Medical Image Analysis 79 (2022) 102471

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

A deep graph neural network architecture for modelling spatio-temporal dynamics in resting-state functional MRI data

Tiago Azevedo[a,*], Alexander Campbell[a], Rafael Romero-Garcia[b,h], Luca Passamonti[c,d], Richard A.I. Bethlehem[b,e], Pietro Liò[a], Nicola Toschi[f,g]

[a] Department of Computer Science, University of Cambridge, UK
[b] Brain Mapping Unit, Department of Psychiatry, University of Cambridge, UK
[c] Department of Clinical Neurosciences, University of Cambridge, UK
[d] Istituto di Bioimmagini e Fisiologia Molecolare (IBFM), Consiglio Nazionale delle Ricerche (CNR), Milano, Segrate, Italy
[e] Autism Research Centre, Department of Psychiatry, University of Cambridge, UK
[f] Department of Biomedicine and Prevention, University of Rome "Tor Vergata", Italy
[g] A. A. Martinos Center for Biomedical Imaging, Massachusetts General Hospital and Harvard Medical School, US
[h] Instituto de Biomedicina de Sevilla (IBiS) HUVR/CSIC/Universidad de Sevilla/ CIBERSAM, Dpto. de Fisiología Médica y Biofísica

ARTICLE INFO

Article history:
Received 8 November 2020
Revised 11 April 2022
Accepted 2 May 2022
Available online 7 May 2022

Keywords:
Deep learning
Graph neural networks
UK Biobank
Time series
Temporal convolutional network
Rs-fMRI
Spatio-temporal dynamics

ABSTRACT

Resting-state functional magnetic resonance imaging (rs-fMRI) has been successfully employed to understand the organisation of the human brain. Typically, the brain is parcellated into regions of interest (ROIs) and modelled as a graph where each ROI represents a node and association measures between ROI-specific blood-oxygen-level-dependent (BOLD) time series are edges. Recently, graph neural networks (GNNs) have seen a surge in popularity due to their success in modelling unstructured relational data. The latest developments with GNNs, however, have not yet been fully exploited for the analysis of rs-fMRI data, particularly with regards to its spatio-temporal dynamics. In this paper, we present a novel deep neural network architecture which combines both GNNs and temporal convolutional networks (TCNs) in order to learn from both the spatial and temporal components of rs-fMRI data in an end-to-end fashion. In particular, this corresponds to intra-feature learning (i.e., learning temporal dynamics with TCNs) as well as inter-feature learning (i.e., leveraging interactions between ROI-wise dynamics with GNNs). We evaluate our model with an ablation study using 35,159 samples from the UK Biobank rs-fMRI database, as well as in the smaller Human Connectome Project (HCP) dataset, both in a unimodal and in a multimodal fashion. We also demonstrate that out architecture contains explainability-related features which easily map to realistic neurobiological insights. We suggest that this model could lay the groundwork for future deep learning architectures focused on leveraging the inherently and inextricably spatio-temporal nature of rs-fMRI data.

© 2022 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Resting-state functional magnetic resonance imaging (rs-fMRI) is one of the most commonly used, noninvasive imaging techniques employed to gain insight into human brain function. The use of rs-fMRI data has proven extremely useful as an investigative tool in neuroscience and, to some extent, as a biomarker of brain disease diagnosis and progression (Fornito et al., 2015). Typical use of rs-fMRI data often involves using graph-theoretical measures (such as centrality measures and community structures) to summarise high-dimensional, whole-brain data for use in downstream tasks. As part of this process, it is common practice to reduce the dimensionality of the data in one of three main ways: (1) by collapsing the temporal dimension (e.g., into connectivity matrices between brain regions), (2) by reducing the spatial dimension (e.g., by aggregating voxelwise signals into predefined brain regions) (Wang et al., 2019), and (3) by employing approaches that collapse both the temporal and spatial dimensions (e.g., in independent component analyses) (Beckmann et al., 2005). These feature engineering steps are performed mostly due to the considerable volume of data in a typical rs-fMRI dataset and its relatively low signal-to-noise ratio (Smith and Nichols, 2018).

* Corresponding author.
E-mail addresses: tiago.azevedo@cst.cam.ac.uk (T. Azevedo), pietro.lio@cst.cam.ac.uk (P. Liò), toschi@med.uniroma2.it (N. Toschi).

https://doi.org/10.1016/j.media.2022.102471
1361-8415/© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Although computationally beneficial, such dimensionality reduction steps inevitably involve disregarding large amounts of information which can be useful depending on the analysis task. For instance, collapsing the temporal dimension of rs-fMRI data reduces the brain to a static volume where the interactions between different brain regions are fixed over time. This stands in contrast to a growing body of research which shows that functional connectivity in the brain is dynamic and constantly changes over time (Avena-Koenigsberger et al., 2017; Liao et al., 2017). As another example, association measures most commonly used are still based on linear models, while it is well known that neuromonitoring data and brain signals, in particular, interact nonlinearly (Duggento et al., 2018; Goelman et al., 2018).

To overcome such limitations, a different approach to the analysis of rs-fMRI data would be to devise a model that is able to combine both feature engineering and the learning of a low-dimensional representation of the brain's functional activity. Such a model would need to be able to accommodate both the spatial and temporal complexities of rs-fMRI data. To date, deep learning architectures have had great success at leveraging specific inductive biases from complex high-dimensional data. Convolutional neural networks (CNNs), for instance, are extremely effective at extracting shared spatial features such as corners and edges from grid-like data (e.g., 2D and 3D images). These features can then be combined into more complex concepts deeper within the network architecture (Spasov et al., 2019). Recurrent neural networks (RNNs), on the other hand, are able to learn features from data that are temporally organised as a sequence of steps (Duggento et al., 2019; Dvornek et al., 2017). In contrast to both CNNs and RNNs, graph neural networks (GNNs) can learn from data that do not have a rigid structure like a grid or a sequence, and can be depicted in the form of unordered entities and relations which constitute graphs. The formulation of GNN models that deal with complex data structures has recently seen fast developments (Zhou et al., 2018; Wu et al., 2019) - such models are therefore strong candidates for the analysis of rs-fMRI data.

In this work, we propose a model that uses GNNs to account for spatial inter-relationships between brain regions, and temporal convolutional networks (TCNs) to capture the intra-temporal dynamics of blood-oxygenated-level-dependent (BOLD) time series. By incorporating GNNs and CNNs in the same end-to-end architecture, we essentially combine intra- and inter-feature learning. In particular, GNNs can lift the limitation of assuming linearity in the interactions between brain region-specific time series by capturing higher-order interactions between regions of interest (ROIs). We further engineered our architecture to specifically retain edge weights (hence circumventing the common and arbitrary practice of thresholding and binarising adjacency matrices) and to contain elements of explainability (Arrieta et al., 2020; Samek et al., 2019). This was done specifically to provide advantages when a neuroscientific explanation of the inner model workings is desirable. To test our architecture, we use the publicly available UK Biobank dataset, which provides rs-fMRI scans from more than 30,000 distinct people. This dataset offers a unique opportunity to formulate novel architectures, while supporting the need of large datasets for reproducible findings with minimal statistical errors (Marek et al., 2020). We also conduct an ablation analysis on a proof-of-concept binary sex prediction task to better evaluate the different contributions of each component of our model. Finally, to assess the effectiveness and flexibility of our architecture, we retrain it using the multimodal Human Connectome Project (HCP) dataset in two distinct experiments, one of which contains multimodal neuroimaging data (i.e., rs-fMRI and structural adjacency matrices derived from diffusion-weighted imaging).

A very preliminary version of this work, based on a 30-fold smaller dataset, was recently presented as a conference contribution (Azevedo et al., 2020). In this current paper, in addition to employing a significantly larger dataset, we expand the choices in the graph threshold hyperparameter, and analyse the effect of including edge weights. Further, the previous contribution only used 1D-CNNs and a graph convolutional networks (GCNs) with binary graphs, as opposed to a general GNN architecture allowing the inclusion of edge weights. In addition, no explainability analysis was performed in the previous conference paper. We release all the code used to develop this work in a public repository for easier adoption by the community (see "Data and Code Availability" section).

2. Related work

Previous work using deep learning for analysing rs-fMRI can be broadly grouped by how the spatial and temporal dimensions are processed. For the vast majority of methods, rs-fMRI is treated as euclidean data arranged on a image grid. A commonly used image representation within this domain is the functional connectivity matrix (FCM): a 2D matrix constructed by using a statistical measure of similarity between ROI-derived time series (Wang et al., 2014). Both multilayer perceptrons (MLPs) and CNNs have been used extensively on FCMs to learn features in order to classify autism spectrum disorder (Heinsfeld et al., 2017; Eslami et al., 2019) and attention deficit hyperactivity disorder (Riaz et al., 2020). A major drawback of using FCMs is that they require an *a priori* choice of similarity measure, possibly introducing unrealistic bias into the data. For example, the often employed Pearson correlation coefficient can only measure linear associations between BOLD signals. More recently, in line with growing interest in dynamic functional connectivity (Allen et al., 2014; Preti et al., 2017), CNNs have been combined with RNNs to learn from time windowed FCMs for tasks such as fluid intelligence prediction (Fan et al., 2020) as well as identifying major depression (Yan et al., 2020). However, in addition to the choice of similarly measure, the construction of classical, dynamic FCMs requires the selection of a window length, which again is arbitrary and not trivial (Hutchison et al., 2013). An alternative to the FCM representation is to use the entire 4D brain volume timeseries as input to convolutional RNNs (Bengs et al., 2020; El-Gazzar et al., 2020; Parmar et al., 2020). Processing voxelwise fMRI data, however, ignores the empirical evidence that functional brain activity may be localised depending on the task and exhibit very strong spatial correlations (Sporns, 2011). This would result in learning computationally expensive features which likely contain largely redundant information.

In line with the view of the human brain as a dynamical functional connectome, more recent deep learning approaches treat rs-fMRI data as a graph. Within this approach, ROIs are commonly employed to represent graph nodes, and edges between nodes are determined by a choice of similarity measure as per FCMs (Sporns, 2011; 2018). In this framework, GNNs can be used to learn features between neighbouring ROIs by propagating information through the edges which connect them. Due to their scalability and interpretability, GNNs for rs-fMRI analysis have been widely used to model tasks such as gender classification (Arslan et al., 2018; Kim and Ye, 2020; Gadgil et al., 2020), age prediction (Gadgil et al., 2020), as well as to find imaging biomarkers for brain disorders such as cognitive impairment (Wen et al., 2018) and autism spectrum disorder (Li et al., 2020). To date, the most common type of graph convolution used for rs-fMRI analysis has been spatial convolutions (Gadgil et al., 2020; Li et al., 2021) although spectral (Parisot et al., 2017; Ktena et al., 2018) and edge convolutions (Wang et al., 2021) have also proven successful for classification tasks. A major limitation of existing works is that graph topology is estimated by taking a group average of FCMs

(Kim and Ye, 2020; Wang et al., 2021). As a result, connectivity between subjects is assumed to be invariant. Furthermore, the initial choice of features used to represent ROIs is not trivial, ranging from graph theoretic measures to connectivity differences. We address these limitations through our novel combined GNN and CNN model architecture which is capable of learning from individual graph topologies as well as learning its own nodes features.

