partial a_{ij}}\sum_{k=1}^{N_t}\sum_{i,j}\left(a_{ij}x_{ik}^2 - a_{ij}x_{ik}x_{jk} + \frac{1}{2}(1-a_{ij})^2\right)\Big|_{a_{ij}=a_{ij}^{\max}} \\
&\qquad = \sum_{k=1}^{N_t}\left(x_{ik}^2 - x_{ik}x_{jk} + (1-a_{ij}^{\max})(-1)\right) \\
&\qquad = \sum_{k=1}^{N_t}\left(1 - x_{ik}x_{jk} - 1 + a_{ij}^{\max}\right) = 0 \\
&\Rightarrow a_{ij}^{\max} = \frac{1}{N_t}\sum_{k=1}^{N_t} x_{ik}x_{jk} = a_{ij}^{(c)}.
\end{aligned}$$

In other words, the correlation matrix $\mathbf{A}_{\text{corr}}$ minimises the total edge-based variation of a normalised data matrix $\mathbf{X}$ given the regularisation term defined in (12). Therefore, it can be equally retrieved by learning the solution to the optimisation problem in (13). Note that the requirement that the signals have to be normalised is naturally given for many neurophysiological signals, such as EEG signals. In our experiment, beginning in section 5, we normalise the data and use the Pearson correlation matrix as the graph for the data.

4.2 Links between graph representations

There are arguments for both using the adjacency or the Laplacian matrix as the GSO in GSP, and both are frequently utilised in the literature. In a direct comparison, Huang *et al.* have observed no noticeable difference between using the adjacency or the Laplacian matrix as the GSO [28]. This observation poses the question of the relation between the use of the adjacency and the Laplacian matrix in GSP. Here, we demonstrate that for large graphs with normally distributed adjacency matrix weights, the two GSOs have similar eigenvectors. This in turn would mean that their respective GFTs are similar.

We here assume that the weights of the adjacency matrix follow a normal distribution with mean μ and standard deviation σ. Accordingly, the entries of the diagonal matrix are given by:

$$d_i = \sum_j a_{ij} = \mu(N-1) + \varepsilon_i,$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2_{\text{diag}})$, and $\sigma_{\text{diag}} = \sigma\sqrt{N-1}$. Note that the standard deviation σ_{diag} of elements d_i on the diagonal is by a factor of $\sqrt{N-1}$ larger than that of the non-diagonal elements a_{ij}. However, for sufficiently large N, the deviation is significantly smaller than the sum of the expected magnitudes of the non-diagonal elements in that row:

$$\begin{aligned}
\sigma_{\text{diag}} = \sigma\sqrt{N-1} \ll & \sigma\sqrt{\frac{2}{\pi}}(N-1) \\
\leq & \sigma\sqrt{\frac{2}{\pi}}(N-1)e^{-\mu^2/2\sigma^2} \\
& + \mu(N-1)\left(1 - 2\Phi\left(-\frac{\mu}{\sigma}\right)\right) \\
= & \sum_{j\neq i} \mathrm{E}\left[|a_{ij}|\right].
\end{aligned}$$

Here, Φ denotes the normal cumulative distribution function. Consequently, the Laplacian $\mathbf{L}$ can be approximated by a matrix $\mathbf{L}'$, which is defined as follows:

$$\begin{aligned}
\mathbf{L} &= \mathbf{D} - \mathbf{A} \\
&= \mu(N-1)\mathbb{1} + \mathrm{diag}(\varepsilon_1, ..., \varepsilon_N) - \mathbf{A} \\
&\approx \mu(N-1)\mathbb{1} - \mathbf{A} =: \mathbf{L}'.
\end{aligned}$$

Crucially, as eigenvectors $\mathbf{u_k}$ of $\mathbf{A}$ are also eigenvectors of $\mathbf{L}'$,

$$\mathbf{L}'\mathbf{u_k} = \mu(N-1)\mathbf{u_k} - \mathbf{A}\mathbf{u_k} = \lambda'_k\mathbf{u_k},$$

the GFT based on the adjacency matrix and the one based on the Laplacian matrix will be similar.

The normality assumption can be a good approximation for highly interconnected networks with negative weights. The Pearson correlation graphs retrieved from the simulated data in the experimental section of this book chapter fulfils both conditions. For

disconnected networks, the assumption is not met as weights between disconnected parts of the graph are zero and thereby not randomly sampled. Considering brain networks, certain pathological conditions, such as split-brain, can cause the brain graph to be disconnected. On the other hand, graphs with only positive graph weights can equally violate the normality assumption if the mean is not much larger than the standard deviation, as this gives rise to a skewed distribution. Many graph retrieval methods yield graphs with only positive weights. The topic of negative graph weights was explored in subsection 3.2.

4.3 Link to principal component analysis (PCA)

PCA projects multivariate samples onto a set of orthogonal components, which are the eigenvectors of the covariance matrix of the data set. Crucially, if the covariance matrix is used for the GFT, then GFT and PCA are mathematically equivalent. Note that for normalised signals with standard deviation one, which is common for some neurophysiological signals such as EEG signals, the correlation and the covariance matrix are the same. Note also that the diagonal entries of the signal correlation matrix are one and will not affect the eigenvectors. In other words, using the correlation matrix of normalised signals for GFT, either with or without diagonal elements, is equivalent to PCA.

5 Methodology

The aim of the experimental section is to expand on previous work by Rui *et al.* [62] and Ménoret *et al.* [27] and investigate dimensionality reduction via GFT projection for classification tasks. The focus will be on understanding the role of graph frequency ordering. To increase the precision needed to compare the graph frequencies, we developed an algorithm to simulate multivariate signals with spatio-temporal features. After transforming the temporal signals into graph frequency signals, our classification model extracts power spectral densities as features from the graph frequency signals and analyses these features using an SVM classifier. We hypothesise that lower graph frequencies amplify spectral features present in the time signals, to the extent that the lower graph frequency signals outperform all other univariate signals tested. On the other hand, higher graph frequency signals, which are based less on the graph structure, should perform equal to or slightly higher than baseline.

5.1 Simulated data set

We developed Algorithm 1 to generate multivariate signal samples $\mathbf{X}_{sim} \in \mathbb{R}^{N_c \times N_t}$ with N_c channels and N_t time samples. The generated samples simulate neuroimaging measurements, such as EEG, with which they share two crucial characteristics: Firstly, the samples have an underlying graph structure, such that each pair of signals has a specific connectivity associated to it. Secondly, the temporal signals have a specific spectral profile.

