# Comparing Spatial and Spectral Graph Filtering for Preprocessing Neurophysiological Signals

*S. Goerttler[1,2], F. He[1], M. Wu[2], D. Blackburn[3] and P. Sarrigiannis[4]*

1. Centre for Comp. Science and Math. Modelling, Coventry University, Coventry, UK
2. Inst. for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore
3. Department of Neuroscience, University of Sheffield, Sheffield, UK
4. Department of Neurophysiology, Royal Devon and Exeter NHS Foundation Trust, Exeter, UK
goerttlers@uni.coventry.ac.uk, ad0067@coventry.ac.uk, wumin@i2r.a-star.edu.sg,
d.blackburn@sheffield.ac.uk, ptolemaios.sarrigiannis@nhs.net

***Abstract*— Signals measured with multiple sensors simultaneously in time form multivariate signals and are commonly acquired in biomedical imaging. These temporal signals are generally not independent of each other, but exhibit a rich spatial structure. Graph filtering, either spatial or spectral, is a method that can leverage this spatial structure for various preprocessing tasks, such as graph denoising. Previous studies have focused on learning the parameters of spatial graph impulse response (GIR) filters, while neglecting spectral graph frequency response (GFR) filters, even though GFR filters offer unique advantages in terms of regularisation and interpretation. In this study, we therefore compare learning GIR filters and GFR filters as a trainable preprocessing step for two different neural networks on an Alzheimer's classification task. We tested both a functional connectivity graph as well as a geometric graph as the base of each filter type, and varied the localisation of the spatial filter. As expected, the retrieved shapes of the trained filters suggest that GFR filters can be interpreted in terms of their graph structure, while the same does not hold for GIR filters. Contrarily, however, we found that only the geometric, highly localised GIR filter outperforms the baseline significantly, surpassing it by 3.8 percentage points. These findings suggest that the observed performance boost of a trained localised GIR filter may in fact not be due to the graph structure. Instead, we hypothesise that this boost is caused by favourable algebraic properties of the filter matrix.**

***Keywords*— *electroencephalogram, graph signal processing, graph filtering, machine learning, Alzheimer's disease.***

## I. Introduction

Neurophysiological signals are often *multivariate signals*, as multiple sensors or channels acquire time series simultaneously. Notably, these time series are not independent of each other: Correlations arise due to the spatial proximity of the sensors or the causal connectivity between the regions measured by the sensors, forming a spatial structure in the multivariate signal [1]. While this structure is commonly disregarded in the spectral analysis of those signals, or analysed separately, *graph signal processing* (GSP) has gained traction over recent years as a framework that promises to integrate the spatial structure into the signal analysis [2, 3]. GSP methods have numerous applications for biomedical imaging, such as magnetic resonance imaging or electroencephalography (EEG) and include *graph denoising* [4], *graph spectral filtering* [5], or the analysis of *graph frequency signals* [6]. Common to all methods listed here is that they linearly transform the multivariate signals across space by means of the *graph Fourier transform*, which is based on the spatial structure of the signals. These tools can also be incorporated in larger models for various subtasks, such as dimensionality reduction [6] or graph pooling [7].

This study investigates graph filtering as a tool to preprocess input data samples for downstream tasks. Two main graph filtering approaches are compared, namely spatial and spectral graph filtering. Graph spatial filtering is used extensively in graph neural networks and is based on the notion of localisation [8]. On the other hand, graph spectral filtering is useful for tasks such as graph denoising, which uses as low-pass graph filter to reduce the *graph total variation* of an input multivariate signal [5]. Setting the graph filter parameters manually is arbitrary and may not result in optimal filtering. A more general solution is to learn the graph filter synchronous with the downstream task from the data. This approach is not new: Previous research explored learning the parameters of a localised GIR filter [8] or of a GFR filter [9] for use in a graph neural network layer. The former approach, called the *ChebNet*, has found more widespread use as it is more localised and computationally more efficient [10]. However, advantages of GIR over GFR filters such as localisation, computational efficiency, or parameter reduction may only hold for large, sparse graphs. Therefore, they may not play a role for neurophysiological signals, where graphs are typically of the order of 100 nodes or less. On the other hand, GFR filters have the advantage that filters can be interpreted in terms of band-pass filters, which also facilitates their regularisation.