3. Methods

3.1. Problem definition

To represent rs-fMRI data as an undirected weighted graph, the brain is spatially parcellated into N regions of interest (ROIs) representing graph nodes indexed by the set $\mathcal{V} = \{1, \dots, N\}$. Let $\boldsymbol{x}_i \in \mathbb{R}^T$ represent the features of node i corresponding to the BOLD time series of length T. The connections between ROIs are represented by an edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ composed of $|\mathcal{E}|$ unordered pairs (i, j), where for every edge k connecting two nodes $(i, j) \in \mathcal{E}$ the connection strength is defined as $\boldsymbol{e}_k \in \mathbb{R}$. Let the tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the resulting graph. Given the graph structure $\mathcal{G}$, let $\boldsymbol{X} \in \mathbb{R}^{N \times T}$, $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{E}| \times 1}$, and $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ denote the nodes features, edge features and adjacency matrix, respectively.

3.2. Temporal convolutional networks

There has been evidence that a convolutional operator could perform equally well (or even better) as compared to RNNs for sequential data. Some advantages of the convolutional operator are, for instance: (1) lower requirements for long input sequences, especially compared to LSTMs and GRUs, which commonly consume big chunks of memory to store partial results for the multiple gates (convolutional kernels, in contrast, are shared across a layer), (2) better parallelisation because a TCN/CNN layer is processed as a whole instead of sequentially as in RNNs, and (3) easier to train (e.g., it is known that LSTM training can commonly encounter issues with vanishing gradients). Other teams in industry and academia have found similar results when using convolutional operations for sequential data, for instance, in sequence-to-sequence prediction/learning (Elbayad et al., 2018; Gehring et al., 2017), machine translation (Kalchbrenner et al., 2016; Kaiser et al., 2018), and others (Chen et al., 2020). In summary, there is evidence that although LSTMs have historically been used for sequential data, CNNs can achieve similar or better performance at a significantly lower cost.

In order to learn a representation of the temporal dynamics contained in rs-fMRI time series, we use temporal convolutional networks (TCNs) (Bai et al., 2018). These are a simplification over the original *WaveNet* architecture used for audio synthesis (van den Oord et al., 2016), which has been seen to provide significantly better results for sequence modelling in comparison to more traditional RNN architectures (e.g., LSTMs) across a range of tasks and datasets. In particular, Bai et al. (2018) posit that convolutional networks should be seen as the natural starting point for sequence modelling tasks, which makes them ideal for extracting information from rs-fMRI time series.

TCNs are based on dilated causal convolutions (Yu and Koltun, 2016), which are special 1D filters where the size of the receptive field exponentially increases over the temporal dimension of the data as the depth of the network increases. The padding of the convolution is 'causal' in the sense that an output at a specific time step is convolved only with elements from earlier time steps from the previous layers, thus preserving temporal order. More formally, given a single ROI time series $\boldsymbol{x}_i \in \mathbb{R}^T$ and a filter $\boldsymbol{f} \in \mathbb{R}^K$, the dilated causal convolution operation of $\boldsymbol{x}$ with $\boldsymbol{f}$ at time t is represented as

$$\boldsymbol{x}_i * \boldsymbol{f}(t) = \sum_{s=0}^{K-1} \boldsymbol{f}(s) \boldsymbol{x}_i(t - d \times s), \tag{1}$$

where $d = 2^{l-1}$ is the dilation factor which, depending on the layer l controls the number of time steps successively skipped. This relation between the dilation factor and the layer l is the one defined in the original paper (Bai et al., 2018), which we follow in this work. In contrast to the original TCN architecture, we use batch normalisation instead of weight normalisation because it empirically provided a more stable training procedure in terms of loss evolution.

3.3. Graph network block

Battaglia et al. (2018) formalise a graph network (GN) framework through the definition of functions that work on graph-structured representations. The main unit of computation in the GN framework is called the *GN block* and contains two update functions and one aggregation function working on the edge and node levels.

The first operation of this GN block, which can be broadly defined as the *edge model*, concerns the update function ϕ^e, which computes updated edge attributes for each edge k based on the original edge's attributes $\boldsymbol{e}_k$ and the features of the connected nodes i and j:

$$\boldsymbol{e'}_k = \phi^e\left(\boldsymbol{e}_k, \boldsymbol{x}_i, \boldsymbol{x}_j\right). \tag{2}$$

Note that for rs-fMRI graph representations, each edge originally contains a single value (i.e., $\boldsymbol{e}_k \in \mathbb{R}$), but after this operation ϕ^e, the resulting dimensionality can be different: $\boldsymbol{e'}_k \in \mathbb{R}^M$, where $M >= 1$. Then, in what can be broadly defined as the *node model*, the block computes updated node features. Firstly, for each node i, it aggregates the edge features per node:

$$\bar{\boldsymbol{e}}'_i = \rho^{e \to v}\left(\mathcal{E}'_i\right), \tag{3}$$

where $\mathcal{E}'_i = \{(\boldsymbol{e'}_k, i, j)\}_{k=1}^{E}$ is the set of edges starting in node i, with node j connected with node i through edge k. Importantly, $\rho^{e \to v}$ needs to be invariant to edge permutations to account for the unordered structure of the data. Averaging and summation are examples of such operations invariant to edge permutations.

Finally, the updated node features are computed using another update function at the node level, for each node i:

$$\boldsymbol{x'}_i = \phi^v\left(\bar{\boldsymbol{e}}'_i, \boldsymbol{x}_i\right). \tag{4}$$

The aggregation function $\rho^{e \to v}$ needs to be invariant to edge permutations, but the update functions (i.e., ϕ^e and ϕ^v) are more flexible. For example, if the features are vectors in 1D space, the update functions could be implemented as multi-layer perceptrons (MLPs); however, a CNN or RNN could be more suitable if the features represent images or sequences, respectively. Section 4.2 details how these functions were implemented in this paper.

Although the rs-fMRI graph representation contains undirected edges, the GN block requires directed edges. To overcome this issue, every time there is a connection between any two nodes i and j, we assume the existence of two edges $(\boldsymbol{e}_k, i, j)$ and $(\boldsymbol{e}_k, j, i)$, one for each direction. The original GN block (Battaglia et al., 2018) further contains one update function and two aggregation functions for global (i.e., graph-level) features; however, we do not use this formalisation as it is not applicable in the fMRI data representation of this paper.

3.4. Graph pooling

After the neural network processes the input as described in the previous sections, each node in the graph will contain a node-wise

representation (i.e., a feature vector) as a result. For the prediction task described in this paper, where a graph-level (as opposed to node-level) prediction is required, these representations need to be pooled (i.e., collated) to be used for a final downstream prediction task.

To this end, it is common practice to employ a global pooling mechanism, in which node features are pooled across the graph (e.g., by averaging or concatenating all node features), thus creating a final, low-dimensional embedding representation of the graph itself. Given the graphs used in this paper all have the same number of nodes, a concatenation pooling mechanism is indeed possible.

However, assuming that distinct nodes (i.e., brain regions in this paper) have different levels of importance for the downstream prediction task (Kiebel et al., 2008; Hilgetag and Goulas, 2020), we assumed that a hierarchical (as opposed to flat) pooling mechanism would create richer embeddings. To this end, we employ the differentiable pooling operator introduced by Ying et al. (2018), commonly called DiffPool, which learns how to sequentially collapse nodes into smaller clusters until only a single node with the final embedding exists.

When describing a Graph Network (GN) block, a sparse representation of nodes and edges is used to describe the operations that a GN block can have; however, DiffPool works on dense representations of a graph. In other words, a graph $\mathcal{G}$ is represented by a dense adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and a feature matrix $\boldsymbol{X} \in \mathbb{R}^{N\times F}$, where N is the number of nodes and F the number of features in each node.

The DiffPool operator, at layer l, thus receives both an adjacency matrix and a node embedding matrix, and computes updated versions of both:

$$\boldsymbol{A}^{(l+1)}, \boldsymbol{X}^{(l+1)} = \text{DiffPool}\left(\boldsymbol{A}^{(l)}, \boldsymbol{X}^{(l)}\right). \tag{5}$$

To achieve this, the DiffPool operator uses a graph neural network (GNN) architecture. Specifically, the same GNN architecture is duplicated to compute two distinct representations: a new embedding $\boldsymbol{Z} \in \mathbb{R}^{N_{(l)}\times F'}$ and an assignment matrix $\boldsymbol{S} \in \mathbb{R}^{N_{(l)}\times N_{(l+1)}}$:

$$\boldsymbol{Z}^{(l)} = \text{GNN}_{l,\text{embed}}\left(\boldsymbol{A}^{(l)}, \boldsymbol{X}^{(l)}\right) \tag{6}$$

$$\boldsymbol{S}^{(l)} = \text{softmax}\left(\text{GNN}_{l,\text{pool}}\left(\boldsymbol{A}^{(l)}, \boldsymbol{X}^{(l)}\right)\right), \tag{7}$$

where $N_{(l)}$ is the number of nodes in layer l, $N_{(l+1)}$ the new number of nodes, each corresponding to a cluster ($N_{(l+1)} < N_{(l)}$), and F' the number of features per node, which can be different from the original size F from the matrix $\boldsymbol{X}$.

The operator ends with the creation of the new node embedding matrix and adjacency matrix, to be inputted to the next layer:

$$\boldsymbol{X}^{(l+1)} = {\boldsymbol{S}^{(l)}}^T \boldsymbol{Z}^{(l)} \tag{8}$$

$$\boldsymbol{A}^{(l+1)} = {\boldsymbol{S}^{(l)}}^T \boldsymbol{A}^{(l)} \boldsymbol{S}^{(l)}, \tag{9}$$

where $\boldsymbol{X}^{(l+1)} \in \mathbb{R}^{N_{(l+1)}\times F'}$ and $\boldsymbol{A}^{(l+1)} \in \mathbb{R}^{N_{(l+1)}\times N_{(l+1)}}$. Overall, Eqs. (6)–(9) are the ones responsible to implement Eq. (5).

4. Experiments overview

4.1. Main dataset - UK biobank

Subject-level structural T1 and T2-FLAIR data as well as ICA-FIX (Salimi-Khorshidi et al., 2014) denoised rs-fMRI data were obtained from UK BioBank (application 20904) (Bycroft et al., 2018)[1]. All data were acquired on a standard Siemens Skyra 3T scanner running VD13A SP4, with a standard Siemens 32-channel RF receive head coil. The structural data were further preprocessed with Freesurfer (v6.0)[2] using the T2-FLAIR weighted image to improve pial surface reconstruction, similarly to Glasser et al. (2013)'s pipeline. Reconstruction included bias field correction, registration to stereotaxic space, intensity normalisation, skull stripping, and white matter segmentation. When no T2-FLAIR data were available, Freesurfer reconstruction was done using the T1 weighted image only.

Following surface reconstruction, the Desikan-Killiany atlas (Desikan et al., 2006) was aligned to each individual structural image, and ROIs were mapped into each individual's space for subsequent time series extraction. To this end, the same atlas was aligned to the functional denoised rs-fMRI data (490 volumes TR/TE = 735/39.00 ms, multiband factor 8, voxel size: 2.4 × 2.4 × 2.4, FA=52 deg, FOV 210x210 mm) using the warping parameters computed during the structural-to-functional alignment obtained using FSL's linear registration (FLIRT), and mean BOLD time series (490 timepoints per scan) were extracted for each ROI. The time series were then scaled subject-wise using the median and interquartile range according to the *RobustScaler* implementation in the *scikit-learn* (Pedregosa et al., 2011) python package. Edge weights were defined as full correlations calculated with the Ledoit Wolf covariate estimator using the *nilearn* python package[3]. Figure 1 shows an example scaled time series and the resulting example graph from a single subject. The total number of subjects used from the UK Biobank was 35,159, in which 18,649 were females and 16,510 were males ($18,649/16,510 \approx 1.13$). The median age was 64, with a minimum age of 44 and a maximum of 81.