Algorithm 1 Neurophysiological signal simulation

Input $N_t, \mathbf{A}_s, h, \alpha, \beta, \gamma$
Output X

1: $\varepsilon_{ij}, \tilde{\varepsilon}_{ij} \sim \mathcal{N}(0, 1^2), \quad i = 1, ..., N_c, \; j = 1, ..., N_t$
2: $\widehat{\boldsymbol{\varepsilon}}_{i*} \leftarrow \left(\mathcal{F}^{-1} \circ h \circ \mathcal{F}\right)(\tilde{\boldsymbol{\varepsilon}}_{i*}), \quad i = 1, ..., N_c$
3: $\mathbf{x}_{*1} \leftarrow \beta \widehat{\boldsymbol{\varepsilon}}_{*1}$
4: **for** $t \leftarrow 2, N_t$ **do**
5: $\quad \widehat{\mathbf{x}}_{*t-1} \leftarrow \mathbf{x}_{*t-1} - \bar{\mathbf{x}}_{*t-1} + \gamma \boldsymbol{\varepsilon}_{*t}$
6: $\quad \mathbf{x}_{*t} \leftarrow \alpha \mathbf{A}_s \widehat{\mathbf{x}}_{*t-1} + \beta \widehat{\boldsymbol{\varepsilon}}_{*t}$
7: **end for**

The algorithm takes as input the number of generated time samples N_t, the weighted adjacency matrix $\mathbf{A}_s \in \mathbb{R}^{N_c \times N_c}$, the filter function h, as well as parameters α, β and γ. While $\mathbf{A}_s$ controls the connectivity, or spatial structure, the filter function h controls the spectral profile. Specifically, using two filter functions h_1 and h_2 allows to simulate two separate conditions, whereby the similarity between the two filter functions controls the similarity between the conditions. This may simulate conditions such as Alzheimer's disease or healthy control in EEG data, where each condition has a specific spectral profile [81]. Generally, the more similar the conditions are, the harder it becomes to classify the dataset. Parameter α controls the strength of the correlation structure, while parameter β controls the strength of the spectral structure. Parameter γ controls the self-amplification of the signals during the simulation.

In each time step of Algorithm 1, firstly the graph signal of the previous time step $\mathbf{x}_{*t-1}$ is centred around zero and Gaussian noise $\gamma \boldsymbol{\varepsilon}_{*t}$ is added to this signal (line 5). Secondly, the adjacency matrix $\mathbf{A}_s$ is used as a GSO to translate the graph signal in time, which enforces the structural connectivity $\mathbf{A}_s$ in the data (line 6). Thirdly, the translated signal is scaled by α and normalised coloured noise $\widehat{\boldsymbol{\varepsilon}}_{*t}$ is added scaled with β, whose spectral density profile is controlled by the filter h (line 6). Finally, the simulated multivariate signal is labelled with its condition.

To simulate our data, we used three different filter functions h_2 for condition 2 (orange lines in Figure 4(C)), resulting in data sets which are either easy, medium or difficult to classify. The connectivity structure matrices $\mathbf{A}_s \in \mathbb{R}^{N_c \times N_c}$ with $N_c = 25$ were generated by drawing weights $a^s_{ij} = a^s_{ji} \sim \mathcal{U}_{[-0.1,0.4]}$ from a uniform distribution for $i > j$, and setting $a^s_{ii} = 0.4$. We set the simulation parameters to $\alpha = 0.5$, $\beta = 1$, and $\gamma = 1$. For each simulated participant, we generated $N_c = 25$ time-series signals with $N_t = 2048$ time samples each.

Figure 4 shows the simulated signals (A) along with their spatial (B) and spectral (D) structure. The spatial structure in (B) is shown as the density of channel pairs at a specific correlation value, together with the spatial structure of real EEG recordings (orange, for a description of the dataset see subsection 3.3). The median of the simulated data was matched to that of the EEG recording using the parameters in the algorithm. The spread of the correlation values in EEG recordings is higher, which may be partly due to volume conduction effects not considered in this simulation [82]. How-

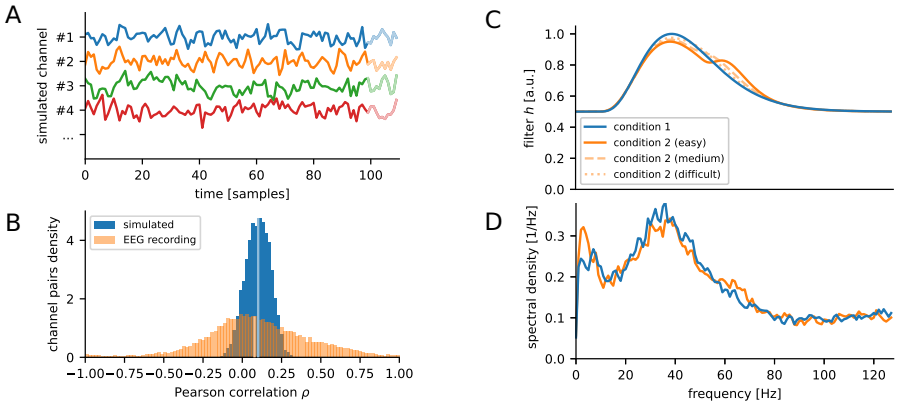

Fig. 4: Simulated neurophysiological signals. (A) Time signals for four selected channels across the first 100 samples. Channels #2, #3 and #4 are positively, weakly and negatively correlated to channel #1, respectively. (B) Density of correlations of channel pairs in blue, exhibiting the spatial connectivity structure. In the simulation, this structure is controlled by the matrix $\mathbf{A}_\mathbf{S}$, but it is also affected by the simulation parameters. Realistic EEG recordings (light orange) exhibit a wider spread of correlations, which is in part due to volume conduction effects not included in the simulation. The median of the simulation (vertical line) was matched to that of the EEG recording. (C,D) Demonstration of power spectral density control and adjustment of classification difficulty. (C) The Fourier filter function h used to colour the noise in Algorithm 1, assuming a sampling frequency of 256 Hz. For each data set, two conditions are simulated. Condition 2 can be varied to make it easy (solid), medium (dashed), or difficult (dotted) to distinguish from condition 1. The difficulty depends on the similarity between the two conditions. (D) Welch power spectral density of simulated signals averaged across all channels for two easily distinguishable conditions. Figures (C) and (D) clearly show that the shape of the power spectral density profile of the simulated signals can be controlled. Parameter α in Algorithm 1 can be used to reduce the power density at lower frequencies

ever, these artefacts can be principally reduced by transforming the EEG signals from the sensor-space to the source-space using sophisticated source reconstruction methods [83], which also allows to recover the true underlying connectivity structure more accurately [84, 85]. The spectral structure (D) of the simulated signals is enforced by the spectral density profile of the coloured noise (C), demonstrating that the spectral structure can be controlled.