Our approach in this study is to systematically compare various GIR and GFR filters for data preprocessing, with the intention to test their applicability as well as to gain more insight into the parameters optimisation. To this end, we implemented the two filter types as a trainable preprocessing step for an EEG classification task. We tested a patient-dependent functional connectivity graph

as well as a geometric graph as the base for each filter, while we vary the degree of localisation for the GIR filter. Further, we apply each graph filters on two different downstream EEG classification network architectures. While the first architecture uses engineered low-frequency *spectral features* in the time domain, the second architecture learns the feature extraction from the data by means of a 1D *convolutional neural network* (CNN). Computing the classification accuracy relative to the graph type for each filter type allows to compare the usefulness of the filter. We find that only the geometric, localised GIR filter improves the classification accuracy of the baseline models considerably by 3.8%. On the other hand, the retrieved filter shapes give insight into how the filters are learnt, by revealing whether filters are learnt consistently and how the learnt filters differ across different network architectures. We show that the learnt parameters of our best-performing filter are inconsistent, suggesting that the graph structure is not crucial for performance.

## II. Theory

Graph filtering is a filtering method in the graph domain in analogy to classical filtering in the temporal domain [2]. Similarly to the classical case, graph filters can both be constructed in terms of their graph frequency response (GFR filter) or their graph impulse response (GIR filter). Graph filtering can be employed on multivariate signals with a spatial structure, which is algebraically given by the graph adjacency matrix $\mathbf{A}$. The spatial structure can also be described by the related *Laplacian matrix* $\mathbf{L}$, which can be computed from $\mathbf{A}$ as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathrm{diag}(\mathbf{A1})$ is the degree matrix of $\mathbf{A}$ and $\mathbf{1}$ is a column vector of ones. We can interpret the adjacency matrix as a *graph shift operator* and the Laplacian matrix as a *graph difference operator*. The eigenmodes of this graph difference operator, i.e. the sorted and normalised eigenvectors of $\mathbf{L}$, constitute the rows of the orthonormal Graph Fourier Transform matrix $\mathbf{GFT}$ (for more details, see Ortega et al. [3]). Its inverse is given by the transposed matrix $\mathbf{GFT}^\top$.

A spectral GFR filter $\mathbf{H}^{(\mathrm{FR})}$ operates on the spatial graph signals $\mathbf{x}_j \in \mathbb{R}^{N_c}$ measured at time step $t_j$, where $N_c$ is the number of sensors or channels. It transforms each spatial signal $\mathbf{x}_j \in \mathbb{R}^{N_c}$ into the graph spectral domain, linearly filters out the spectral components, and transforms the signal back into the signal domain:

$$\mathbf{H}^{(\mathrm{FR})}\mathbf{x}_j = \mathbf{GFT}^\top \begin{pmatrix} h_1 & 0 & \cdots & 0 \\ 0 & h_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_n \end{pmatrix} \mathbf{GFT}\,\mathbf{x}_j = \tilde{\mathbf{x}}_j, \tag{1}$$

where $\tilde{\mathbf{x}}_j$ is the filtered spatial signal. The graph filter is hereby characterised by the filter parameters $h_i$, with $0 \leq h_i \leq 1$. The parameters $h_i$ with $i < N_c/2$ filter out low graph frequencies corresponding to similarities between the channels, while the remaining parameters filter out high graph frequencies corresponding to differences. To give an example, a GFR filter with all high-frequency parameters set to zero filters out channel differences and thereby noise; accordingly, the application of this particular graph filter is also called graph denoising.

On the other hand, the spatial GIR filter is constructed as a polynomial of either the graph shift operator [2], in analogy to the classical case, or as a polynomial of the graph difference operator. In the former case, the degree of the polynomial term indicates how often the graph operator "shifts" the signal along the graph, meaning that lower-order terms correspond to localised operations on the graph, while higher-order terms are more global. We limit our study to the more common construction with the graph difference operator [8]. The filter coefficients $\theta_i$ are typically regularised as follows: Firstly, the Laplacian matrix is normalised by dividing it by its largest eigenvalue, $\tilde{\mathbf{L}} = \mathbf{L}/\lambda_{\max}$. Secondly, Chebyshev polynomials $T_i$ are used, further adjusting the coefficients. Thirdly, only filter coefficients up to a cutoff $k$ are considered, where lower $k$ correspond to higher spatial localisation. Hence, the GIR filter with regularised coefficients acting on a graph signal $\mathbf{x}_j$ is given by:

$$\mathbf{H}^{(\mathrm{IR})}\mathbf{x}_j = \left( \sum_{i=0}^{k-1} \theta_i T_i(\tilde{\mathbf{L}}) \right) \mathbf{x}_j = \tilde{\mathbf{x}}_j. \tag{2}$$