4.2. Model implementation

The neural network architecture depicted in Fig. 2 was implemented using Pytorch (Paszke et al., 2019), and Pytorch Geometric (Fey and Lenssen, 2019) for the specific graph neural network components. The edge feature matrix $\boldsymbol{E} \in \mathbb{R}^{E\times 1}$ defined in Section 3.1 was implemented as two sparse matrices: a sparse representation of the adjacency matrix $\mathbf{E}_i \in \mathbb{R}^{2\times E}$, and a sparse representation of the edge features $\mathbf{E}_a \in \mathbb{R}^{E\times 1}$ (i.e., there was only one feature per edge corresponding to the correlation value). The number of nodes N was 68 (corresponding to each brain region from the Desikan-Killiany atlas), the number of node features F was the number of timepoints (i.e., 490), and E is the number of edges in the graph. The number of edges depends on the threshold percentage used to retain only the strongest correlations. Given the non-conclusive evidence on the optimal threshold percentage in the vast majority of functional connectivity literature (Garrison et al., 2015), in this work this threshold was included in the hyperparameters to be optimised.

The full list of hyperparameters and respective value ranges were as follows:

- `dropout`: [0, 0.5] (uniform distribution)
- `threshold`: {5, 10, 20, 30, 40} (categorical)
- `learning rate`: [1e−5, 1e−1] (log uniform distribution)
- `weight decay`: [1e−12, 1e−1] (log uniform distribution)

The model starts by employing a temporal convolutional network (TCN) architecture (Bai et al., 2018) to extract a lower-dimensional embedding representation from the rs-fMRI time series in each node. This was implemented by using three blocks,

[1] https://biobank.ctsu.ox.ac.uk/crystal/crystal/docs/brain_mri.pdf

[2] http://surfer.nmr.mgh.harvard.edu/

[3] https://nilearn.github.io/

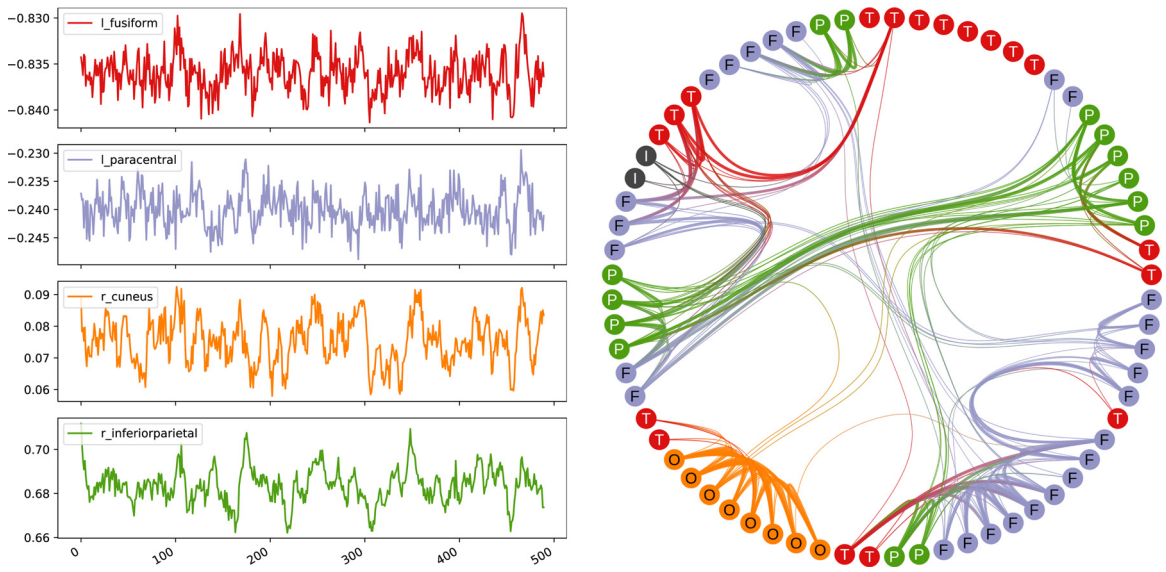

Fig. 1. Left: Mean BOLD time series extracted from four brain regions (see legend) from one subject's data, after scaling. Right: Graph representation of the same subject's data, at 10% threshold as described in Section 4.2. Thicker edges represent a stronger correlation between nodes, in this case with values between approximately 0.54 and 0.87. Each node is labelled and coloured according to the brain region it represents (i.e., T/F/O/P/I correspond to Temporal, Frontal, Occipital, Parietal, and Insula).

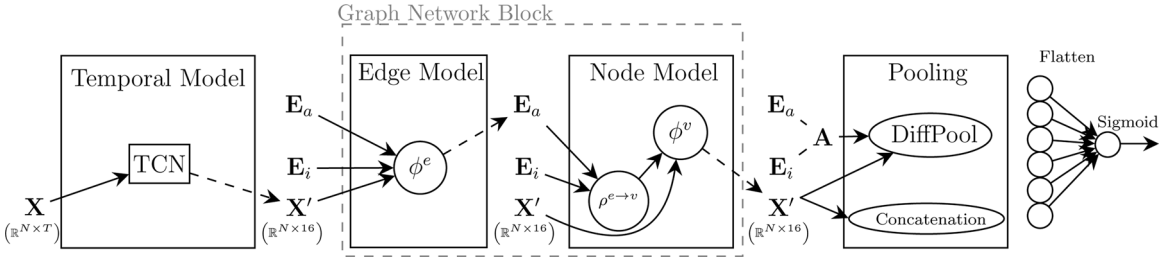

Fig. 2. Main working blocks of the spatio-temporal model. The temporal model creates an initial representation from the original node features $\boldsymbol{X}$ (i.e., temporal dynamics). This is followed by transformations operated by the Graph Network Block which leverages the structure of data represented in edge features $\boldsymbol{E}_a$ and its sparse connectivity $\boldsymbol{E}_i$. Finally, a Pooling mechanism (either DiffPool or concatenation) creates a graph representation which is flattened and employed for a final prediction task.

each of which containing two layers of *1D convolutions, 1D batch normalisation, ReLU activation*, and *dropout*. Each block uses a kernel with size 7 (i.e., $K = 7$ in Eq. (1)), containing a skip connection, and increases the number of output channels at each block, specifically 8, 16, and 32. Dilation factor d was set to $d = 2^{l-1}$, where l is the block (i.e. $l \in \{1, 2, 3\}$). After these three blocks (i.e., six layers), node features from all channels are flattened to form the input to a linear transformation which reduces each node representation to a fixed embedding of size 16. These transformations thus reduce the original node feature matrix from size $N \times T$ to size $N \times 32 \times T$ after the three blocks, and finally to size $N \times 16$, corresponding to the final embedding.

The Graph Network (GN) block is then applied, in which the update functions ϕ^e and ϕ^v in Eqs. (2) and (4) are multi-layer perceptrons (MLPs), and the function $\rho^{e\to v}$ in Eq. (3) is a set of aggregators following Corso et al. (2020)'s work (i.e., edge-wise mean, min, max, standard deviation, and sum). We stack 3 GN blocks, after each of which we apply an *1D batch normalisation* over the node's features and a *ReLU activation*. The original dimensions of $\boldsymbol{X}$, $\boldsymbol{E}_i$, and $\boldsymbol{E}_a$ before the GN block are kept after these transformations.

We employed two types of pooling mechanisms, both of which reduce the node feature matrix from a size of $N \times 16$ to a size of 1×16: a concatenation over all node's features followed by a single-layered MLP, and the hierarchical pooling mechanism (i.e., DiffPool). For DiffPool, which expects a dense graph representation, data are first transformed into a symmetric adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, which is a weighted matrix when considering edge features, and binary otherwise. Similarly to the original DiffPool paper (Ying et al., 2018), we employed three layers of the graph neural network operator from Morris et al. (2019) (to make use of weighted adjacency matrices) followed by a *1D batch normalisation*, with a final skip connection.

We empirically evaluated various architectural choices on a single training fold of the dataset, and how they influenced performance on the corresponding validation fold. This ad-hoc evaluation showed that a higher number of TCN layers and overall model complexity slightly improves performance. However, given that some of these parameter choices (e.g. with 4 TCN blocks) produced a significantly higher number of learnable parameters and prohibitive GPU memory constrains, effectively inhibiting experimentation, the final TCN architecture was chosen as described in this section (with 3 TCN blocks and a final embedding size of 16).

A conceptual summary of the whole model is shown in Fig. 2.

4.3. Training procedure

In order to assess the validity of our model, we performed proof-of-concept experiments through the well-known binary sex prediction task (Jiang et al., 2019; Weis et al., 2019). We used a 5-fold stratified cross-validation procedure: the UK Biobank dataset was divided into training and test sets five times, in which each test set corresponds to 20% of the original size, and a sample would only belong to a test set once (i.e., all test sets are mutually

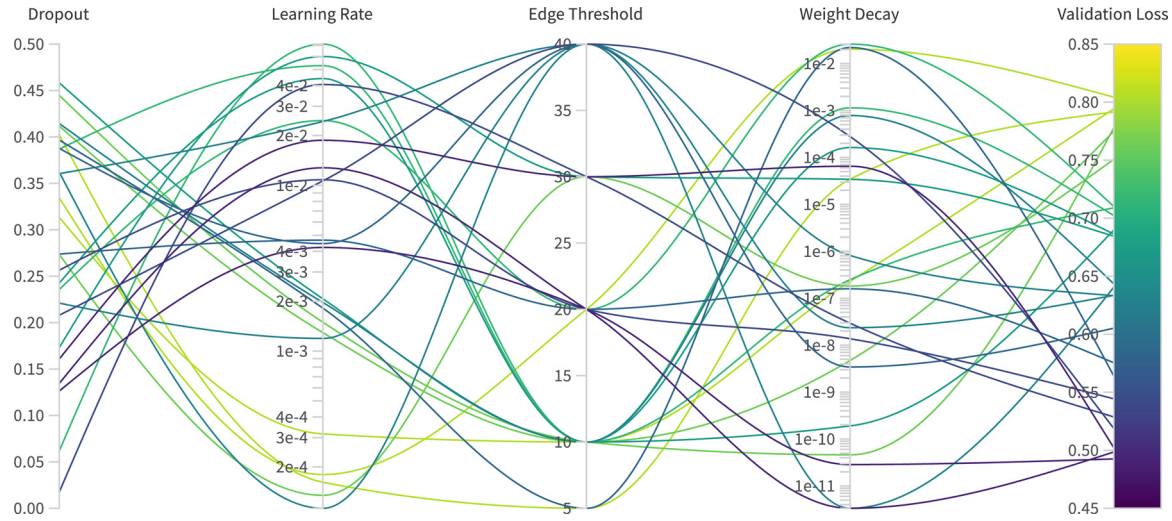

Fig. 3. Values of hyperparameters corresponding to each validation loss achieved for one illustrative inner sweep of one fold. For each one of the 25 training runs (each represented by a curved line), a set of random values is chosen for dropout, learning rate, edge threshold and weight decay, which ultimately results in the model's validation loss.

exclusive). This division was done in a stratified fashion considering the sex label, bucketised age, and bucketised BMI measures (for each variable we created 8 equal-sized buckets based on sample quantiles). For each test set, the training set is further divided once to generate single inner training and validation sets, using the same stratification strategy as for the training/test case.