5.2 Classification model

The goal of our classification model is to GFT-transform the multivariate signal into graph frequency signals and subsequently classify the samples into the two conditions, using only one graph frequency signal at a time. These graph frequency signals may

capture spectral features present in multiple time signals, or even spatio-temporal features not present in isolated one-dimensional time signals. The structure of our graph frequency analysis is illustrated in Figure 5. The approach allows us to determine the classification performance for each graph frequency, which we used to assess the expressiveness of the spectral structures in this graph frequency signal.

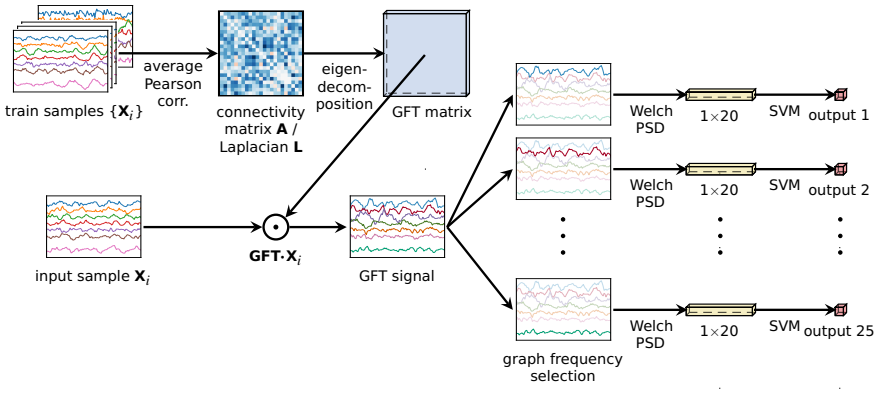

Fig. 5: Illustration of the graph frequency analysis. For each cross-validation iteration, all simulated samples from the training set are used to construct either the connectivity matrix $\mathbf{A}$ or the Laplacian matrix $\mathbf{L}$, from which the GFT matrix is computed as the eigendecomposition (see subsection 2.4). Using the GFT matrix, the input sample $\mathbf{X}_i$ is transformed to its GFT signal $\tilde{\mathbf{X}}_i$. The signal is further split into the 25 graph frequency signals. Lastly, a support vector machine classifier is trained on the 20 time spectral features extracted from each graph frequency signal using Welch's power spectral density method. The performance of each graph frequency can then be used to assess the quality of the spectral features in the graph frequency signal

The first step of the analysis consists of retrieving the graph structure from the training samples, for which we used the functional connectivity. Note that we need to use the same graph for all samples to keep the graph Fourier modes constant. Specifically, we computed the correlation matrix for each sample in the training set and subsequently averaged all matrices, yielding a common weighted adjacency matrix. Secondly, we computed the GFT-matrix and carried out the GFT, yielding $N_c = 25$ graph frequency signals. We used the weighted adjacency and the Laplacian matrix for the GFT in two separate experiments. Thirdly, we extracted spectral features from those signals for each sample. To this end, we computed the Welch power spectral density with a window of 32 time samples for each transformed signal and selected the first $N_f = 20$ power spectral densities as our features. Assuming a sampling frequency of 256 Hz, the extracted spectral densities cover frequencies in the range of 0-76 Hz at a resolution of 4 Hz. Lastly, we trained an SVM classifier separately for each graph frequency to classify the labelled samples, using only the 20 features calculated from each graph frequency signal. Specifically, we used a common SVM configuration consisting of a non-linear

radial basis function kernel with kernel coefficient $\gamma = 1/(N_f \mathrm{Var}(X))$, where X is the feature vector, and a regularisation strength of $C = 1$. Note that we did not optimise these hyperparameters for the following two reasons: Firstly, our study was focused on model comparison rather than performance optimisation. Secondly, the feature extraction in our model is fixed, meaning that hyperparameter configuration biases are unlikely. The baseline classification models, to be introduced in subsection 5.4, follow the same analysis steps and differ from this main model only in that their graph structure is modified.

5.3 Testing

Although our method allows to arbitrarily generate data, this process is still time-consuming: On the one hand, each time sample in a simulated participant is generated from the previous sample, restricting the use of vectorised computations. On the other hand, large numbers of samples on the order of tens of thousands are required to achieve the precision needed to compare individual graph frequencies. We therefore introduce a modified, perpetual version of cross-validation, as illustrated in Figure 6, which to the best of our knowledge has not been used before. The main goal is to successively generate unlimited pairs of training and test sets, with the constraint that each generated sample is only used once for testing. Initially, $N_s = 100$ samples are generated, divided evenly in condition 1 and condition 2. This data set is split into $k = 10$ folds. Nine of these are used for training, resulting in 90 training samples, while the remaining 10 samples are used for testing. For the subsequent iteration, data is generated to form a new fold with $N_s/k = 10$ samples, which is added to the training set, whereas the testing fold is shifted by one fold as illustrated in Figure 6. Overwriting the discarded fold with the new fold ensures that only $N_s = 100$ samples have to be stored in memory, or $N_s + N_s(1 - 1/k) = 190$ samples if the initially generated samples are reused during the last iterations. The final accuracy scores are averaged across all samples in the testing set and across all iterations.

The sample size of the training set is chosen to mimic the number of recorded samples in realistic neuroimaging data sets, such as the sparse data set mentioned in subsection 3.3. For each simulated graph structure, we ran 90 iterations, excluding the initial configuration, yielding 1000 generated samples, which is equal to the number of testing samples. We simulated 40 graph structures for each difficulty level, resulting in overall 120000 generated samples, each of which is a data matrix of size 25×2048. While matching the training size to real-life experiments ensures that the performance values of the simulated experiment are similar to those of the experiment, repeating these simulations arbitrarily often allows to increase the precision of the retrieved performance to the required level.