Importantly, any GFR filter can be expressed as a GIR filter without cutoff, and vice versa, that is, there is a one-to-one mapping between the $N_c$ filter parameters $h_i$ and $\theta_i$. The principal difference between the two filter types lies predominantly in how the parameters are constructed or interpreted. Given the full multivariate signal $\mathbf{X} \in \mathbb{R}^{N_c \times N_t}$, the graph-filtered multivariate signal is computed as $\tilde{\mathbf{X}} = \mathbf{H}^{(\mathrm{FR/IR})}\mathbf{X}$. Here, $N_t$ is the number of time steps, while the superscript (FR/IR) indicates the choice of the filter type.

## III. Methods

### III-A. Data Set

The data used in this study was acquired from 20 Alzheimer's disease patients and 20 healthy controls with a 23-channel EEG system at a sampling rate of 2048 Hz, which we further downsampled by a factor of 10 to a sampling rate of 204.8 Hz. For every patient except one, 3 sections of roughly 12 seconds were selected by a clinician, resulting in 119 samples. Each sample can be windowed into smaller subsamples of length 6 or 3 seconds to further increase the sample size to 238 or 476, respectively. Note that the pseudo-replicated samples corresponding to one patient are not independent of each other. To cancel out volume

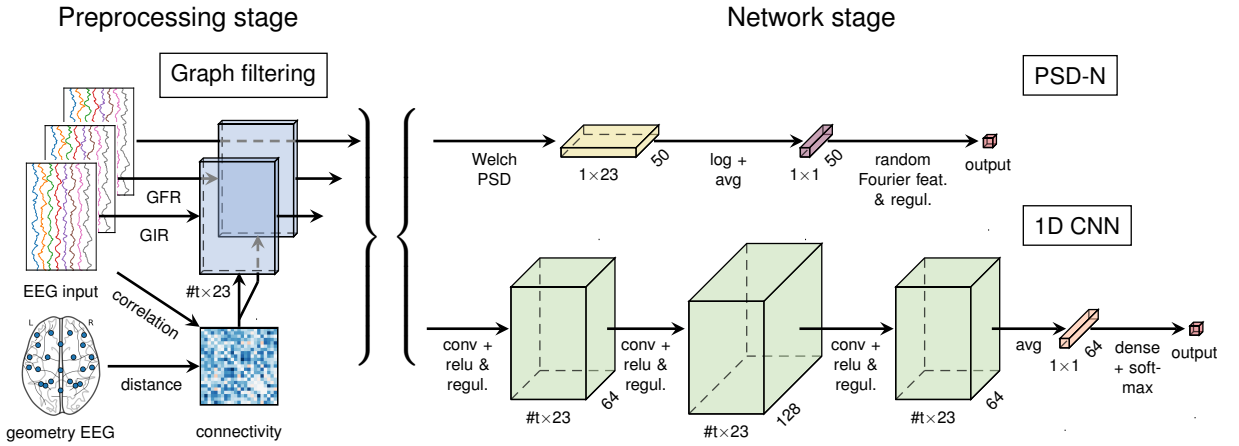


Figure 1. Illustration of the graph filtering preprocessing stage in combination with the two network architectures. Either no graph filter (back), a graph spectral filter (centre) or a graph impulse response filter (front) is applied to the input EEG sample, whereby the filter parameters are trainable. The graph filter is based on the connectivity matrix, which can be computed either from the input as the pairwise correlation, or from the geometry of the EEG electrodes. In the network stage, the first network (upper, PSD-N) extracts power spectral densities and interprets them using a random Fourier features classifier. The second network (lower) is a 1D CNN consisting of three convolutional layers

conduction artifacts, bipolar channels were created. Additional details regarding the dataset can be found in Blackburn et al. [11].

