EEG-based Graph Neural Network Classification of Alzheimer's Disease: An Empirical Evaluation of Functional Connectivity Methods

Dominik Klepl,[1] Fei He,[2,*] Min Wu,[3] Daniel J. Blackburn,[4] and Ptolemaios G. Sarrigiannis[5]

[1] *Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry CV1 2JH, UK*
[2] *Centre for Computational Science and Mathematical Modelling, Coventry University, Coventry CV1 2JH, UK*
[3] *Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), 138632, Singapore*
[4] *Department of Neuroscience, University of Sheffield, Sheffield, S10 2HQ, UK*
[5] *Department of Neurophysiology, Royal Devon and Exeter NHS Foundation Trust, Exeter, EX2 5DW, UK*

Alzheimer's disease (AD) is the leading form of dementia worldwide. AD disrupts neuronal pathways and thus is commonly viewed as a network disorder. Many studies demonstrate the power of functional connectivity (FC) graph-based biomarkers for automated diagnosis of AD using electroencephalography (EEG). However, various FC measures are commonly utilised as each aims to quantify a unique aspect of brain coupling. Graph neural networks (GNN) provide a powerful framework for learning on graphs. While a growing number of studies use GNN to classify EEG brain graphs, it is unclear which method should be utilised to estimate the brain graph. We use eight FC measures to estimate FC brain graphs from sensor-level EEG signals. GNN models are trained in order to compare the performance of the selected FC measures. Additionally, three baseline models based on literature are trained for comparison. We show that GNN models perform significantly better than the other baseline models. Moreover, using FC measures to estimate brain graphs improves the performance of GNN compared to models trained using a fixed graph based on the spatial distance between the EEG sensors. However, no FC measure performs consistently better than the other measures. The best GNN reaches 0.984 area under sensitivity-specificity curve (AUC) and 92% accuracy, whereas the best baseline model, a convolutional neural network, has 0.924 AUC and 84.7% accuracy.

I. INTRODUCTION

Alzheimer's disease (AD), a neurodegenerative disease, is the most common form of dementia. AD patients exhibit progressive deterioration of memory and other cognitive functions. From a neuroscience perspective, AD leads to synaptic loss and cellular death, which progressively occurs over multiple brain regions [1]. Disruption of communication pathways amongst brain regions is observed in AD [2–4]. Due to this distributed nature of AD, it can be recognised as a network disorder. Thus, graph theory is well suited for analysing and classifying AD as it provides a general framework for studying various pathological processes' interactions across multiple spatiotemporal scales.

Functional connectivity (FC) is one of the methods to construct and study brain graphs. The edges of FC graphs represent the statistical dependencies between brain regions rather than the physical connectome, i.e. structural connectivity. FC brain graphs can be constructed from any functional brain imaging modality, such as M/EEG, fMRI or PET. In this paper, we focus on EEG. EEG has been shown to be an effective tool for studying the changes in brain activity in AD cases [5–7]. Compared to other modalities, EEG is economical, non-invasive, easy to administer and has a superior temporal resolution. On the other hand, it suffers from a low spatial resolution as the activity is measured by means of electrodes placed on the scalp of the subject.

Emerging evidence shows large-scale alterations in functional connectivity (FC) in AD, such as increased connectivity in the low-frequency bands [8–10]. Graph-based studies show that AD is characterised by reduced complexity [6] and loss of small-world organisation, assessed by the clustering coefficient and characteristic path length [11–14].

However, there are multiple FC methods commonly used within the literature. Furthermore, each FC method may quantify a different aspect of brain coupling. There are both linear and nonlinear FC methods that quantify the coupling of signals' phases, amplitudes and even cross-frequency interactions. For simplicity, we do not consider cross-frequency interactions and focus on FC measures quantifying brain coupling between two signals within the same frequency range, i.e. frequency band. Moreover, there is an uncertainty about reproducibility of some FC methods [15] and robustness to volume conduction effects [16]. In this paper, we select and evaluate a number of commonly used methods to quantify FC from EEG data.

The most common and simplest FC measure is the Pearson's correlation coefficient (corr) [17]. The phase coupling of a pair of signals is commonly measured by coherence (coh) [13, 14], imaginary part of coherency (iCOH) [15, 18], phase lagged index (PLI) and its weighted version (wPLI), and phase locking value (PLV) [15]. The correlation of amplitude envelopes of two signals can also be measured (amp-corr) [15]. Finally,

* Correspondence to: fei.he@coventry.ac.uk

information-theoretic measures are commonly used to measure FC, such as mutual information (MI) [19]. We exclude the directed (i.e. causal) connectivity measures from this comparative study, as these are generally not considered to be FC measures but rather effective connectivity measures.