5.4 Baseline models

The principal goal of the baseline framework is to assess the gain of using GSP in our data classification procedure. Crucially, such a framework must be able to pinpoint the performance gain to the graph structure in the data, and not incidental gains due to other

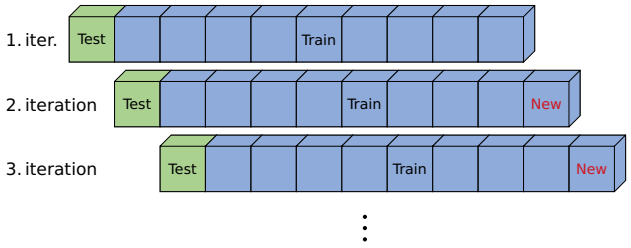

Fig. 6: Illustration of the perpetual cross-validation used during the analysis. The whole data set is split into $k = 10$ folds. One fold is assigned to be the testing set, while the remaining $k-1$ folds comprise the training set. For the subsequent iteration, a fold with newly generated data is added to the training set, while the testing fold is shifted by one fold

aspects of the methodology. To illustrate this point, we firstly discuss the baseline model in Huang *et al.* [28]: Here, surrogate signals are constructed for the baseline model by GFT-transforming the signals, then randomly flipping half of the signs of the Fourier components, before finally performing the inverse GFT. Their model is based on graph band-pass filtering the signals, however, the same operation was not performed on the surrogate signals. In other words, not only the graph, but also operations carried out on the signals differ between model and baseline. As a second illustration, Menorét *et al.* used graph-free models such as PCA as their baseline model [27], however, such baselines principally differ from the tested model in more ways than just the graph structure. In contrast, here we compose a set of three baseline models which are aimed at solely altering the graph structure, while preserving the processing steps used in the model.

The first baseline model is the *permuted nodes GSP model*, which uses the graph underlying the data, but randomly permutes the nodes of this graph. In this way, the graph Fourier modes have the same weights as the actual model, but at different locations. Therefore, the model distorts the graph structure, while retaining important characteristics of the modes such as the sparsity or diversity. Figure 7 illustrates the effect of this permutation on the modes in relation to the graph structure. While the top row (A-E) shows the lowest and highest modes on the graph, the bottom row (F-J) shows the modes for the permuted graph for the same graph frequencies. The graph Fourier modes clearly characterise the graph structure at lower graph frequencies for the unaltered graph. However, any such relation between the graph and the modes is distorted for the permuted graph. We expect this baseline model to perform equal to or worse than our classification model across all graph frequencies, given that the signal transformation in this model is not based on the spatial structure. Out of the three baseline models, this model is closest to the original classification model.

The second, related baseline model is the *random graph GSP model*, which uses GSP with a randomly generated graph. Specifically, we generated an undirected graph without self-loops and normally distributed weights $w_{ij} = w_{ji} \sim \mathcal{N}(1,1^2)$, such that roughly

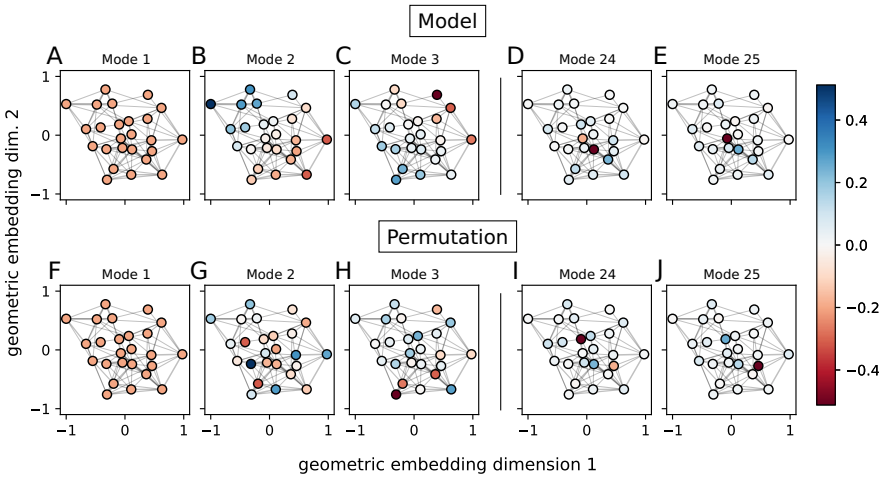

Fig. 7: Retrieved functional connectivity graph of a typical simulated training dataset, visualised as a geometric graph in two dimensions. The geometric positions were determined with the Fruchterman-Reingold force-directed algorithm using the NetworkX package in Python [86]. Node colourings represent the graph Fourier modes computed from the graph Laplacian. The top row (A-E) shows the three lowest (modes 1-3) and highest (modes 24 and 25) graph Fourier modes for this graph. While the low-frequency modes clearly exhibit a wave-like pattern covering the whole graph, a similar pattern is not visible for the higher modes. The bottom row (F-J) shows the same modes for a permuted graph, which is used in the permuted nodes baseline model. The permutation distorts any low-frequency ordering on the graph, while not changing the sparsity of the modes. Mode 1 (A,F) corresponds to the DC mode and is not affected by the permutation

84 % of the weights are positive. While this model transforms the data using GFT, these transformations are not based on the actual graph structure. Similarly to the first baseline model, we do not expect this model to outperform the original model for any of the graph frequencies due to the random nature of the transformation.

The third baseline model is the *single channel model*, which does not transform the data and is equivalent to conventional signal processing. It can also be viewed as GSP with the identity transform, given algebraically by the identity matrix $\mathbb{1}$. Following this notion, the "graph frequency signals" are given by the single channels and have no natural ordering. This trivially means that the mean performance is the same across all channels. Note also that the "graph Fourier modes" are vectors which are 1 at the channel index, and 0 elsewhere, which means that these modes are maximally sparse. This model is the only model that excludes the graph structure, making it an indispensable baseline model.

6 Results

Figure 8 illustrates the results of our analysis of the artificially generated data. Specifically, it shows the classification accuracy in dependence on the graph frequency of the transformed signal for the three simulated difficulty levels. The main model, shown in blue, is compared against the three baseline models. The accuracy scores in the figures are averaged across all testing samples, as described in subsection 5.3.

To begin with, the model performance increases steadily with increasing graph frequency, contrary to our expectations. This effect is even more pronounced in the adjacency-based model, which increases at a higher rate. The effect does not seem to depend on the difficulty level, albeit being more pronounced at the easier difficulty level. Crucially, the same effect is not present in the graph-based baseline models, whose performance is constant across all graph frequencies, apart from the graph DC frequency. While the two baselines outperform the model at lower frequencies, their accuracy is slightly lower than the model at higher frequencies. Differences between the two baselines are not significant and seem to be negligible. The accuracy of the single channel model, on the other hand, is trivially constant across all channels. As expected, the accuracy is mostly lower than that of all other models. The accuracy of the lowest graph frequency signal, namely the graph DC component, is by far the lowest for the three graph-based models. Note that the graph Fourier mode corresponding to the graph DC component is constant (Laplacian GFT) or near constant (adjacency GFT) across all entries and thereby highly dense.