### *III-B. Graph Extraction*

We extracted two types of graphs for the graph filtering. The first type models the *functional connectivity* and is estimated directly from the data, while the second type models the geometric structure and is based on the geometric location of the EEG electrodes. To extract the functional connectivity graph, we compute one graph $\mathbf{A}_p^{(\mathrm{FC})}$ for each patient. Specifically, we compute the entry $a_{ij}$ of $\mathbf{A}_p^{(\mathrm{FC})}$ as the Pearson correlation for the channel pair $(i,j)$ with $i \neq j$, incorporating the data of all sampled sections of patient $p$. The diagonal entries $a_{ij}$ are set to zero. Note that the symmetry of the Pearson correlation entails the symmetry of the adjacency matrix, $a_{ij} = a_{ji}$. Note also that the correlation between two channels can be negative, which means that $\mathbf{A}_p^{(\mathrm{FC})}$ is not necessarily positive semi-definite. As a consequence, the eigenvalues of the Laplacian matrix computed from the adjacency matrix can be negative.

Secondly, to compute the geometric graph $\mathbf{A}^{(\mathrm{G})}$, we use the midpoint $\mathbf{r}_i$ between the theoretical geometric locations of the EEG electrodes for each bipolar channel $i$. The connectivity $a_{ij} \in \mathbf{A}^{(\mathrm{G})}$ between two bipolar channels $i$ and $j$ is then computed as

$$a_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_j\|_2^2}{2\rho^2}\right), & \text{if } i \neq j \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $\|\cdot\|_2$ is the L2 norm, meaning that $\|\mathbf{r}_i - \mathbf{r}_j\|_2$ is the Euclidean distance between the two bipolar channels. We set the scaling parameter $\rho$ to 50 mm. Note that as opposed to the functional connectivity graph, each participant is assigned the same graph.

### *III-C. Trainable Graph Filter Preprocessing*

We develop graph filtering as a trainable preprocessing step preceding the EEG classification task, as shown in Figure 1 on the left-hand side. The graph filter is either a GFR or a GIR filter; no graph filter serves as a baseline model. For the GIR filter, we test cutoff values of $k = 2$, $k = 12$, and $k = N_c = 23$. This means that we tested 4 filter types and 1 baseline altogether. Each filter type can be based on either the functional connectivity graph or the geometric graph, resulting in 9 filter configurations.

The graph spectral filtering operation is given by a multiplication of the multivariate signal $\mathbf{X}_p$ of participant $p$ with the filter matrix $\mathbf{H}_p^{(\mathrm{FR})}$:

$$\tilde{\mathbf{X}}_p = \mathbf{H}_p^{(\mathrm{FR})}\mathbf{X}_p = \mathbf{GFT}_p^{\top}\mathrm{diag}(h_1,\ldots,h_n)\,\mathbf{GFT}_p\,\mathbf{X}_p. \tag{4}$$

The matrix $\mathbf{GFT}_p$ is based on the connectivity graph, and depends on the participant $p$ in the case of the functional connectivity graph $\mathbf{A}_p^{(\mathrm{FC})}$. Note that the trainable GFR filter coefficients $h_i$ universally apply to all patients. We initialise the filter parameters $h_i$ as 0.5 with added noise and restrict them to the range $[0,1]$ by training the latent variables $\tilde{h}_i = S^{-1}(h_i)$, where $S$ is the logistic sigmoid function. This regularisation technique significantly limits the possible graph filters to a much smaller parameter space of viable filters.

Table 1. Average testing accuracy over five runs for each model configuration. Bold font marks the best performing filter for filters with the same configuration, while grey values mark values below the baseline. The results indicate that model configurations using a GIR filter with a low cutoff $k$ generally outperform both other filter types and the base model

| **testing accuracy [%]** | | PSD-N | | | | | 1D CNN | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | base | GFR | GIR | | | base | GFR | GIR | | |
| graph retrieval | length [s] | | | $k=2$ | $k=12$ | $k=23$ | | | $k=2$ | $k=12$ | $k=23$ |
| Pearson correlation | 3 | 82.0 | 82.1 | **82.7** | 78.3 | 73.5 | **61.2** | 60.6 | 60.9 | 51.1 | 45.8 |
| | 6 | 82.8 | 81.9 | **83.4** | 79.4 | 71.8 | **60.6** | 58.3 | 56.3 | 49.2 | 46.3 |
| | 12 | 83.5 | 83.7 | **84.9** | 80.3 | 70.3 | 59.0 | **60.5** | 59.0 | 48.1 | 47.7 |
| geometric distance | 3 | 82.0 | 81.0 | 82.3 | 81.5 | **82.5** | 61.2 | 63.8 | 71.3 | **71.4** | 69.4 |
| | 6 | 82.8 | 82.4 | **84.1** | 82.5 | 83.3 | 60.6 | 61.0 | **67.6** | 64.4 | 60.6 |
| | 12 | 83.5 | 83.0 | **84.0** | 83.4 | 83.0 | 59.0 | 60.0 | **62.5** | 61.2 | 60.2 |