Graph-based features were also successfully used to train machine learning classifiers to diagnose brain disorders using EEG automatically. Manually engineered graph features such as node strength [20] and vectorised adjacency matrix [21] seem to be promising graph-based biomarkers of AD, as both approaches achieve high classification accuracy. Additionally, there were some attempts to utilise deep learning, convolutional neural network (CNN), for automatic graph-based feature extraction. Specifically, CNN was trained to classify AD and schizophrenia using adjacency matrices which are image-like representations of FC graphs [22]. However, an image representation of a graph cannot effectively capture all the properties, as a graph is a non-euclidean object.

Graph neural network (GNN) extends the logic of convolution operation to graphs by aggregating information from linked nodes, based on the assumption that nodes connected by an edge are similar. However, there is a limited number of GNN applications for EEG brain graph classification. Moreover, it is unclear which method should be used to infer the graph structure for GNN application. A fully connected graph is commonly used in the literature [23]. However, such an approach does not leverage any information encoded by FC brain graphs. A second option is using the distances between spatial positions of EEG electrodes to define the graph structure [23, 24]. Furthermore, Demir et al. [23] utilise distance thresholding and k-nearest-neighbours methods to filter-out unimportant edges. Such edge-filtering can be important as some edges might be redundant or even introduce additional noise, thus hindering the model from learning optimal solution. Only a handful of studies use FC measures such as coherence [24] and wPLI [25]. Additionally, Liu et al. [25] use a minimum spanning tree algorithm (MST) in order to produce sparse brain graphs. This is in contrast to threshold-based edge filtering, as MST can select edges with various edge weights and ensures that the resulting graph is connected. In contrast, Zhong et al. [26] utilise learnable mask in order to learn the optimal graph structure for a specific classification task without relying on any FC measure.

In this study, we systematically evaluate the effects of using various FC methods to infer EEG brain graphs in training GNN for the classification of AD patients. Two types of graph filtering are used to induce graph-sparsity in order to improve the performance of GNN. To compare and evaluate the classification performance of various FC-based GNNs, a GNN-based baseline is trained using a fixed graph structure for all brain graphs, represented by the euclidean distance between spatial positions of EEG sensors. Three additional baseline models are established: two SVM baselines fitted on node strength (SVM-NS) and vectorised adjacency matrix (SVM-vector), respectively, and a CNN trained on images of adjacency matrices. Fig. 1 shows the considered model architectures.

II. DATA AND PRE-PROCESSING

The EEG dataset consists of 20 AD patients and 20 healthy control participants (HC) below 70 years. A subset of this dataset has been previously used in Blackburn et al. [5]. All AD participants were recruited in the Sheffield Teaching Hospital memory clinic. AD participants were diagnosed between one month and two years before data collection, and all were in the mild to moderate stage of the disease at the time of recording. Age and gender-matched HC participants with normal neuropsychological tests and structural MRI scans were recruited. The EEG data used in this study was approved by the Yorkshire and The Humber (Leeds West) Research Ethics Committee (reference number 14/YH/1070). All participants gave their informed written consent.

EEG was acquired using an XLTEK 128-channel headbox, Ag/AgCL electrodes with a sampling frequency of 2 kHz using a modified 10-10 overlapping a 10-20 international electrode placement system with a referential montage with a linked earlobe reference. The recordings lasted 30 minutes, during which the participants were instructed to rest and not to think about anything specific. Within each recording, there were two-minute-long epochs during which the participants had their eyes closed (alternating with equal duration eyes-open epochs, not used in this work).

All the recordings were reviewed by an experienced neurophysiologist on the XLTEK review station with time-locked video recordings (Optima Medical LTD). For each participant, three 12-second-long artefact-free epochs were isolated. Finally, the following 23 bipolar channels were created: F8–F4, F7–F3, F4–C4, F3–C3, F4–FZ, FZ–CZ, F3–FZ, T4–C4, T3–C3, C4–CZ, C3–CZ, CZ–PZ, C4–P4, C3–P3, T4–T6, T3–T5, P4–PZ, P3–PZ, T6–O2, T5–O1, P4–O2, P3–O1 and O1–O2 [5].