Table 1 shows the mean accuracy at high graph frequencies of each model. Only the easiest simulated difficulty level was considered, as the effects are most pronounced at this level. To compute the mean, the five highest graph frequencies were averaged, given that all models have plateaued in this range. The adjacency-based main model leads the table significantly with 87.0% accuracy, outperforming the Laplacian-based counterpart by 1% point. The two graph-based baseline models have an accuracy of roughly 85%, irrespective of the graph representation used. The single channel baseline model trails all other models, performing more than 5% points lower than the best-performing model. These quantitative results indicate that our model significantly outperforms the baseline models at high graph frequencies.

7 Discussion

In the experiment, we observed the effect that the performance of the classification model increases with increasing frequency of the graph frequency signal. This suggests that projections on low-frequency graph Fourier modes, which mix nearby channels, do not amplify spectral features. Conversely, projections on high-frequency modes, which mix channels from distant parts of the graph, seem to accumulate spectral features more effectively. We hypothesise that the spectral features in closely connected channels are similar, because spectral features may spread to nearby locations during the data generation. Features from distant channels, on the other hand, are better isolated from one another, allowing them to contribute more diverse spectral features to the graph frequency signals. The graph-based baseline models are in line with our hypothesis. Both

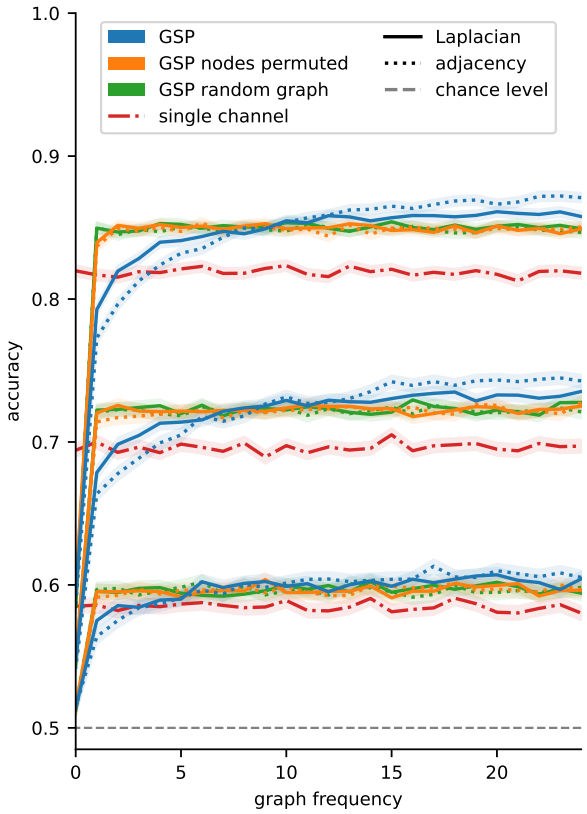

Fig. 8: Classification accuracy as a function of the graph frequency of the transformed signals. Three data sets with varying classification difficulty were simulated, whereby the difficulty was controlled by modifying the filter h as shown in Figure 4(C). Easy, medium, and difficult classification difficulties result in the high, medium, and low accuracies overlaid in the figure, respectively. The model is shown in blue, while the baseline models are shown in orange, green and red. Models using the graph Fourier transform with the Laplacian (adjacency) matrix are shown in solid (dotted) lines. Note that the single channel baseline model is not based on Graph Signal Processing (GSP) and thereby does not make use of the adjacency or the Laplacian matrix. This also means that its univariate signals are not ordered by graph frequency, but by graph node

models solely differ from the main model in that they use a graph unrelated to the data. This means that projections on all non-DC modes mix channels from random location, as shown in Figure 7. However, these patterns effectively correspond to mid-frequency modes on the actual graph across all non-DC graph frequencies, which means that their performance matches those of mid-frequency modes and does not change with graph frequency.

model	graph representation	mean accuracy [%] (5 highest frequ.)	CI [%]
GSP	Laplacian	86.0	[85.8,86.1]
	adjacency	**87.0**	[86.8,87.1]
GSP (nodes permuted)	Laplacian	84.9	[84.7,85.0]
	adjacency	84.9	[84.8,85.1]
GSP (random graph)	Laplacian	85.1	[84.9,85.2]
	adjacency	84.9	[84.7,85.0]
single channel	–	81.7	[81.6,81.9]

Table 1: Mean accuracy of all tested models for the easiest simulated difficulty level, averaged across the highest five graph frequencies. The model with the highest accuracy is the main model based on Graph Signal Processing (GSP) which uses the adjacency matrix for computing the graph Fourier transform, followed by the main model using the Laplacian matrix. The model with the lowest accuracy is the single channel baseline model, which is more than 5% points lower than the best-performing model

We further analyse effects due to the sparsity of the graph Fourier mode. To begin with, the eigenmodes in the single channel baseline model are simply the rows of the identity matrix and thereby maximally sparse. As expected, the performance of this model is lower than that of all graph-based models at higher graph frequencies, possibly because these modes do not source information from more than one channel, that is, their receptive field is minimal. The graph DC Fourier mode, on the other hand, is the mode which is maximally dense. Even though its receptive field covers all channels, its performance is nevertheless the lowest by far. We attribute the poor performance of highly dense modes to interference effects: If too many channels are mixed together, spectral features interfere with each other due to phase differences between the channels [87], thereby attenuating these spectral features. Therefore, the graph Fourier modes with the highest performance are those which are neither too sparse nor too dense. Importantly, the permutation model baseline allows us to exclude that the main effect observed in the model, namely the increase in performance with increasing graph frequency, is due to the sparsity of the modes: While the sparsity of the permuted eigenmodes are the same as those of the model, the main effect is not present in the permutation model.

Taken together, there seem to be two mechanisms at play that explain the performance of the graph frequency signals. Firstly, the performance depends on how many spectral features are accumulated during the transformation. While nearby channels share spectral features, distant channels have different spectral features. Generally, only transformations that are based on the actual graph can selectively mix nearby or distant channels. Secondly, the more channels are mixed together, that is, the denser the graph Fourier mode is, the more the channels interfere with each other. These two mechanisms explain our main result, namely the superior performance of high-frequency graph frequency signals, as follows: Projections on high-frequency graph Fourier modes mix

only few, but distant channels together; hence, they optimally accumulate diverse spectral features, while keeping interference effects in check.