Table 2. Linear model fit for all testing accuracy values. Standard errors and $p$-values were computed using an analysis of variance (ANOVA). Significance levels are indicated in the footnote. The upper section displays the intercept testing accuracy for each network for the baseline. The lower section displays the percentage point relative to this baseline for each filter configuration. Positive coefficients indicate that the filter outperforms the baseline. The highest filter coefficient, marked in bold, indicates a significant average accuracy gain of 3.8 percentage points as compared to baseline (no graph filtering). The lowest filter coefficient is marked in grey

| network | **testing accuracy coefficient [%]** | | | |
|---|---|---|---|---|
| PSD-N | 82.5(7)*** | | | |
| 1D CNN | 60.5(7)*** | | | |
| | GFR | GIR | | |
| graph retrieval | | $k=2$ | $k=12$ | $k=23$ |
| Pearson corr. | −0.3(9) | −0.3(9) | −7.1(9)*** | −12.3(9)*** |
| geom. distance | 0.4(9) | **3.8(9)*** ** | 2.5(9)** | 1.7(9). |

Standard errors in parentheses
. $p<0.1$, * $p<0.05$, ** $p<0.01$, *** $p<0.001$

Conversely, the GIR filter is given by

$$\tilde{\mathbf{X}}_p = \left( \sum_{i=0}^{k-1} \theta_i T_i(\tilde{\mathbf{L}}_p) \right) \mathbf{X}_p = \mathbf{H}_p^{(\mathrm{IR})} \mathbf{X}_p, \qquad (5)$$

where $\tilde{\mathbf{L}}_p$ is based on either the patient-dependent functional connectivity graph $\mathbf{A}_p^{(\mathrm{FC})}$ or the geometric graph $\mathbf{A}^{(\mathrm{G})}$. Here, the parameters $\theta_i$ are the trainable parameters, which we initialise as zero with added noise.

### *III-D. Network Architectures*

The preprocessed data is passed to the network stage, for which we tested two network architectures. The two networks are shown in Figure 1 on the right-hand side. The first network (PSD-N) interprets spectral features in the time domain, namely power spectral densities retrieved using Welch's method, thereby implementing a common strategy to classify Alzheimer's disease [12]. As the spectral features do not need to be learned, the network has only about 4,000 trainable parameters. Specifically, 50 power spectral densities are extracted in the range from 0.5 Hz to 25 Hz. The logarithmised features are averaged across all channels and interpreted by a random Fourier features layer with an output dimension of 2048, followed by a final dense layer. We apply a dropout of 50% after the random Fourier features layer for regularisation. Furthermore, we use a hinge loss as the loss function, and optimise our objective function using the Adam optimiser. The random Fourier features layer in combination with the hinge loss substitutes a support vector machine classifier [13]. Unlike the support vector machine, however, the random Fourier features layer can be trained and allows gradients to be passed through the network stage back to the preprocessing stage. Note that computing power spectral densities uses the square function, which is not a convex operation and may lead to additional local minima in the objective function.

The second network architecture is a 1D CNN adapted from Wang et al. [14]. Unlike the PSD-N, the 1D CNN learns the feature extraction, which significantly increases the number of trainable parameters to roughly 70,000. Each of the three network layers consists of a convolutional layer and a ReLU activation function in combination with a regularisation scheme. The convolution is only carried out along the time with respective filters of size 8, 5 and 3. Due to the limited size of our dataset, we halved the number of filters in the three layers as opposed to the original network to 64, 128 and 64, respectively. We applied 50% dropout after the activation function to regularise the network. The averaged features of the last layer are interpreted by a dense layer and converted to probabilities using the softmax function. As in the first network architecture, the gradients were optimised using the Adam optimiser.