A. EEG Pre-processing

First, a zero-phase 5th order Butterworth filter is employed to remove frequencies below 0.1 Hz and above 100 Hz; a zero-phase 4th order Butterworth stop-band filter is used to remove frequencies between 49 and 51 Hz related to power-noise. The EEG data were then down sampled to 250 Hz using an 8th order Chebyshev type I filter and scaled to zero mean and unit standard deviation.

In order to increase the sample size and to demonstrate that the classification performance is epoch independent, the 12-seconds-long epochs were split into 3-second-long non-overlapping segments. Thus, for each subject there

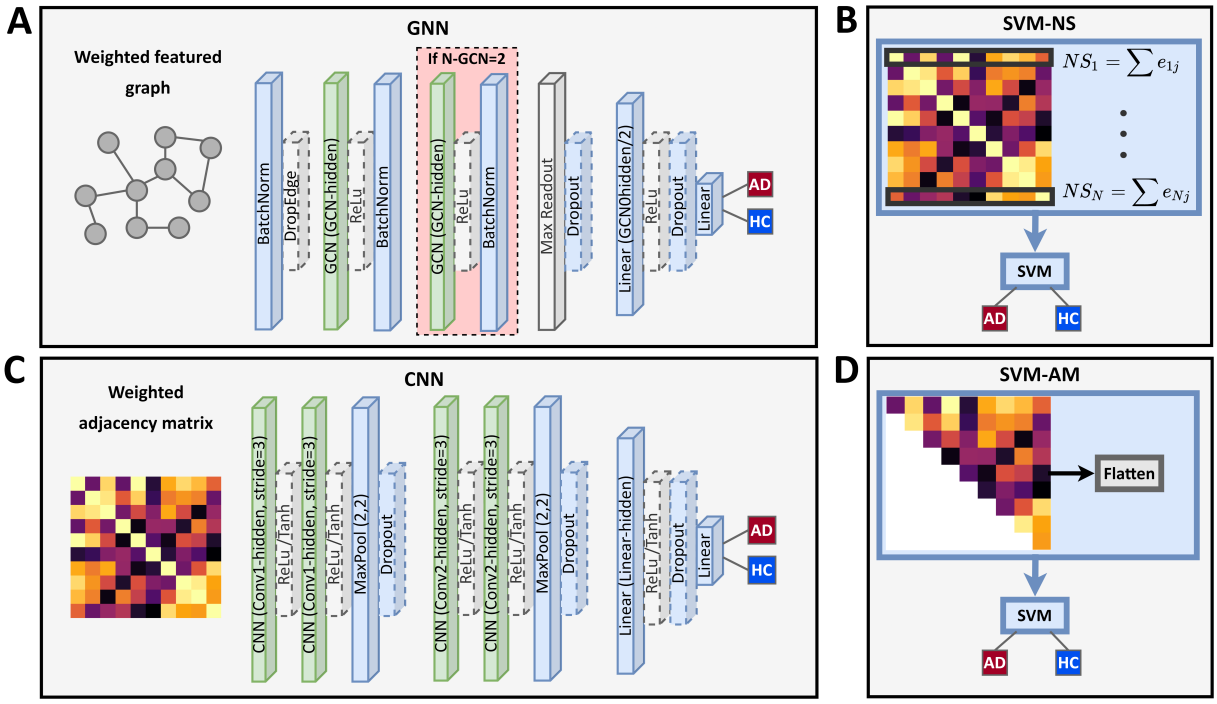

FIG. 1: Overview of model architectures developed for classification of AD from EEG-FC-brain graphs. (A) A graph neural network (GNN) takes weighted featured brain graphs where the node features consist of power spectral densities ($0-100Hz$). The N-GCN hyper-parameter controls the number of graph convolutional layers. (B) Support vector machine trained using the node strengths (i.e. the sum of edge weights of neighbouring nodes) as input features (SVM-NS). (C) Convolutional neural network (CNN) trained on the brain graphs represented by weighted adjacency matrices. (D) The flattened upper triangular part of the weighted adjacency matrices is used to train a support vector machine (SVM-AM).

are 12 EEG segments. Finally, frequency bands are created from each EEG segment using a zero-phase 5th order Butterworth filter. Six frequency bands are considered: δ $(0.5-4Hz)$, θ $(4-7Hz)$, α $(7-15Hz)$, β $(15-31Hz)$, γ $(31-100Hz)$ and full $(0.5-100Hz)$.