Our observation that high-frequency graph frequency signals outperform both low-frequency graph frequency signals as well as unaltered single channels may prove useful for GSP-based applications such as dimensionality reduction or pruning of neural networks. However, note that the focus of our study was on understanding the specific role of individual graph frequencies. Hence, our study did not explicitly test and compare using a range of graph frequency signals for classification, which would almost certainly increase accuracy and thereby prove more useful for applications. An alternative approach to dimensionality reduction is to reduce the dimensionality of the data in the feature domain, instead of the signal domain, which has the advantage that interference effects are avoided. Our method may be repurposed for feature selection by limiting the spectral features to those of the most relevant channels, such as the channels that contribute most to the high-frequency graph frequency signals.

Interestingly, the adjacency-based model outperforms the Laplacian-based model slightly at higher graph frequencies. Following our interpretation of the results, this suggests that projections on high-frequency adjacency-based graph Fourier modes may sample more distant parts of the graph than their Laplacian-based counterparts. Generally, however, the choice of the graph representation for the GFT does not have a strong impact on the results, which is in line with previous observations by Huang *et al.* [28].

The superior performance of high-frequency over low-frequency graph frequency signals aligns with results obtained by Ménoret *et al.* on fMRI data, who observed the same effect for five out of the seven graphs they used, including a correlation-based graph [27]. However, their results are contradictory: They also found that this effect was reversed for a smooth graph optimised by minimising the TV with the algorithm by Kalofolias [78]; however, we showed in subsection 4.1 that TV-optimised graphs share crucial characteristics with correlation-based graphs. On the other hand, our main result contradicts results obtained by Ouahidi *et al.*, who found that keeping low-frequency graph frequency signals leads to the highest classification accuracy on fMRI data [36]. We attribute this discrepancy to differences in the spectral and spatial structure between neurophysiological signals such as EEG or MEG on the one hand, which we simulated in our experiment, and fMRI signals on the other. Firstly, EEG and MEG signals have arguably a richer spectral structure compared to fMRI signals, which means that interference effects play a much greater role in those signals. Secondly, fMRI data have a richer spatial structure: To illustrate this, Ouahidi *et al.* extracted a graph with 360 nodes from their data set, while our simulation was limited to a multivariate signal with 25 nodes. Both differences are likely to have an effect on the two proposed mechanisms explaining our results.

Our simulated data may differ from real EEG or MEG data in crucial ways. To begin with, the spatial structure of our simulated data is not very pronounced, i.d., the pairwise correlation-based functional connectivities does not vary much. In contrast, the spatial structure in real-life EEG data is much more pronounced, which may partly be the result of volume conduction effects. Additionally, the spectral structure may differ

from real data as well. While the spectral structure in our simulation is enforced by adding coloured noise, the spectral structure in real-life neuroimaging data may rely more strongly on the network, forming spatio-temporal structures not present in our simulation. Nevertheless, we argue that the generated data capture the essential characterisics of neurophysiological signals relevant for GSP, as shown in Figure 4. Hence, their analysis advocates the use of GSP for real-life neurophysiological EEG signals, and gives valuable insight into possible GSP mechanisms.

8 Conclusion

In this book chapter, we have derived central GSP concepts theoretically, reviewed selected applications of GSP, and investigated the meaning of graph frequency in GSP experimentally, with applications to neuroimaging in mind. In our view, one of the most significant challenges that GSP is currently facing is the ambiguity of the GFT, which is due to the number of choices for how to retrieve the graph (see subsection 3.1) and for which graph representation to use. To this end, we put the focus of this book chapter on understanding, validating, and interpreting GSP concepts, rather than performance optimisation.

The theoretical section of this book chapter aimed to give the reader a comprehensive introduction into GSP concepts, and how they are derived. Importantly, we juxtapose the alternative formulations of the GFT, and investigated similarities between these formulations, which may guide researchers in selecting and interpreting a specific formulation. Lastly, we explored links to established methods such as PCA, allowing the reader to place GSP in a wider context.

From an empirical perspective, the ambiguity of the GFT highlights the need for a thorough validation procedure. To this end, we have introduced a baseline testing framework for GSP in subsection 5.4, consisting of a set of overall three baseline models. For our experiment, we have developed a simulation algorithm to generate arbitrary amounts of labelled multivariate signals. Using the simulated data, we have classified the datasets by extracting spectral features from graph frequency signals and subsequently feeding the features from each graph frequency signal into an SVM classifier. Our results link the classification accuracy to the graph frequency, thereby enabling us to make inferences about the expressiveness of the spectral features at each graph frequency. Specifically, we obtained the result that high-frequency graph frequency signals outperform both low-frequency signals, as well as all baseline models. The accuracy gradient between low-frequency and high-frequency signals may be leveraged for dimensionality reduction or for pruning graph neural networks. Future work may be directed at testing dimensionality reduction for classification on real-life neurophysiological datasets. To use this method on EEG data, more sophisticated functional connectivity retrieval methods may be required, as well as complex source-space reconstruction methods.

Generating the data proved useful for two reasons: On the one hand, it allowed us to increase the precision of our results arbitrarily, laying bare the underlying GSP mechanisms. On the other hand, knowing how spectral features spread through the network narrowed down possible interpretations of the results. Likewise, the baseline framework

proved valuable, as it enabled us to pinpoint the measured effect to the graph structure in the data. Together, our experiment showcased how a combination of simulating the data and a strong validation procedure can propel our understanding of the mechanisms behind GSP. We believe that the mechanisms uncovered here may ultimately inform the design of GSP-based applications for neurophysiological signal processing.

Our study only tested simulated EEG-like multivariate signals. Further studies may focus on simulating more realistic EEG datasets, as wells as simulating other imaging modalitites such as fMRI, in order to find precise and modality-specific mechanisms in GSP. Lastly, future work may also focus on gaining an understanding of mechanisms underlying other GSP applications, such as graph filtering.

Acknowledgements

The authors would like to thank Dominik Klepl for important discussions.

References

[1] W. Bourgeois, A.-C. Romain, J. Nicolas, and R. M. Stuetz, "The use of sensor arrays for environmental monitoring: interests and limitations," *Journal of Environmental Monitoring*, vol. 5, no. 6, pp. 852–860, 2003.

[2] M. T. Vlaardingerbroek and J. A. Boer, *Magnetic resonance imaging: theory and practice*. Springer Science & Business Media, 2013.