The neural network was trained over 150 epochs with the RMSprop optimiser (Tieleman et al., 2012) and Binary Cross-Entropy loss function. The training procedure was set to stop early if the validation loss did not reduce further after 33 consecutive epochs. Learning rate is reduced by a factor of 0.1 with patience of 30. A hyperparameter search was included in the inner training/validation sets, in which 25 random runs were launched exploring random values of dropout, edge threshold, learning rate, and weight decay (see Section 4.2 for ranges explored). In each random run, the model with the smallest validation loss was saved, and the model with the smallest validation loss across the 25 runs was selected to be evaluated in the test set. This procedure is done separately for each test set, and metrics are then averaged across the five test sets.

We used *Weights & Biases* Biewald (2020) to log our training procedure and generate the random hyperparameters for all the 25 models in each inner sweep. These inner sweeps were run across two different servers, and each model took between 20 min and 11 h to train depending on GPU type and early stopping. All these details are stored using *Weights & Biases*, and can be accessed through our public repository (see "Data and Code Availability"). Figure 3 shows the results for the inner sweep of one of the folds for illustrative purposes. While a certain amount of variability is visible, some trends are evident in this particular split: the best models (i.e., with lower validation loss) tend to be achieved with higher edge thresholds, higher learning rates, and lower dropout rates. We highlight that different sweeps could result in different trends.

4.4. Evaluation

As shown in Fig. 2, our model consists of (1) a TCN block that learns intra-temporal features from the mean BOLD time series of each ROI, followed by (2) a GN block which leverages the spatial inter-relationships between ROIs, and finally (3) a hierarchical pooling mechanism which leverages all the information in the input, from the temporal rs-fMRI dynamics to the graph structure and the edge features of that graph.

To understand the inner workings of this combination, we conducted an ablation analysis to quantify the contributions of each component of our model for the specific prediction task. Firstly, we consider two cases where the GN block is not used, hence essentially evaluating the importance of edge weights for this prediction task. In one case the graph structure is completely ignored (i.e., no GN block and concatenation pooling), and in another case a binary graph is used only for the final hierarchical pooling part (i.e., no GN block and DiffPool applied to a binary graph).

In order to investigate the influence of the different GN components, we consider not only the case where both *node model* and *edge model* are used in the GN Block, but also a case where only the *node model* is applied. For each of these two cases, both a concatenation pooling and DiffPool with weighted adjacency matrices are considered.

We compare our approach to two deep learning models. The first one, by Gadgil et al. (2020) (which we named CNSLAB) is based on a voting scheme across timesteps, and the second, by Wang et al. (2021), named cGCN, uses averaged FCMs. For both we used the best hyperparameters selected from each paper/repository and trained those models on our preprocessed data.

We also compare our approach to baseline models where data structure is not leveraged; here, the entire data representation is flattened and fed into two non-deep learning models. To this end, we employed: (1) a support vector machine (SVM) classifier with a linear kernel and hyperparameter search over the regularisation parameter, and (2) a XGBoost (Chen and Guestrin, 2016) classifier with hyperparameter search over several parameters.

4.5. External multimodal dataset - human connectome project

To further evaluate the effectiveness and flexibility of our end-to-end architecture, we analysed its behaviour in a multimodal setting, i.e. when adjacency matrices and timeseries are derived from distinct imaging procedures (fMRI and diffusion-weighted MRI, respectively). We employed the preprocessed Human Connectome Project (HCP) fMRI Data. This dataset consists of four 15-minute-long fMRI sessions (TR = 0.72s) per subject, acquired on a 3T scanner with isotropic spatial resolution of 2mm in 1003 healthy subjects, and preprocessed according to Glasser et al. (2013). For each subject, this results in 4 distinct sessions/samples per sub-

Table 1
Ablation analysis, with metrics averaged across the five test sets, with standard deviation in parenthesis. Aggregator on the right-hand side of the arrow, "N" corresponds to only *node model*, and "N + E" corresponds to full Graph Network block. **Params** stands for number of parameters.

Model	AUC	Accuracy	Sensitivity	Specificity	Params
N + E → Concat	**0.92** (0.004)	**0.85** (0.006)	**0.85** (0.006)	0.84 (0.012)	291,898
N + E → DiffPool	0.82 (0.020)	0.75 (0.016)	0.72 (0.030)	0.77 (0.025)	287,420
N → Concat	**0.92** (0.003)	0.84 (0.004)	0.84 (0.028)	**0.85** (0.029)	291,337
N → DiffPool	0.84 (0.020)	0.76 (0.020)	0.75 (0.013)	0.77 (0.038)	286,859
→ DiffPool	0.84 (0.010)	0.76 (0.008)	0.75 (0.019)	0.77 (0.023)	278,843
→ Concat	**0.92** (0.012)	0.84 (0.013)	0.84 (0.024)	0.84 (0.023)	283,321
CNSLAB (Gadgil et al., 2020)	0.86 (0.003)	0.78 (0.005)	0.76 (0.024)	0.79 (0.018)	198,937
cGCN (Wang et al., 2021)	0.77 (0.021)	0.70 (0.018)	0.66 (0.028)	0.74 (0.040)	45,065
XGBoost	0.89 (0.003)	0.81 (0.005)	0.80 (0.008)	0.82 (0.006)	-
SVM	0.79 (0.015)	0.79 (0.017)	0.82 (0.098)	0.76 (0.101)	-

ject with 1200 timesteps for each sample and component. In order to ensure comparability to the UK Biobank experiments, every timeseries was truncated to 490 timepoints. Similarly to the steps described in Section 4.1, the Desikan-Killiany atlas (Desikan et al., 2006) was aligned to each individual structural image, warped into single subject space, and employed to extract ROI- and subject-wise timeseries which were scaled subject-wise. Diffusion data was processed locally using multi-tissue, multishell constrained spherical deconvolution (Jeurissen et al., 2014) to obtain orientation distribution function estimates, which were then passed to probabilistic fiber tracking (10^8 tracks, subsampled to 10^7 tracks through Spherical-deconvolution Informed Filtering of Tractograms Smith et al., 2013). Structural connectivity matrices were obtained by length-normalised streamline counts between the same ROIs described above. A total of 3668 graphs were used (1692 males and 1976 females), where nodes correspond to Desikan-Killiany ROIs, node features correspond to 490 time points, and the adjacency matrix corresponds to the structural connectivity extracted from probabilistic tractography.

All training and evaluation steps were kept identical across all datasets.

5. Results

5.1. General results

Table 1 shows the results of our ablation analysis across three different backbones - no graph block, only *node model*, and full graph network block - each with two different aggregators (i.e., concatenation and DiffPool). We identify each one of these cases using a "Backbone → Aggregator" notation, in which *Aggregator* can be "Concatenation" or "DiffPool", and *Backbone* can be "N" for only *node model*, "N + E" for both *node model* and *edge model* (i.e., full GN Block), and empty otherwise. We also include results from the baselines experiments.

Our model performs significantly better as compared to all baselines but in which the model without a GNN block (i.e., "→ Concat') is similarly good. The SVM baseline performs worse overall and involves an increase in the standard deviation of performance parameters, possibly indicating that our model is more robust to different dataset divisions (i.e., folds), while retaining the flexibility and representation ability described above. Using DiffPool as a final aggregator appears to result in worse overall performance when compared to the concatenation counterpart and, in some metrics, to some baselines. Using the *edge model* did not bring significantly better results when compared to using the *node model* only, possibly indicating that the information contained in the edge attributes is successfully leveraged by the *node model* alone for this particular prediction task.

The results presented so far consider an adjacency matrix threshold below 50% as a hyperparameter at training time, a common data reduction practice in the connectivity analysis field. We further analysed the results of using no threshold at all, and explored the type of activation function as a hyperparameter instead (i.e., *ReLU* or *tanh* activations). This choice was made explicitly since retaining 100% of the adjacency matrix elements results in a share of negative correlation elements, whose physiological significance is likely to be important in brain connectivity (Zhan et al., 2017). The results of this analysis are presented in Table 2.

The performance was slightly lower for most cases which did not include a threshold, especially for the N + E → DiffPool model. A possible explanation would be the excessive "noise" (i.e., low, possibly spurious correlations) not allowing the graph's dominating spatial structure to be successfully leveraged in a practical timeframe, in turn possibly resulting in some degree of overfitting. However, performance metrics remain comparable or better to what is illustrated in Table 1, suggesting that these models are still able to extract spatial information from the data after training despite of the significant increase in memory usage.

5.2. Evaluating architectural choices

To better understand the utility of TCNs when compared to the more traditional LSTMs, we reran six ablations using the UK Biobank dataset, in which we substituted the TCN block with a LSTM block. Striving for a fair comparison between LSTM and TCN, we used the same number of layers in both (i.e., three layers), and chose the feature dimension in the hidden state such that the total number of learnable parameters would be similar. We made sure that the 25 runs per fold would have the same hyperparameter ranges in both the TCN and LSTM cases. Table 3 shows that the LSTM models achieve similar performance to the TCN models; however, this comes at a significantly higher computational cost. Due to computational constraints, we are not able to fairly compare the runtimes among all models because of the use of different servers with different GPU cards (see Section "Acknowledgements"). However, for representation purposes, there are two folds in the "N → DiffPool" model (i.e., folds 4 and 5) which were run in the same GPU for both the TCN and LSTM cases; in this case, the average runtime per model training went from 1 h and 35 min (fold 4) and 1 h and 38 min (fold 5) in the TCN case, to an average of 3 h and 14 min (fold 4) and 2 h and 47 min (fold 5) in the case of the LSTM. Given that these experiments were run on the most recent NVIDIA A100 GPUs, which are able to speedup runtimes by a large factor when compared to older GPUs, we expect these differences to be more striking when running the models on more commonly used hardware.

Table 2
Results with no thresholded graphs, with metrics averaged across the five test sets, with standard deviation in parenthesis. Aggregator on the right-hand side of the arrow, "N" corresponds to only *node model*, and "N + E" corresponds to full Graph Network block.

Model	AUC	Accuracy	Sensitivity	Specificity
N + E → Concat	0.92 (0.002)	0.84 (0.004)	0.85 (0.014)	0.83 (0.017)
N + E → DiffPool	0.77 (0.012)	0.70 (0.011)	0.68 (0.080)	0.72 (0.067)
N → Concat	0.93 (0.003)	0.85 (0.003)	0.83 (0.017)	0.86 (0.019)
N → DiffPool	0.85 (0.007)	0.77 (0.008)	0.77 (0.026)	0.77 (0.017)

Table 3
Results when using an LSTM instead of a TCN in the temporal block (UK Biobank fMRI dataset).

Model	AUC	Accuracy	Sensitivity	Specificity
N + E → Concat	0.93 (0.002)	0.85 (0.004)	0.85 (0.006)	0.85 (0.005)
N + E → DiffPool	0.84 (0.014)	0.76 (0.015)	0.74 (0.051)	0.77 (0.031)
N → Concat	0.93 (0.004)	0.85 (0.007)	0.85 (0.020)	0.86 (0.019)
N → DiffPool	0.84 (0.020)	0.76 (0.017)	0.78 (0.025)	0.74 (0.035)
→ DiffPool	0.82 (0.035)	0.73 (0.034)	0.73 (0.083)	0.74 (0.090)
→ Concat	0.91 (0.003)	0.83 (0.003)	0.82 (0.020)	0.83 (0.012)

Table 4
Results when no TCN block is used to train and evaluate on the UK Biobank dataset.