III. METHODS

A. Functional-Connectivity-based Brain Graph Inference

In this paper, we selected eight commonly used methods for constructing brain graphs from EEG signals. Namely, these are: the absolute value of Pearson's correlation (corr), spectral coherence (coh), the imaginary part of coherency (iCOH), phase lagged index (PLI), weighted phase lagged index (wPLI), phase locking value (PLV), mutual information (MI) and amplitude envelope correlation (AEC).

We estimate FC brain graphs for each EEG segment and frequency band separately. Thus, for each subject, we obtain 72 brain graphs (12 segments $\times$ 6 frequency bands). A brain graph G can be represented by an $N \times N$ adjacency matrix A where $N = 23$. As we consider only FC measures, all edges are undirected, and thus the number of inferred edges can be reduced from N^2 to $[N \times (N-1)/2]$. However, for simplicity, we keep the N^2 edges in the $N \times N$ adjacency matrix A. Thus, each entry of the adjacency matrix A^{FC}_{xy} represents the edge weight between nodes, i.e. the EEG channels x and y are measured by the connectivity measure FC. All of the selected measures are normalised to $[0,1]$ where 0 indicates no coupling and 1 indicates a perfect coupling.

The adjacency matrix using the absolute values of Pearson's correlation coefficients between nodes x and y is given by:

$$A^{corr}_{xy} = \left| \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \right|, \qquad (1)$$

where $\bar{x}$ is the mean of x. The absolute value is calculated as we are only interested in the coupling magnitude. Next, the adjacency matrix of coherence is given by:

$$A^{coh}_{xy}(f) = \frac{|CS_{xy}(f)|^2}{CS_{xx}(f)CS_{yy}(f)}, \qquad (2)$$

where CS_{xy} and CS_{xx} are cross-spectral and auto-spectral densities respectively. The coherence within of a frequency band B is then calculated as mean of $G^{coh}_{xy}(f)$ where $f \in B$.

The imaginary part of coherency (iCOH) measures phase consistency similar to coh and accounts for volume

conduction effects. The adjacency matrix using iCOH is computed as:

$$A_{xy}^{iCOH}(f) = \frac{\Im(CS_{xy}(f))}{\sqrt{CS_{xx}(f)CS_{yy}(f)}}, \tag{3}$$

where $\Im$ denotes the imaginary component. The iCOH within a frequency band B is then calculated as the mean of $G_{xy}^{iCOH}(f)$ where $f \in B$

The phase and amplitude of an EEG signal can be calculated from the analytic representation z of signal x

$$z(t) = x(t) + \mathrm{i}\,\tilde{x}(t), \tag{4}$$

where i is the imaginary component and $\tilde{x}(t)$ is the corresponding Hilbert transform. Then the phase and amplitude can be obtained from $z(t)$ as

$$\phi(t) = \arctan\left(\frac{\tilde{x}(t)}{x(t)}\right), \tag{5}$$

and

$$amp(t) = \sqrt{[x(t)]^2 + [\tilde{x}(t)]^2}. \tag{6}$$

Phase lag index (PLI) quantifies the asymmetry in phase distributions of two signals and measures only non-zero phase locking [27]. The adjacency matrix using PLI is defined as:

$$A_{xy}^{PLI} = \left|\frac{1}{T}\sum_{t=1}^{T} \operatorname{sign}\sin(\phi_x(t) - \phi_y(t))\right|, \tag{7}$$

where ϕ_x is obtained using Eq. 5. Weighted phase lag index (wPLI) is an extension of PLI, which aims to remove the effects of amplitude and volume conduction by maximally weighting the ± 90 deg phase differences and thus omitting uniformly driven differences [28]. The adjacency matrix using wPLI is computed as

$$A_{xy}^{wPLI} = \left|\frac{1}{T}\sum_{t=1}^{T} \frac{|\sin(\phi_x(t) - \phi_y(t))|}{\sin(\phi_x(t) - \phi_y(t))}\right|. \tag{8}$$

Phase locking value (PLV) is another approach to quantify the consistency of phase differences between signals, and its associated adjacency matrix is computed as

$$A_{xy}^{PLV} = \left|\frac{1}{T}\sum_{t=1}^{T} \mathrm{e}^{-\,\mathrm{i}(\phi_x(t) - \phi_y(t)}\right|. \tag{9}$$

Amplitude envelope correlation (AEC) aims to quantify the coupling based on the amplitudes of the signals. The adjacency matrix using AEC is computed with Eq. 1 where x and y are the amplitudes of respective signals computed using Eq. 6.