[3] M. Tubaishat and S. Madria, "Sensor networks: an overview," *IEEE potentials*, vol. 22, no. 2, pp. 20–23, 2003.

[4] P. Sawangjai, S. Hompoonsup, P. Leelaarporn, S. Kongwudhikunakorn, and T. Wilaiprasitporn, "Consumer grade eeg measuring sensors as research tools: A review," *IEEE Sensors Journal*, vol. 20, no. 8, pp. 3996–4024, 2019.

[5] 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, 2019.

[6] R. Lindsey and L. Dahlman, "Climate change: Global temperature," *Climate. gov*, vol. 16, 2020.

[7] T. Kurokawa, T. Oki, and H. Nagao, "Multi-dimensional graph fourier transform," *arXiv preprint arXiv:1712.07811*, 2017.

[8] B. Upadhyaya, M. Kitamura, and T. Kerlin, "Multivariate signal analysis algorithms for process monitoring and parameter estimation in nuclear reactors," *Annals of Nuclear Energy*, vol. 7, no. 1, pp. 1–11, 1980.

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

[10] K. J. Miller, P. Shenoy, J. W. Miller, R. P. Rao, J. G. Ojemann *et al.*, "Real-time functional brain mapping using electrocorticography," *Neuroimage*, vol. 37, no. 2, pp. 504–507, 2007.

[11] J. Vrba and S. E. Robinson, "Signal processing in magnetoencephalography," *Methods*, vol. 25, no. 2, pp. 249–271, 2001.

[12] G. H. Glover, "Overview of functional magnetic resonance imaging," *Neurosurgery Clinics*, vol. 22, no. 2, pp. 133–139, 2011.

[13] D. P. Subha, P. K. Joseph, R. Acharya U, and C. M. Lim, "Eeg signal analysis: a survey," *Journal of medical systems*, vol. 34, pp. 195–212, 2010.

[14] P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, *Introduction to EEG-and speech-based emotion recognition*. Academic Press, 2016.

[15] A. Horvath, A. Szucs, G. Csukly, A. Sakovics, G. Stefanics, and A. Kamondi, "Eeg and erp biomarkers of alzheimer's disease: a critical review." *Frontiers in bioscience (Landmark edition)*, vol. 23, pp. 183–220, 2018.

[16] R. Lin, R.-G. Lee, C.-L. Tseng, H.-K. Zhou, C.-F. Chao, and J.-A. Jiang, "A new approach for identifying sleep apnea syndrome using wavelet transform and neural networks," *Biomedical Engineering: Applications, Basis and Communications*, vol. 18, no. 03, pp. 138–143, 2006.

[17] A. Al-Ezzi, N. Kamel, I. Faye, and E. Gunaseli, "Review of eeg, erp, and brain connectivity estimators as predictive biomarkers of social anxiety disorder," *Frontiers in psychology*, vol. 11, p. 730, 2020.

[18] G. Cheron, G. Petit, J. Cheron, A. Leroy, A. Cebolla, C. Cevallos, M. Petieau, T. Hoellinger, D. Zarka, A.-M. Clarinval *et al.*, "Brain oscillations in sport: toward eeg biomarkers of performance," *Frontiers in psychology*, vol. 7, p. 246, 2016.

[19] B. Swiderski, S. Osowski, and A. Rysz, "Lyapunov exponent of eeg signal for epileptic seizure characterization," *Proceedings of the 2005 European Conference on Circuit Theory and Design, 2005.*, vol. 2. IEEE, 2005, pp. II–153.

[20] S. J. Luck, *An introduction to the event-related potential technique*. MIT press, 2014.

[21] M. R. M. George, G. Potts, D. Kothman, L. Martin, and C. Mukundan, "Frontal deficits in alcoholism: an erp study," *Brain and cognition*, vol. 54, no. 3, pp. 245–247, 2004.

[22] W. S. Pritchard, "Cognitive event-related potential correlates of schizophrenia." *Psychological bulletin*, vol. 100, no. 1, p. 43, 1986.

[23] J. M. Peters, M. Taquet, C. Vega, S. S. Jeste, I. S. Fernández, J. Tan, C. A. Nelson, M. Sahin, and S. K. Warfield, "Brain functional networks in syndromic and non-syndromic autism: a graph theoretical study of eeg connectivity," *BMC medicine*, vol. 11, no. 1, pp. 1–16, 2013.

[24] F. de Vico Fallani, J. Richiardi, M. Chavez, and S. Achard, "Graph analysis of functional brain networks: practical issues in translational neuroscience," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1653, p. 20130521, 2014.

[25] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[26] 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, 2020.

[27] M. Ménoret, N. Farrugia, B. Pasdeloup, and V. Gripon, "Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction," *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2017, pp. 618–622.

[28] W. Huang, T. A. Bolton, J. D. Medaglia, D. S. Bassett, A. Ribeiro, and D. Van De Ville, "A graph signal processing perspective on functional brain imaging," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 868–885, 2018.

[29] S. Itani and D. Thanou, "A graph signal processing framework for the classification of temporal brain data," *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1180–1184.

[30] H. Behjat and M. Larsson, "Spectral characterization of functional mri data on voxel-resolution cortical graphs," *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2020, pp. 558–562.

[31] 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," *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 4549–4553.

[32] S. S. Saboksayr, G. Mateos, and M. Cetin, "Eeg-based emotion classification using graph signal processing," *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1065–1069.

[33] K. Glomb, J. R. Queralt, D. Pascucci, M. Defferrard, S. Tourbier, M. Carboni, M. Rubega, S. Vulliemoz, G. Plomp, and P. Hagmann, "Connectome spectral analysis to track eeg task dynamics on a subsecond scale," *NeuroImage*, vol. 221, p. 117137, 2020.

[34] 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, 2013.

[35] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 484–499, 2017.

[36] Y. El Ouahidi, H. Tessier, G. Lioi, N. Farrugia, B. Pasdeloup, and V. Gripon, "Pruning graph convolutional networks to select meaningful graph frequencies for fmri decoding," *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 937–941.

[37] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[38] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[39] 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, 2011.

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

[41] A. Gavili and X.-P. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Transactions on Signal Processing*, vol. 65, no. 23, pp. 6303–6318, 2017.

[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[43] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2017.

[44] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[45] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[46] J. Shi and J. M. Moura, "Graph signal processing: Dualizing gsp sampling in the vertex and spectral domains," *IEEE Transactions on Signal Processing*, vol. 70, pp. 2883–2898, 2022.