Model	AUC	Accuracy	Sensitivity	Specificity	Params (Before)
N → Concat	0.93 (0.004)	0.85 (0.004)	0.85 (0.014)	0.86 (0.013)	23,541,071 (291,337)
N + E → Concat	0.93 (0.002)	0.85 (0.003)	0.84 (0.017)	0.86 (0.015)	24,022,742 (291,898)

In summary, our experiments confirm findings that RNNs and CNNs can provide similar performance, but the former come with a significantly higher computational cost.

We further evaluated the impact of including the TCN block in the model. In this "no TCN" experiment, we omitted the TCN block and therefore only the GNN components are present, with a much larger temporal feature representation (i.e., 490 raw timepoints instead of the 16 features created by the TCN block). Table 4 shows that performance metrics were similar between the TCN and no TCN versions, but the latter resulted in an almost 100-fold increase in the number of parameters. This means that removing the TCN block came at a very significant cost of an unnecessary explosion in the number of learnable parameters, making the model unnecessarily complex both at training and test time. The important task of finding a good representation in machine learning goes therefore beyond the simple performance analysis (i.e., metrics), and by using a TCN block we are able to find a lower embedding in a realistic time/complexity frame.

5.3. Visualisation of TCN kernels

The weights of the TCN layers can be visually inspected. We visualised the first two layers of one of the trained N + E → Concat models. Fig. 4 shows the weights learned from the first TCN layer (each row corresponding to one of the 8 output channels of that layer), while Fig. 5 depicts the same for the second TCN layer (each row corresponding to one of the 8 output channels and the columns corresponding to the 8 kernels of size 7 coming from the previous 8 channels).

In both figures, and with little exceptions, it can be seen that the output channels in the first two TCN convolutional layers will be a non-trivial weighted multiplication of input channels, as illustrated by the non-trivial patterns in the kernel weights. Given the qualitative variability observed in these weights (which are learned at training time), we argue that these kernels are likely filtering and selecting different, non-mutually redundant patterns present in the original time series. One possible counterexample is the kernel for the 7th output channel in the first TCN convolutional layer illustrated in Fig. 4, which is essentially applying a simple low pass filter by smoothing the original time series from the input channel. It is likely that quantitative analysis and comparison of the kernel weights has the potential to yield interpretable information on which type of brain dynamics may contribute most to the final prediction. Given that these weights are also influenced by additional factors such as normalisation strategy and subsequent nonlinear operations, further research is needed in order to establish a framework to fully exploit this information.

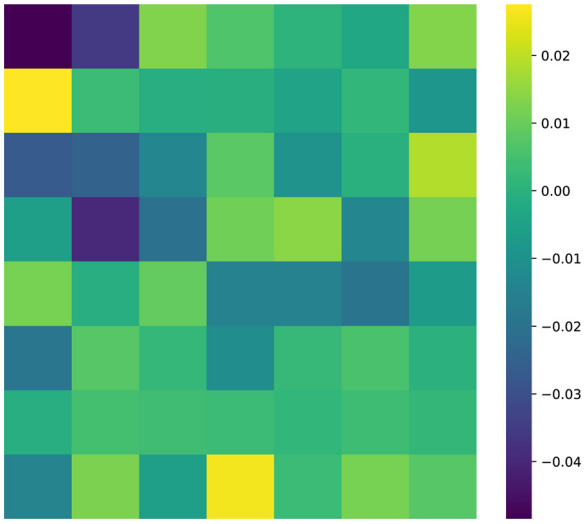

Fig. 4. Weights of the kernels in the first TCN convolutional layer in a N + E → Concat model. Rows correspond to the 8 output channels of this layer, and each column is a position in the kernel array of size 7.

5.4. Explainability of DiffPool clusters

Although deep neural networks are usually regarded as "black boxes", in this paper we strived to inject explainability elements by inspecting which mechanisms were learned during training. To this end, we designed a strategy to inspect the hierarchical spatial pooling mechanism provided by the DiffPool architecture. We analysed the assignment matrices from the first DiffPool layer $\boldsymbol{S}^{(1)}$

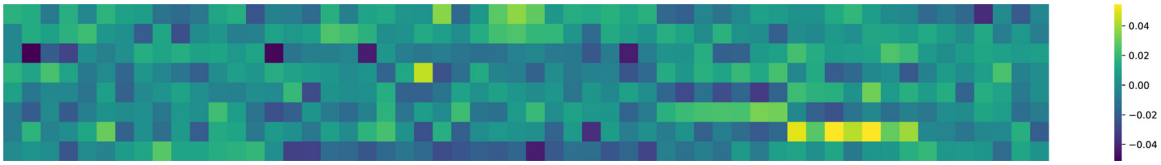

Fig. 5. Weights of the kernels in the second TCN convolutional layer in a N + E → Concat model. Rows correspond to the 8 output channels of this layer, and each column is a position in the 8 kernels of size 7 that come from the 8 input channels (56 columns in total).

(see Eq. (7)), over all participants across all test sets. This is of particular interest because it corresponds to an aggregation of subsets of brain regions which our architecture has considered optimal while learning a particular prediction task. These aggregations can therefore be considered "optimal" for that task within this architecture, and provide insight into the neurophysiology which may drive the formation of such patterns. An assignment matrix corresponds to how the original nodes in the graph will be mapped into new nodes. In this respect, a simple and useful way of summarising this behaviour across individuals is to count how many times two ROIs have ended up in the same cluster, regardless of cluster size and number. More formally, we create an association matrix $\boldsymbol{S}' \in \mathbb{R}^{68\times 68}$, where each element $S'_{i,j}$ is the number of times brain regions i and j have been assigned together in the first DiffPool layer. This means that the higher the value of $S'_{i,j}$, the more often information from brain regions i and j is pooled when learning to predict binary sex. It is important to note that matrix thresholding (see `threshold` hyperparameter in Section 4.2) can - and often will - introduce disconnected nodes in the graph. Since the number of disconnected nodes would vary across individuals, this would introduce unrealistic imbalances/biases in the association matrix $\boldsymbol{S}'$; therefore, in this section, we only employed unthresholded matrices. In specific, we used the best performing DiffPool model (i.e., N → DiffPool) described in Section 4.2.

Figure 6 depicts the association matrix $\boldsymbol{S}'$ for the best performing DiffPool model (i.e., N → DiffPool) trained on unthresholded matrices, with dendrograms resulting from hierarchical clustering of this latter matrix (performed for visualisation purposes). The hierarchical clustering algorithm and the corresponding dendrograms are calculated using the *seaborn* (Waskom, 2021) python package. In addition, we generated a more traditional brain connectivity visualisation by selecting the four main clusters defined by the dendrograms for the N → DiffPool model and overlaying their anatomical correspondence on a sample brain surface in Fig. 7.

An advantage of this explainability strategy (i.e., the use of the association matrix $\boldsymbol{S}'$) is the flexibility inherent in the multiple granularities provided by hierarchical clustering. When choosing large clusters (e.g., four like in Fig. 7) one can illustrate the general aggregation patterns across the brain's anatomy, while by selecting smaller clusters (e.g. twelve clusters) one can reveal more local patterns in the data. In Fig. 8 we depict the brain clusters for the N → DiffPool model with the remaining different granularities (i.e., 8 and 12). This multiscale explainability framework can provide a significant advantage in terms of explainability and interpretation, and is only possible when using the DiffPool strategy.

When looking at how the GNNs clustered the brain regions to optimise and achieve best sex prediction, it is possible to find that clustering into four sets of brain regions showed interesting properties in terms of neurobiological explainability. More specifically, the brain regions were grouped in a manner that mirrors to a certain degree the well-known cytoarchitectural and functional properties of the cerebral cortex. For example, in Fig. 7 cluster 1 (dark blue) included the bilateral frontal cortex as well as occipito-parietal regions that have a well-known role in working-memory, executive functions, and visuo-spatial processing, amongst many other cognitive functions. The left temporal cortex grouped with the paracentral lobule, while the right temporal cortex clustered with the pre-cuneus (light green and light blue, respectively). Cluster 3 (dark green) included several midline cortical areas that form the classic limbic-emotional system.

We do not wish to overinterpret our results or make "reverse neuroscience" inferences in the sense of interpreting *post hoc* the behavioural meaning of a set of regions without having directly analysed their behavioural relevance. However, we speculatively note that the clusters emerged may have some neurobiological relevance in terms of explaining some of the behavioural differences described between males and females in terms of cognitive, motor and emotional skills (Nieuwenhuys et al., 2008; Poeppel et al., 2020). Future work, particularly directed at investigating the links between brain and behavioural measures, is warranted to confirm whether the clustering of regions that our model has generated to achieve optimal sex classification is relevant at phenotypical level. In summary, these results demonstrate the explainability capacity of our model when using the DiffPool aggregator.

To evaluate the robustness of the DiffPool clusters, we compared the association matrices $\boldsymbol{S}'$ for all the five folds for the best performing DiffPool model (i.e., "N → DiffPool") trained on the unthresholded matrices (see Fig. 9). Despite some differences, it is possible to see a similar overall qualitative structure across the folds. To quantify this difference in a simple way, we calculated the normalised difference between every pair of association matrices i and j as:

$$\text{Normalised difference} = \frac{\boldsymbol{S}'_{\text{i}} - \boldsymbol{S}'_{\text{j}}}{\boldsymbol{S}'_{\text{i}} + \boldsymbol{S}'_{\text{j}}}. \tag{10}$$

The various normalised differences can be seen in Fig. 10, with every pair showing an average normalised different below 30%, therefore demonstrating an acceptable stability and robustness of the clusters learned by DiffPool across folds.