Mutual information (MI) quantifies the amount of known information about a second signal after observing the first signal. The adjacency matrix using MI is calculated as:

$$A_{xy}^{MI} = \sum_{x_i, y_j} P_{XY}(x_i, y_j) \log\left(\frac{P_{XY}(x_i, y_j)}{P_X(x_i)P_Y(y_j)}\right), \tag{10}$$

where P_{XY} and P_X are the joint and marginal probability distributions, respectively.

1. *Edge filtering methods*

It is worth noting that we did not use any corrections for false positives. Thus, the true brain graph structure might be masked by noise due to spurious coupling. Traditionally, a surrogate threshold might be used to control such spurious edges. However, such a procedure is computationally expensive as it requires re-computing the connectivity measure on multiple random surrogate versions of the original signals to estimate a null surrogate distribution. Instead, we implement two edge-filtering methods to select only important edges and thus produce sparse graphs. Compared to the surrogate threshold method, edge-filtering is a fast and efficient, albeit naive method to deal with potentially noisy brain graphs. We also utilise the fully-connected graphs, i.e. without any edge selection, in the classification models in order to test the effect of edge-filtering.

The first edge-filtering method is an FC-strength-based top-k% filter ($k \in \{10, 20, 30\}$), which selects only the top k% strongest edges of the given graph and removes the rest. This approach assumes that edge weight, i.e. the connectivity strength, is directly related to the importance of an edge. However, this assumption might not be valid.

A minimum-spanning-tree-based filter (MST-k) addresses this concern as it selects a mix of edge weights and always produces a connected graph, i.e. a path exists among all nodes. Briefly, the MST algorithm aims to extract a backbone of a graph with N nodes by selecting $N-1$ edges such that the sum of weights is minimised. In the case of brain graphs, a stronger edge weight implies a higher degree of coupling; thus, we use an inverted MST algorithm which maximises the sum of weights instead. When $k = 1$, MST-k is equal to a single iteration of the MST algorithm. For $k > 1$, the edges selected by the previous iterations are removed from the graph, and the MST algorithm is re-run. Thus, MST-k filter selects $k(N-1)$ edges.

B. Graph Neural Network Classification

A graph neural network (GNN) is an extension of an artificial neural network that is capable of learning on graph-structured data. Specifically, we implement a graph convolutional network (GCN) for a graph classification task (Fig. 1A).

The input to the GCN classifier is in the form of a graph: $G = \{N, E, F\}$, where N, E, and F are sets of nodes, edges and node features. The nodes are fixed in our case, as this is the number of EEG electrodes. The set of edges E is given by the adjacency matrix A computed by the FC measures introduced in the previous section.

Hyper-parameter	Values	GNN	SVM-NS	SVM-AM	CNN
Cost	{0,1}		✓	✓	
Batch size	{16, 32, 64, 128, 256}	✓			✓
Learning rate	{0.0001, 0.001, 0.01, 0.1}	✓			✓
GCN-hidden	{256, 516, 1024, 2048, 4096}	✓			
N-GCN	{1,2}	✓			
Conv1-hidden	{16, 32, 64, 128}				✓
Conv2-hidden	{16, 32, 64, 128}				✓
Linear-hidden	[32, 64, 96, 160, 256, 516]				✓
Activation/Kernel	{ReLU, Tanh}				✓
DropEdge	{True, False}	✓			
Drop-rate	{0.1, 0.25, 0.5, 0.75, 0.8, 0.9}	✓			✓
Frequency band	δ, θ, α, β, γ, full	✓	✓	✓	✓
Graph filter	full, top-{10,20,30}, MST-{1,2,3}	✓	✓	✓	✓
Gamma	{0.8, 0.85, 0.9, 0.95}	✓			✓

TABLE I: Possible values of hyper-parameters

Finally, the node feature matrix F is an $N \times D$ matrix where each row encodes a D-dimensional feature for the corresponding node. Specifically, spectral power density computed over 1 Hz increments in an interval between 0 and 100 Hz, forming a 100-dimensional node feature vector (i.e. D=100).