[47] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovacevic, "Signal denoising on graphs via graph filtering," *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 872–876.

[48] K. J. Friston, C. D. Frith, P. F. Liddle, and R. S. Frackowiak, "Functional connectivity: the principal-component analysis of large (pet) data sets," *Journal of Cerebral Blood Flow & Metabolism*, vol. 13, no. 1, pp. 5–14, 1993.

[49] F. He and Y. Yang, "Nonlinear system identification of neural systems from neurophysiological signals," *Neuroscience*, vol. 458, pp. 213–228, 2021.

[50] J. S. Damoiseaux and M. D. Greicius, "Greater than the sum of its parts: a review of studies combining structural connectivity and resting-state functional connectivity," *Brain structure and function*, vol. 213, no. 6, pp. 525–533, 2009.

[51] Y. Li and G. Mateos, "Learning similarity-preserving representations of brain structure-function coupling," *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 922–926.

[52] X.-J. Wang and H. Kennedy, "Brain structure and dynamics across scales: in search of rules," *Current opinion in neurobiology*, vol. 37, pp. 92–98, 2016.

[53] B. Horwitz, "The elusive concept of brain connectivity," *Neuroimage*, vol. 19, no. 2, pp. 466–470, 2003.

[54] M. Fraschini, S. M. Pani, L. Didaci, and G. L. Marcialis, "Robustness of functional connectivity metrics for eeg-based personal identification over task-induced intra-class and inter-class variations," *Pattern Recognition Letters*, vol. 125, pp. 49–54, 2019.

[55] T. Cattai, G. Scarano, M.-C. Corsi, F. D. V. Fallani, and S. Colonnese, "Eeg as signal on graph: a multilayer network model for bci applications," *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 932–936.

[56] O. Sporns, "Structure and function of complex brain networks," *Dialogues in clinical neuroscience*, 2022.

[57] R. Sedgewick, *Algorithms in java, part 5: Graph algorithms*. Addison-Wesley Professional, 2003.

[58] Q. Wang, M. A. Powell, A. Geisa, E. Bridgeford, and J. T. Vogelstein, "Why do networks have inhibitory/negative connections?" *arXiv preprint arXiv:2208.03211*, 2022.

[59] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, pp. 89–97, 2004.

[60] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi, A. Gramfort, B. Thirion, and G. Varoquaux, "Machine learning for neuroimaging with scikit-learn," *Frontiers in neuroinformatics*, vol. 8, p. 14, 2014.

[61] D. J. Blackburn, Y. Zhao, M. De Marco, S. M. Bell, F. He, H.-L. Wei, S. Lawrence, Z. C. Unwin, M. Blyth, J. Angel *et al.*, "A pilot study investigating a novel non-linear measure of eyes open versus eyes closed eeg synchronization in people with alzheimer's disease and healthy controls," *Brain sciences*, vol. 8, no. 7, p. 134, 2018.

[62] L. Rui, H. Nejati, and N.-M. Cheung, "Dimensionality reduction of brain imaging data using graph signal processing," *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1329–1333.

[63] G. Fracastoro, D. Thanou, and P. Frossard, "Graph transform optimization with application to image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 419–432,

2019.

[64] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3864–3868.

[65] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 446–475, 2018.

[66] J. S. Walker, "Fourier analysis and wavelet analysis," *Notices of the AMS*, vol. 44, no. 6, pp. 658–670, 1997.

[67] M. G. Preti and D. Van De Ville, "Decoupling of brain function from structure reveals regional behavioral specialization in humans," *Nature communications*, vol. 10, no. 1, p. 4747, 2019.

[68] A. Pentari, G. Tzagkarakis, K. Marias, and P. Tsakalides, "Graph denoising of impulsive eeg signals and the effect of their graph representation," *Biomedical Signal Processing and Control*, vol. 78, p. 103886, 2022.

[69] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[70] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional arma filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3496–3507, 2021.

[71] T. Song, W. Zheng, P. Song, and Z. Cui, "Eeg emotion recognition using dynamical graph convolutional neural networks," *IEEE Transactions on Affective Computing*, vol. 11, no. 3, pp. 532–541, 2018.

[72] P. Zhong, D. Wang, and C. Miao, "Eeg-based emotion recognition using regularized graph neural networks," *IEEE Transactions on Affective Computing*, vol. 13, no. 3, pp. 1290–1301, 2020.

[73] A. Demir, T. Koike-Akino, Y. Wang, M. Haruna, and D. Erdogmus, "Eeg-gnn: Graph neural networks for classification of electroencephalogram (eeg) signals," *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 1061–1067.

[74] S. Tang, J. A. Dunnmon, K. Saab, X. Zhang, Q. Huang, F. Dubost, D. L. Rubin, and C. Lee-Messer, "Self-supervised graph neural networks for improved electroencephalographic seizure analysis," *arXiv preprint arXiv:2104.08336*, 2021.

[75] G. Kalantar and A. Mohammadi, "Graph-based dimensionality reduction of eeg signals via functional clustering and total variation measure for bci systems," *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 4603–4606.

[76] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, 2019.

[77] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

[78] V. Kalofolias, "How to learn a graph from smooth signals," *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 920–929.

[79] 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, 2022.

[80] A. Einizade, M. Mozafari, M. Rezaei-Dastjerdehei, E. Aghdaei, A. M. Mijani, and S. H. Sardouie, "Detecting adhd children based on eeg signals using graph signal processing techniques," *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*. IEEE, 2020, pp. 264–270.

[81] J. Jeong, "Eeg dynamics in patients with alzheimer's disease," *Clinical neurophysiology*, vol. 115, no. 7, pp. 1490–1505, 2004.

[82] S. P. van den Broek, F. Reinders, M. Donderwinkel, and M. Peters, "Volume conduction effects in eeg and meg," *Electroencephalography and clinical neurophysiology*, vol. 106, no. 6, pp. 522–534, 1998.

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

[84] J.-M. Schoffelen and J. Gross, "Source connectivity analysis with meg and eeg," *Human brain mapping*, vol. 30, no. 6, pp. 1857–1865, 2009.

[85] M. Lai, M. Demuru, A. Hillebrand, and M. Fraschini, "A comparison between scalp-and source-reconstructed eeg networks," *Scientific reports*, vol. 8, no. 1, p. 12269, 2018.

[86] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[87] R. W. Thatcher, "Coherence, phase differences, phase shift, and phase lock in eeg/erp analyses," *Developmental neuropsychology*, vol. 37, no. 6, pp. 476–496, 2012.