Both networks serve as baselines and allow to measure the effect of graph filter preprocessing on the performance, relative to graph type and graph retrieval method. Suitable hyperparameters of both baseline network architectures were determined through preliminary

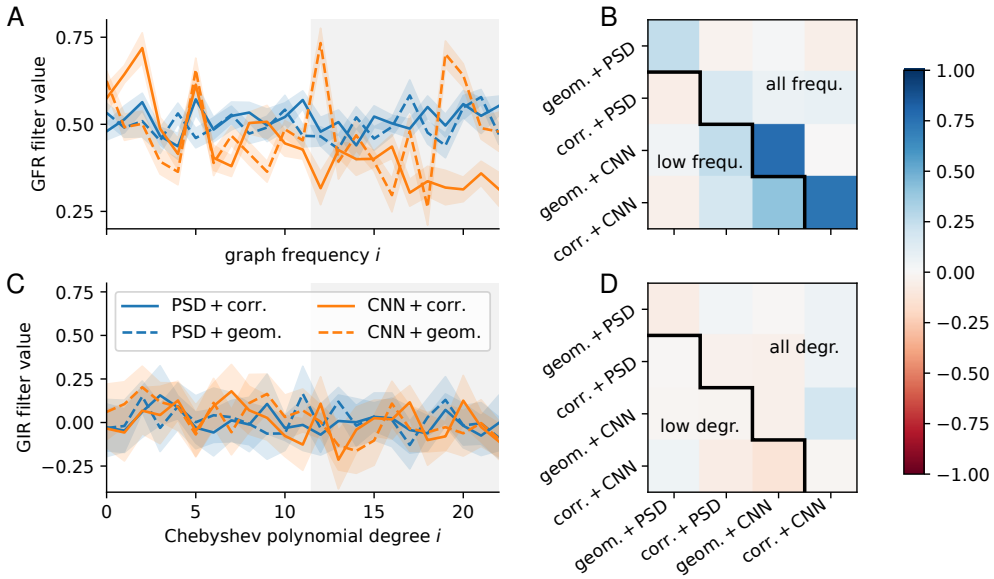


Figure 2. (A,C) Mean shape for GFR filter (A) and GIR filter with $k = N_c$ (C), averaged across sample window length and repeats. Shaded regions indicate the standard error. Values outside the grey area indicate low frequency (A) or low polynomial degree (C). (B, D) Correlation of filters between the configurations (off-diagonal) or within the same configuration (diagonal). Squares below the diagonal indicate filter correlations restricted to low frequencies or low polynomial degrees, corresponding to values outside the grey area in (A) and (C). Mean filter shapes in (A) align slightly at lower frequencies, in line with correlations of up to 0.40 between 1D CNN configurations in (B). Auto-correlations for the 1D CNN are particularly high at 0.76 (geom.) and 0.73 (corr.). Filter shapes in (A) corresponding to the 1D CNN exhibit a slight downward trend. GIR filter values in (C) do not deviate strongly from the initialisation value of 0.50. Correlations of the learnt GIR filter shapes in (D) are negligible

testing. As this study is primarily focused on relative performance differences between model configurations with the same network architecture, the hyperparameters were not further optimised.

### *III-E. Network Combinations*

Combining the 9 filter configurations of the preprocessing stage with the 2 networks results in overall 18 models. Furthermore, we tested input sample lengths of 3, 6 and 12 seconds for each of the models, amounting to 54 model configurations overall. Each model configuration was run 5 times to limit network initialisation effects.

### *III-F. Training and Testing*

All models were trained for 100 epochs at a learning rate of 0.001. Due to the relatively small size of the dataset, the batch size was set to 8. We employed 10-fold cross validation, meaning that 90% of the dataset were used for training and the remaining 10% for testing in each iteration. Samples of each patient were strictly kept in one fold to avoid data leakage. Specifically, the samples of 2 patients of each condition were assigned to the testing set, while all other samples were used for training.