5.5. Evaluation on an external multimodal dataset

Table 5 shows the results when training and evaluating our architecture on the Human Connectome Project (HCP) dataset, both for the multimodal (rs-fMRI and diffusion data) and unimodal (only rs-fMRI data) cases. Performance metrics of our model are lower, as compared to the UK Biobank analyses, when considering only rs-fMRI data (i.e., 3–5% difference for concatenation and around 20% difference when using DiffPool). This may illustrate known concerns about the behaviours of deep learning models in general, and graph learning models in particular, when data is scarce; indeed, in the case of rs-fMRI data only, the non-DL baselines reach similar, or slightly better performances when compared to all DL models (both our model and the DL baselines), confirming that DL models can struggle with smaller datasets. However, in the multimodal case, when complementary information from both rs-fMRI (i.e. functional data) and diffusion-weighted MRI (i.e. structural data) are used, our model performs notably better than all baselines. This highlights how our model can flexibly leverage multiple data sources, achieving performances that in some cases are higher than the unimodal results obtained with the much larger UK Biobank dataset. This also emphasises the anticipated outcome

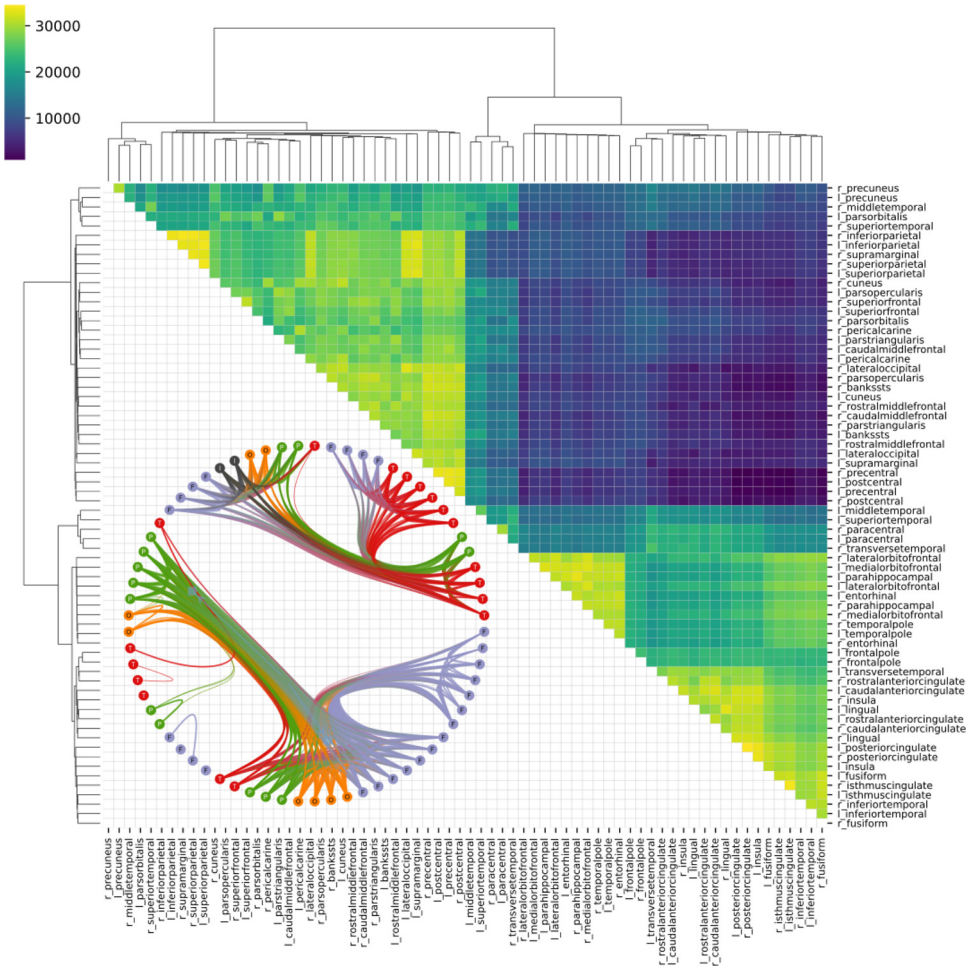

Fig. 6. Upper-triangle of the association matrix $\mathbf{S}'$ for N → DiffPool model generated when predicting binary sex on unthresholded matrices, with dendrograms from hierarchical clustering. Each element $S'_{i,j}$ indicates how many times brain regions i and j are pooled together. On the lower left corner, a graph representation of the same association matrix $\mathbf{S}'$, thresholded at 25% with nodes identified and coloured according to their general brain region (i.e., T/F/O/P/I correspond to Temporal, Frontal, Occipital, Parietal, and Insula); thicker edges represent a higher $S'_{i,j}$ value, in this graph representation ranging from 23,911 to 34,503.

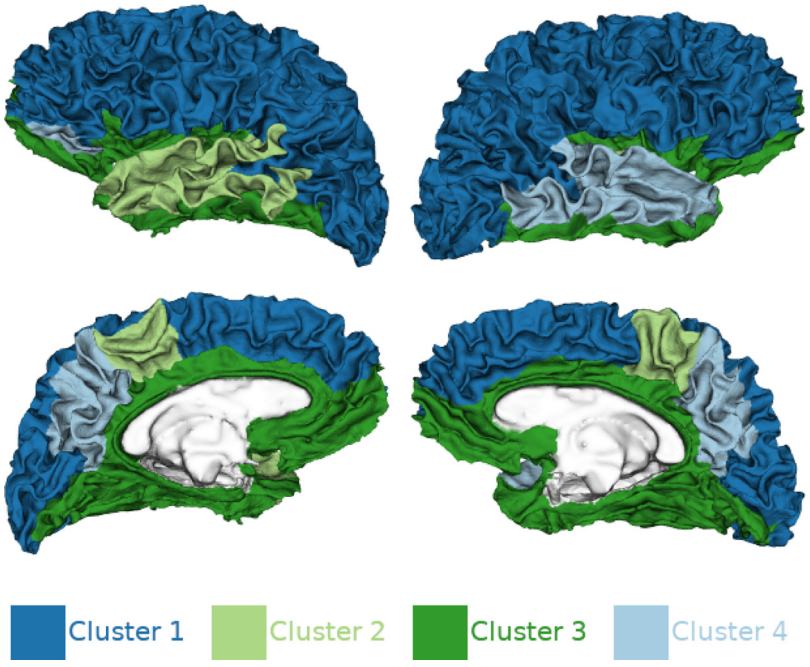

Fig. 7. Four main brain clusters on association matrix $\mathbf{S}'$ generated from N → DiffPool model predicting binary sex on unthresholded matrices. Each colour corresponds to one cluster.

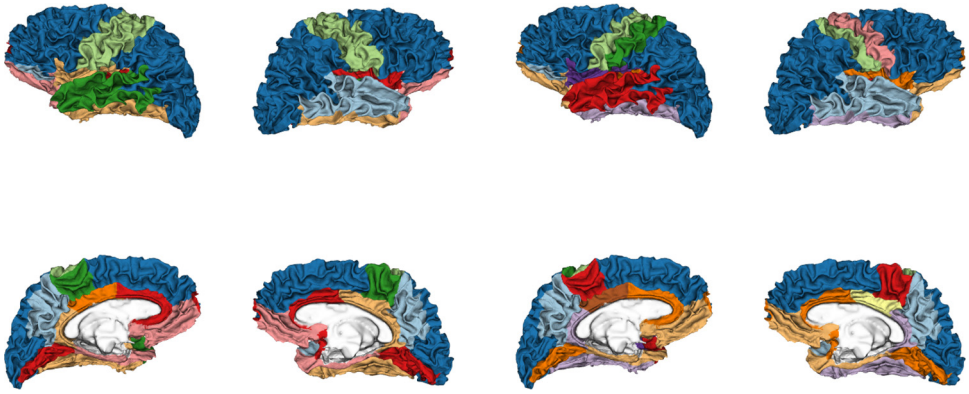

Fig. 8. Main brain clusters on association matrix $\boldsymbol{S}'$ generated from N → DiffPool model predicting binary sex on unthresholded matrices. Each colour corresponds to one cluster. Left: Eight main clusters. Right: Twelve main clusters.

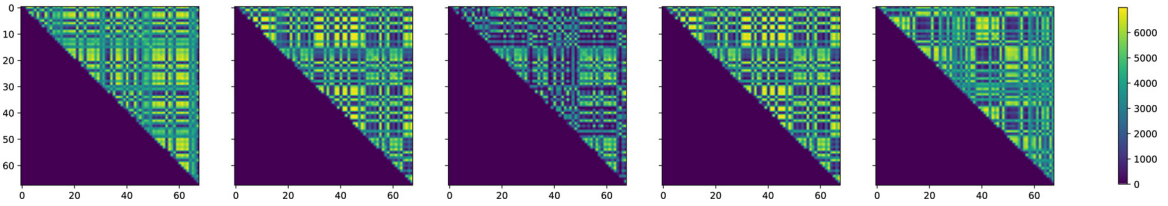

Fig. 9. Association matrices $\boldsymbol{S}'$ for all the five folds for the model "N → DiffPool" trained on the unthresholded matrices.

Table 5
Results when training and evaluating on the HCP dataset, both for the multimodal (rs-fMRI and diffusion data) and unimodal (only rs-fMRI data) cases. Metrics averaged across the five test sets, with standard deviation in parenthesis. Aggregator on the right-hand side of the arrow, "N" corresponds to only *node model*, and "N + E" corresponds to full Graph Network block.

Model	AUC	Accuracy	Sensitivity	Specificity
		no GNN		
→ Concat	0.89 (0.034)	0.81 (0.038)	0.79 (0.050)	0.83 (0.031)
		Using both rs-fMRI and diffusion data		
N + E → Concat	0.94 (0.010)	0.85 (0.016)	0.83 (0.047)	0.87 (0.058)
N + E → DiffPool	0.89 (0.019)	0.81 (0.019)	0.78 (0.047)	0.84 (0.053)
N → Concat	**0.95** (0.012)	**0.88** (0.018)	**0.86** (0.045)	**0.90** (0.026)
N → DiffPool	0.93 (0.018)	0.85 (0.024)	0.79 (0.044)	**0.90** (0.035)
CNSLAB (Gadgil et al., 2020)	0.81 (0.029)	0.74 (0.022)	0.69 (0.051)	0.79 (0.033)
cGCN (Wang et al., 2021)	0.62 (0.019)	0.57 (0.027)	0.51 (0.205)	0.61 (0.220)
XGBoost	0.88 (0.018)	0.81 (0.021)	0.77 (0.036)	0.84 (0.017)
SVM	0.82 (0.020)	0.82 (0.022)	0.79 (0.044)	0.85 (0.058)
		Using only rs-fMRI data		
N + E → Concat	0.88 (0.025)	0.81 (0.030)	**0.80** (0.056)	0.82 (0.037)
N + E → DiffPool	0.63 (0.027)	0.59 (0.012)	0.47 (0.080)	0.70 (0.059)
N → Concat	**0.89** (0.019)	0.82 (0.019)	**0.80** (0.046)	0.83 (0.054)
N → DiffPool	0.68 (0.018)	0.64 (0.014)	0.59 (0.041)	0.68 (0.057)
CNSLAB (Gadgil et al., 2020)	0.82 (0.031)	0.75 (0.023)	0.70 (0.053)	0.79 (0.040)
cGCN (Wang et al., 2021)	0.65 (0.039)	0.59 (0.024)	0.41 (0.175)	0.75 (0.167)
XGBoost	**0.89** (0.014)	0.82 (0.019)	0.78 (0.025)	0.85 (0.030)
SVM	0.83 (0.022)	**0.83** (0.024)	0.78 (0.044)	**0.87** (0.064)

that even DL models perform better in the presence of richer and varied data rather than when merely increasing dataset size, provided the model is able to leverage data richness. This does not happen with the non-DL baselines, which perform almost equally when comparing unimodal and multimodal data.