## IV. Results

Table 1 shows the testing accuracy for each model configuration, averaged over 5 runs. In order to retrieve the accuracy enhancement for each filter, relative to the baseline, we fitted a linear model to the data and performed an analysis of variance (ANOVA) to retrieve standard errors and significance levels, as shown in table 2. The model includes one coefficient for each combination of filter and graph retrieval type, and one coefficient for each network. The results indicate that the PSD-N achieved a base accuracy of 82.5%, which is comparable to typical Alzheimer's disease classification accuracies between 80% and up to 90% [15]. The 1D CNN only achieved a base accuracy of 60.5%, thereby scoring roughly 22 percentage points lower than the PSD-N. While the geometric GFR filter outperformed the correlation GFR filter by roughly 0.7 percentage point, neither filter deviates significantly from the baseline. The geometric GIR filter with $k = 2$, however, outperformed the baseline by 3.8 percentage points with high significance. Both increasing the number of filter parameters or basing the filter on the correlation impaired the performance. The accuracy of the correlation GIR filter without cutoff is 12.3 percentage points lower than that of the baseline.

Figure 2 shows the learned GFR (A) and the GIR filter with $k = 23$ (C) visually, averaged over the window length and repeats, along with the respective mean correlation of all filters within (diagonal) or between configurations in (B) and (D). In (A), the CNN GFR filters may function as a low-pass filter, indicative of graph

denoising. The PSD-N, on the other hand, considers only low-frequency features; correspondingly, the GFR filters are neither low-pass nor high-pass filters. Further, the filters appear to correlate at lower graph frequencies. This intuition is supported by the correlation matrix in (B), which shows that the shape correlation for CNN filters across graph types is 0.40, and even as high as 0.24 across configurations with different graph type and network. The correlation matrix also shows that filter shapes are consistent within the same configuration, with values of up to 0.76.

In (C), the learnt GIR filter shapes with $k = 23$ show no significant deviation from its initialisation values, indicating that the filter optimisation may have failed. The respective correlation matrix in (D) demonstrates that the filter shapes are not consistently learnt, even not within the same configuration.

## V. Discussion

In this study, we found that a geometric, highly localised GIR preprocessing filter improved the accuracy across two EEG classification networks highly significantly by 3.8 percentage points. This result demonstrates the usefulness of graph filter preprocessing for neurophysiological classification specifically, and multivariate signal classification more generally. In line with Kipf and Welling [16], we observed that increasing the number of filter parameters reduces the accuracy. Our analysis of the filter shapes suggests that too many parameters hinder the optimisation of the graph filter, possibly because the additional parameters add more local minima to the objective function.

We further showed that the learnt GFR filters correlate with each other at low graph frequencies and could be interpreted in terms of their function, such as graph denoising, for the network. This indicates that GFR filters are better regularised and add fewer local minima to the objective function; however, these advantages did not translate into performance gains. Learnt GIR filters, on the other hand, are not consistent across configurations and repeats, suggesting that they poorly reflect the graph structure. We propose that their usefulness for preprocessing stems from algebraically suitable properties of the filter instead. Future studies may verify our hypothesis by basing the GIR filter on arbitrary graph structures and investigating the algebraic properties of the resulting filters.

## Acknowledgements

The EEG data was funded by a grant from the Alzheimer's Research UK (ARUK-PPG20114B-25). The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.

## References

[1] E. Pereda, R. Q. Quiroga, and J. Bhattacharya, "Nonlinear multivariate analysis of neurophysiological signals," *Progress in neurobiology*, vol. 77, no. 1-2, pp. 1–37, 2005.

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

[3] 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.

[4] A. Pentari, G. Tzagkarakis, K. Marias, and P. Tsakalides, "Graph-based denoising of eeg signals in impulsive environments," *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1095–1099.

[5] 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.

[6] 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.

[7] Y. Ma, S. Wang, C. C. Aggarwal, and J. Tang, "Graph convolutional networks with eigenpooling," *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 723–731.

[8] 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.

[9] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[10] 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.

[11] 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.

[12] R. Cassani, M. Estarellas, R. San-Martin, F. J. Fraga, and T. H. Falk, "Systematic review on resting-state eeg for alzheimer's disease diagnosis and progression assessment," *Disease markers*, vol. 2018, 2018.

[13] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.

[14] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1578–1585.

[15] R. Cassani, T. H. Falk, F. J. Fraga, M. Cecchi, D. K. Moore, and R. Anghinah, "Towards automated electroencephalography-based alzheimer's disease diagnosis using portable low-density devices," *Biomedical Signal Processing and Control*, vol. 33, pp. 261–271, 2017.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.