6. Discussion

In this paper we presented a novel deep learning architecture which can successfully use the high-dimensional and noisy rs-fMRI data, by leveraging their temporal dynamics and spatial associations represented by what is commonly called the connectivity between brain locations. In contrast with previous literature, we use TCNs to model intra-subject temporal dynamics and combine them with GNNs to model inter-regional associations. We illustrated and analysed the effectiveness of our model in a proof-of-concept binary sex prediction task, which also included an ablation analysis with variations of the spatial pooling mechanism. To the best of our knowledge, this work is the first to leverage both the spatial and temporal information in rs-fMRI data in a single, end-to-end framework that: (1) includes temporal convolutions and graph neural networks, and (2) provides the flexibility to extract human-readable, explainability-related patterns which are directly related to the neurobiology and neuroanatomy of the respective brains. We were able to analyse the clusters created by the graph hierarchical pooling mechanism which turned out to carry sensible neurobiological insights. Importantly, we included edge features (i.e., weights) when leveraging the graph structure in the network; this information is often ignored in the few papers which currently apply GNNs to the study of fMRI data (Kim and Ye, 2020). The abla-

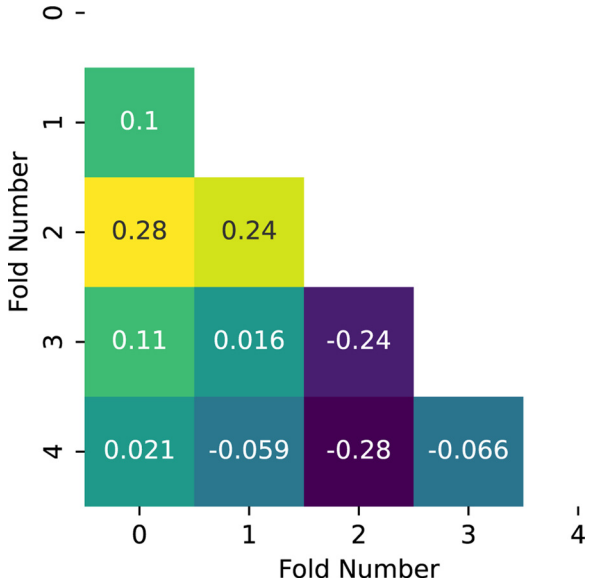

Fig. 10. Averaged normalised differences between association matrices across the five folds of the model "N → DiffPool" depicted in Fig. 9.

tion study showed how the graph network block successfully leveraged the weights of the spatial dynamics, indicating the importance of designing an architecture specifically targeted for spatio-temporal rs-fMRI data. These results point to an advantage of using subject-specific FCMs because the baseline obtained using group-averaged FCMs (i.e. cGCN) consistently performed worse against all other models, including non-DL baselines. Contrary to our initial hypothesis, using a hierarchical pooling mechanism (i.e., DiffPool) did not provide an improvement in overall performance when compared to concatenation pooling and, in some cases, to baselines. The most notable exception is the multimodal setting with the HCP dataset, in which the hierarchical pooling mechanism occasionally provides similar results to our best model. Still, we posit that the compelling explainability potential of DiffPool is advantageous in settings like neuroscience investigation. In this context, additional explorations of hierarchical pooling mechanisms could represent an exciting future research direction.

One of the aims of this paper was to provide additional contributions beyond the goal of end-to-end modelling of functional brain activity, hoping to provide a tool that can be tailored to the analysis of medical images. For instance, the set of experiments using unique multimodal data from the HCP dataset illustrate how our approach can be of interest in the emerging multimodal brain connectivity community. Also, we are not aware of any other work in the neuroimaging field which uses a hierarchical pooling mechanism for the purpose of generating explainable patterns from fMRI data - a crucial aspect when interacting with the neuroscience community. While temporal convolutional networks (TCNs) and graph neural networks (GNNs) have been successfully introduced in previous literature, our contribution in this paper also lies in the combined use of these building blocks for the specific case of modelling rs-fMRI data. Importantly, we have also motivated our choice of TCN kernels with respect to LSTM models through a head-to-head comparison in Section 5.2.

We hope that this paper can lay the groundwork for future exploration of flexible architectures which are able to leverage the entirety of neuromonitoring data arising from the extremely complex spatio-temporal interplay of groups of firing neurons (which, in addition, can only be observed indirectly). By demonstrating improved performance in a proof-of-concept task which is commonly employed in benchmarking models for functional brain data, comparing to both non-DL and DL baselines, and sharing all code and implementation details, we hope that our work will have an impact on future research which will further improve spatio-temporal modelling specific to fMRI data. As we demonstrated with the multimodal Human Connectome Project (HCP) dataset, our architecture can very easily include other types of data (e.g., multimodal structural and temporal data). The architecture can be further extended to include possible confounds (e.g. age, IQ, cognitive status) that could drive the prediction task in other brain disorders. Furthermore, while this is out of the scope of the present paper, additional analyses can be conducted to study the robustness of the architecture to finer parcellations beside the Desikan-Killiany atlas, possibly leading to additional neurobiological insights depending on which regions are represented in the parcellations. Another exciting recent trend that can be included in our architecture is to allow the network to learn the underlying connectivity from scratch (Kazi et al., 2020; Jie et al., 2020) instead of computing associations or other handcrafted features like the ones used in this and other papers (Li et al., 2021; Li et al., 2020).

Data and Code Availability

The code used to process the data from the UK Biobank is publicly available at https://github.com/ucam-department-of-psychiatry/UKB. The code used to conduct the analysis described in this paper is publicly available at https://github.com/tjiagoM/spatio-temporal-brain.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Tiago Azevedo: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Alexander Campbell:** Conceptualization, Methodology, Writing – review & editing. **Rafael Romero-Garcia:** Resources, Data curation. **Luca Passamonti:** Validation, Writing – review & editing. **Richard A.I. Bethlehem:** Resources, Data curation, Writing – review & editing, Visualization. **Pietro Liò:** Conceptualization, Resources, Writing – review & editing, Supervision, Funding acquisition. **Nicola Toschi:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Funding acquisition.

Acknowledgements

T.A. is funded by the W. D. Armstrong Trust Fund, University of Cambridge, UK. R.A.I.B. is funded by a British Academy Post-Doctoral fellowship and the Autism Research Trust. R.R.G is funded by the Guarantors of Brain. L.P. is funded by the Medical Research Council (MRC) grant (MR/P01271X/1) at the University of Cambridge, UK.

The UK Biobank data (application 20904) were curated and analysed using a computational facility funded by an MRC research infrastructure award (MR/M009041/1) and supported by the NIHR Cambridge Biomedical Research Centre and a Marmaduke Shield Award to Dr. Richard A.I. Bethlehem and Varun Warrier. The views expressed are those of the authors and not necessarily those of the NHS, the NIHR or the Department of Health and Social Care.

The multimodal data were provided by the Human Connectome Project, WU-Minn Consortium (PIs: David Van Essen and Kamil

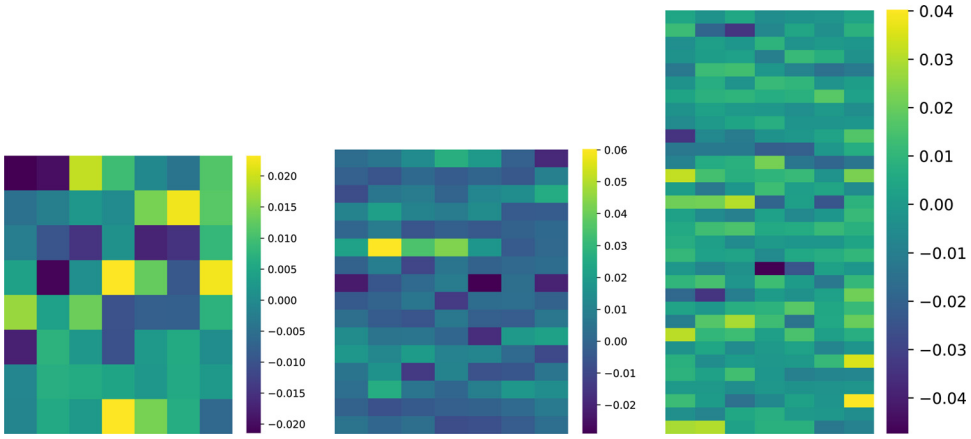

Fig. A.1. Kernels from the first TCN layer of a "N + E → Concat" model trained with three different number of kernels but all remaining hyperparameters the same. Number of kernels from left to right: 8, 16, and 32.

Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

The models in this work were developed and evaluated on two different servers. Runs were performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). In addition, some of the models were trained and evaluated on Titan V GPUs generously donated to N.T. by NVIDIA under the NVIDIA GPU grant program.

Appendix A. Influence of Number of Kernels in First TCN Layer

We selected the hyperparameters of the best "N + E → Concat" model for one fold, and trained three different models with different number of kernels in the first layer (i.e., 8, 16, and 32) while keeping all other hyperparameters fixed. The kernels from the first TCN layer across these three models can be visualised in Fig. A.1. While the explicit significance of the kernel weights is hard to ascertain, the kernel weights appear to have large variability and are therefore likely selecting different, non-redundant patterns present in the original time series. In addition, it is possible to see that the choice of the number of kernels in the first layer influences the patterns learned in this first layer.

References

Allen, E.A., Damaraju, E., Plis, S.M., Erhardt, E.B., Eichele, T., Calhoun, V.D., 2014. Tracking whole-brain connectivity dynamics in the resting state. Cereb. Cortex 24 (3), 663–676.

Arrieta, A.B., Díaz-Rodríguez, N., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion 58, 82–115.

Arslan, S., Ktena, S.I., Glocker, B., Rueckert, D., 2018. Graph saliency maps through spectral convolutional networks: application to sex classification with brain connectivity. In: Lecture Notes in Computer Science. Springer International Publishing, pp. 3–13.

Avena-Koenigsberger, A., Misic, B., Sporns, O., 2017. Communication dynamics in complex brain networks. Nat. Rev. Neurosci. 19 (1), 17–33.

Azevedo, T., Passamonti, L., Lio, P., Toschi, N., 2020. A deep spatiotemporal graph learning architecture for brain connectivity analysis. In: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE.

Bai, S., Kolter, J. Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., Pascanu, R., 2018. Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.

Beckmann, C.F., DeLuca, M., Devlin, J.T., Smith, S.M., 2005. Investigations into resting-state connectivity using independent component analysis. Philos. Trans. R. Soc. B 360 (1457), 1001–1013.

Bengs, M., Gessert, N., Schlaefer, A., 2020. 4D Spatio-temporal deep learning with 4D fMRI data for autism spectrum disorder classification. arXiv preprint arXiv:2004.10165 [cs, eess]

Biewald, L., 2020. Experiment tracking with weights and biases. Software available from https://www.wandb.com/.

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L.T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., Cortes, A., Welsh, S., Young, A., Effingham, M., McVean, G., Leslie, S., Allen, N., Donnelly, P., Marchini, J., 2018. The UK biobank resource with deep phenotyping and genomic data. Nature 562 (7726), 203–209.

Chen, T., Guestrin, C., 2016. XGBoost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. ACM Press.

Chen, Y., Kang, Y., Chen, Y., Wang, Z., 2020. Probabilistic forecasting with temporal convolutional neural network. Neurocomputing 399, 491–501.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., Veličković, P., 2020. Principal neighbourhood aggregation for graph nets. Advances in Neural Information Processing Systems.

Desikan, R.S., Ségonne, F., Fischl, B., Quinn, B.T., Dickerson, B.C., Blacker, D., Buckner, R.L., Dale, A.M., Maguire, R.P., Hyman, B.T., Albert, M.S., Killiany, R.J., 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. Neuroimage 31 (3), 968–980.

Duggento, A., Guerrisi, M., Toschi, N., 2019. Recurrent neural networks for reconstructing complex directed brain connectivity. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE.

Duggento, A., Passamonti, L., Valenza, G., Barbieri, R., Guerrisi, M., Toschi, N., 2018. Multivariate Granger causality unveils directed parietal to prefrontal cortex connectivity during task-free MRI. Sci. Rep. 8 (1).

Dvornek, N.C., Ventola, P., Pelphrey, K.A., Duncan, J.S., 2017. Identifying autism from resting-state fMRI using long short-term memory networks. In: Machine Learning in Medical Imaging. Springer International Publishing, pp. 362–370. doi:10.1007/978-3-319-67389-9_42.

El-Gazzar, A., Quaak, M., Cerliani, L., Bloem, P., van Wingen, G., Thomas, R. M., 2020. A hybrid 3DCNN and 3DC-LSTM based model for 4D spatio-temporal fMRI data: an ABIDE autism classification study. arXiv preprint arXiv:2002.05981 [cs]

Elbayad, M., Besacier, L., Verbeek, J., 2018. Pervasive attention: 2D convolutional neural networks for sequence-to-sequence prediction. arXiv preprint arXiv:1808.03867.

Eslami, T., Mirjalili, V., Fong, A., Laird, A.R., Saeed, F., 2019. ASD-DiagNet: a hybrid learning approach for detection of autism spectrum disorder using fMRI data. Front. Neuroinform. 13, 70.

Fan, L., Su, J., Qin, J., Hu, D., Shen, H., 2020. A deep network model on dynamic functional connectivity with applications to gender classification and intelligence prediction. Front. Neurosci. 14, 881.

Fey, M., Lenssen, J.E., 2019. Fast graph representation learning with PyTorch geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds.

Fornito, A., Zalesky, A., Breakspear, M., 2015. The connectomics of brain disorders. Nat. Rev. Neurosci. 16 (3), 159–172.

Gadgil, S., Zhao, Q., Pfefferbaum, A., Sullivan, E.V., Adeli, E., Pohl, K.M., 2020. Spatio-temporal graph convolution for resting-state fMRI analysis. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. Springer International Publishing, pp. 528–538.

Garrison, K.A., Scheinost, D., Finn, E.S., Shen, X., Constable, R.T., 2015. The (in)stability of functional brain network measures across thresholds. Neuroimage 118, 651–661.
Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N., 2017. Convolutional sequence to sequence learning. In: Precup, D., Teh, Y.W. (Eds.), Proceedings of the 34th International Conference on Machine Learning. PMLR, pp. 1243–1252.
Glasser, M.F., Sotiropoulos, S.N., Wilson, J.A., Coalson, T.S., Fischl, B., Andersson, J.L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J.R., Essen, D.C.V., Jenkinson, M., 2013. The minimal preprocessing pipelines for the human connectome project. Neuroimage 80, 105–124.
Goelman, G., Dan, R., Keadan, T., 2018. Characterizing directed functional pathways in the visual system by multivariate nonlinear coherence of fMRI data. Sci. Rep. 8 (1). doi:10.1038/s41598-018-34672-5.
Heinsfeld, A.S., Franco, A.R., Craddock, R.C., Buchweitz, A., Meneguzzi, F., 2017. Identification of autism spectrum disorder using deep learning and the ABIDE dataset. NeuroImage 17, 16–23.
Hilgetag, C.C., Goulas, A., 2020. 'Hierarchy' in the organization of brain networks. Philos. Trans. R. Soc. B 375 (1796), 20190319.
Hutchison, R.M., Womelsdorf, T., Allen, E.A., Bandettini, P.A., Calhoun, V.D., Corbetta, M., Penna, S.D., Duyn, J.H., Glover, G.H., Gonzalez-Castillo, J., Handwerker, D.A., Keilholz, S., Kiviniemi, V., Leopold, D.A., de Pasquale, F., Sporns, O., Walter, M., Chang, C., 2013. Dynamic functional connectivity: promise, issues, and interpretations. Neuroimage 80, 360–378.
Jeurissen, B., Tournier, J.-D., Dhollander, T., Connelly, A., Sijbers, J., 2014. Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data. Neuroimage 103, 411–426.
Jiang, R., Calhoun, V.D., Fan, L., Zuo, N., Jung, R., Qi, S., Lin, D., Li, J., Zhuo, C., Song, M., Fu, Z., Jiang, T., Sui, J., 2019. Gender differences in connectome-based predictions of individualized intelligence quotient and sub-domain scores. Cereb. Cortex 30, 888–900.
Jie, B., Liu, M., Lian, C., Shi, F., Shen, D., 2020. Designing weighted correlation kernels in convolutional neural networks for functional connectivity based brain disease diagnosis. Med. Image Anal. 63, 101709.
Kaiser, L., Gomez, A.N., Chollet, F., 2018. Depthwise separable convolutions for neural machine translation. In: International Conference on Learning Representations.
Kalchbrenner, N., Espeholt, L., Simonyan, K., van den Oord, A., Graves, A., Kavukcuoglu, K., 2016. Neural machine translation in linear time. arXiv preprint arXiv:1610.10099.
Kazi, A., Cosmo, L., Navab, N., Bronstein, M., 2020. Differentiable graph module (DGM) for graph convolutional networks. arXiv preprint arXiv:2002.04999.
Kiebel, S.J., Daunizeau, J., Friston, K.J., 2008. A hierarchy of time-scales and the brain. PLoS Comput. Biol. 4 (11), e1000209.
Kim, B.-H., Ye, J.C., 2020. Understanding graph isomorphism network for rs-fMRI functional connectivity analysis. Front. Neurosci. 14.
Ktena, S.I., Parisot, S., Ferrante, E., Rajchl, M., Lee, M., Glocker, B., Rueckert, D., 2018. Metric learning with spectral graph convolutions on brain connectivity networks 169, 431–442.
Li, X., Zhou, Y., Dvornek, N., Zhang, M., Gao, S., Zhuang, J., Scheinost, D., Staib, L.H., Ventola, P., Duncan, J.S., 2021. BrainGNN: Interpretable brain graph neural network for fMRI analysis, 74. Elsevier, p. 102233.
Liao, X., Cao, M., Xia, M., He, Y., 2017. Individual differences and time-varying features of modular brain architecture. Neuroimage 152, 94–107.
Marek, S., Tervo-Clemmens, B., Calabro, F.J., Montez, D.F., Kay, B.P., Hatoum, A.S., Donohue, M.R., Foran, W., Miller, R.L., Feczko, E., Miranda-Dominguez, O., Graham, A.M., Earl, E.A., Perrone, A.J., Cordova, M., Doyle, O., Moore, L.A., Conan, G., Uriarte, J., Snider, K., Tam, A., Chen, J., Newbold, D.J., Zheng, A., Seider, N.A., Van, A.N., Laumann, T.O., Thompson, W.K., Greene, D.J., Petersen, S.E., Nichols, T.E., Yeo, B.T., Barch, D.M., Garavan, H., Luna, B., Fair, D.A., Dosenbach, N.U., 2020. Towards reproducible brain-wide association studies https://www.biorxiv.org/content/10.1101/2020.08.21.257758v1.
Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M., 2019. Weisfeiler and Leman go neural: higher-order graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, pp. 4602–4609.
Nieuwenhuys, R., Voogd, J., van Huijzen, C., 2008. The Human Central Nervous System. Springer Berlin Heidelberg.
Parisot, S., Ktena, S.I., Ferrante, E., Lee, M., Moreno, R.G., Glocker, B., Rueckert, D., 2017. Spectral graph convolutions for population-based disease prediction. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (Eds.), Medical Image Computing and Computer Assisted Intervention - MICCAI 2017. Springer International Publishing, Cham, pp. 177–185.
Parmar, H., Nutter, B., Long, R., Antani, S., Mitra, S., 2020. Spatiotemporal feature extraction and classification of Alzheimer's disease using deep learning 3D-CNN for fMRI data. J. Med. Imaging 7 (5), 056001.
Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: an imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.
, 2020. In: Poeppel, D., Mangun, G.R., Gazzaniga, M.S. (Eds.), The Cognitive Neurosciences, sixth ed.. The MIT Press, Cambridge, MA.
Preti, M.G., Bolton, T.A., Van De Ville, D., 2017. The dynamic functional connectome: state-of-the-art and perspectives. Neuroimage 160, 41–54.
Riaz, A., Asad, M., Alonso, E., Slabaugh, G., 2020. DeepFMRI: end-to-end deep learning for functional connectivity and classification of ADHD using fMRI. J. Neurosci. Methods 335, 108506.
Salimi-Khorshidi, G., Douaud, G., Beckmann, C.F., Glasser, M.F., Griffanti, L., Smith, S.M., 2014. Automatic denoising of functional MRI data: combining independent component analysis and hierarchical fusion of classifiers. Neuroimage 90, 449–468.
Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R., 2019. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Vol. 11700. Springer Nature.
Smith, R.E., Tournier, J.-D., Calamante, F., Connelly, A., 2013. SIFT: spherical-deconvolution informed filtering of tractograms. Neuroimage 67, 298–312.
Smith, S.M., Nichols, T.E., 2018. Statistical challenges in "big data" human neuroimaging. Neuron 97 (2), 263–268.
Spasov, S., Passamonti, L., Duggento, A., Liò, P., Toschi, N., 2019. A parameter-efficient deep learning approach to predict conversion from mild cognitive impairment to Alzheimer's disease. Neuroimage 189, 276–287.
Sporns, O., 2011. The human connectome: a complex network. Ann. N. Y. Acad. Sci. 1224 (1), 109–125.
Sporns, O., 2018. Graph theory methods: applications in brain networks. Dialogues Clin. Neurosci. 20 (2), 111–121.
Tieleman, T., Hinton, G., et al., 2012. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. Mach. Learn. 4 (2), 26–31.
Li, X., Zhou, Y., Dvornek, N. C., Zhang, M., Zhuang, J., Ventola, P., Duncan, J. S., 2020. Pooling regularized graph neural network for fMRI biomarker analysis-arXiv preprint arXiv:2007.14589.
van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. WaveNet: a generative model for raw audio. arXiv preprint arXiv:1609.03499.
Wang, H.E., Bénar, C.G., Quilichini, P.P., Friston, K.J., Jirsa, V.K., Bernard, C., 2014. A systematic framework for functional connectivity measures. Front. Neurosci. 8.
Wang, L., Li, K., Chen, X., Hu, X.P., 2019. Application of convolutional recurrent neural network for individual recognition based on resting state fMRI data. Front. Neurosci. 13.
Wang, L., Li, K., Hu, X.P., 2021. Graph convolutional network for fMRI analysis based on connectivity neighborhood. Netw. Neurosci. 5 (1), 83–95.
Waskom, M.L., 2021. seaborn: statistical data visualization. J. Open Source Softw. 6 (60), 3021. doi:10.21105/joss.03021.
Weis, S., Patil, K.R., Hoffstaedter, F., Nostro, A., Yeo, B.T.T., Eickhoff, S.B., 2019. Sex classification by resting state brain connectivity. Cereb. Cortex 30, 824–835.
Wen, D., Wei, Z., Zhou, Y., Li, G., Zhang, X., Han, W., 2018. Deep learning methods to process fMRI data and their application in the diagnosis of cognitive impairment: a brief overview and our opinion. Front. Neuroinform. 12.
Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P. S., 2019. A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596.
Yan, B., Xu, X., Liu, M., Zheng, K., Liu, J., Li, J., Wei, L., Zhang, B., Lu, H., Li, B., 2020. Quantitative identification of major depression based on resting-state dynamic functional connectivity: a machine learning approach. Front. Neurosci. 14.
Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J., 2018. Hierarchical graph representation learning with differentiable pooling. In: Advances in Neural Information Processing Systems, pp. 4800–4810.
Yu, F., Koltun, V., 2016. Multi-scale context aggregation by dilated convolutions. ICLR.
Zhan, L., Jenkins, L.M., Wolfson, O.E., GadElkarim, J.J., Nocito, K., Thompson, P.M., Ajilore, O.A., Chung, M.K., Leow, A.D., 2017. The significance of negative correlations in brain connectivity. J. Comp. Neurol. 525 (15), 3251–3265.
Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2018. Graph neural networks: a review of methods and applications. arXiv preprint arXiv:1812.08434.