GCN is based on the message-passing framework, which assumes that neighbouring nodes are similar in terms of their node features. Briefly, a GCN layer updates the node features (i.e. messages) using the optionally transformed messages collected from neighbouring nodes. On a node level, a single GCN layer effectively aggregates information from the 1-hop neighbourhood of each node. Thus, stacking L GCN layers represents aggregation from L-hop neighbourhood. Formally, the GCN layer is implemented on a node-level as follows [29]:

$$x_i^l = \Theta_1 x_i^{l-1} + \Theta_2 \max_{j \in G_i} e_{ij} x_j^{l-1}, \tag{11}$$

where x_i^l is the node features of node i at the lth layer, x_i^0 is the ith row of the input node feature matrix F, and Θ is a learnable linear transformation which maps the node features from shape $[1, D]$ to [1, GCN-hidden]. G_i and e_{ij} are the neighbourhood of node i and the edge weight connecting nodes i and j given by the set of edges E respectively. The GCN-hidden is a tunable hyper-parameter of the GCN architecture. A rectified-linear-unit (ReLU) activation is applied to the output of GCN and batch normalisation is performed [30]. We refer to the node-wise outputs of GCN as node embeddings.

After L GCN layers are applied, the output is constructed by node embeddings in the form of a $N \times H$ matrix, where H is the hidden size given by GCN-hidden. In order to produce a graph-level embedding, a maximum readout layer is applied, resulting in an H-dimensional graph embedding r for each graph g.

$$r_g = \max_{i=1}^{N} x_i^L, \tag{12}$$

where x_i^L is the output of the Lth GCN layer for the ith node. Following the readout layer, two linear layers are applied to produce the final classification with output dimensions $H/2$ and 2 (number of classes), respectively. Two linear layers were used to allow for further refining of the graph embedding before outputting the predicted class probabilities.

Additionally, in order to improve the generalisability and reduce the risk of overfitting, dropout layers are utilised (1A). Briefly, the dropout layer randomly zeroes elements of the input tensor with p probability drawn from a binomial distribution, where p is a hyper-parameter. A dropout is applied to the graph embeddings, i.e. after the readout layer and after the first linear layer. Furthermore, an edge dropout is implemented, which randomly removes edges from the input graph. The inclusion of the edge dropout in the model is controlled by a hyper-parameter.

In summary, the GNN used in this study has several hyper-parameters, as shown in Table I, which control (1) the model architecture, (2) the form of input data and (3) the training process and prevent overfitting. In particular, (1) is enabled by the number of GCN layers (N-GCN) and the inclusion of edge dropout (DropEdge). (2) is enabled by frequency band and graph filter. Finally, (3) is enabled by dropout probability (drop-p), learning rate, gamma and batch size.

C. Baseline Models

In order to enable a fair assessment of the advantages of using graph-based learning (i.e. the GNN), three baseline classifiers are trained and compared with. These baseline models utilise the same graph-structured input data extracted using different FC measures, frequency bands and graph filters, and the same evaluation process. Thus, we argue this to be a fair comparison of models.

The three selected baseline models are based on previously used classifier strategies for learning on FC brain graphs: SVM trained on node strength (SVM-NS) [20], SVM trained on vectorised adjacency matrix (SVM-AM)

[21], and CNN trained on image of adjacency matrix (CNN) [22, 31].

1. Support vector machine baseline models

The SVM-NS and SVM-AM are both trained using an SVM classifier. SVM has only one hyper-parameter, namely the cost, as shown in Table I. Additionally, in order to select an appropriate kernel for SVM, we include two kinds of kernels as hyper-parameters: radial and polynomial (up to 3rd order). Both of the SVM-based baseline models are trained on manually extracted features. All features are first normalised to zero mean and unit standard deviation.

The SVM-NS is trained on node strengths (Fig. 1B). Node strength is defined as the sum of edge weights of one node and can be interpreted as a measure of node importance. Thus, each brain graph is represented by an 23-dimensional feature vector $NS = (ns_1, ns_2, ..., ns_N)$, where N is the number of nodes ($N = 23$).

The SVM-AM is trained on vectorised weighted adjacency matrices (Fig. 1D). As we use only undirected FC measures, the $N \times N$ adjacency matrix of a brain graph is symmetric. Thus, we can use the upper triangular matrix only and flatten it to form a 253-dimensional feature vector $(N \times (N-1)/2)$. Principal component analysis (PCA) is optionally employed for dimensionality reduction with the number of components selected such that 95% of the variance is captured. The inclusion of the PCA step is controlled by a hyper-parameter.

2. Convolutional neural network

CNN classifiers are trained on the weighted adjacency matrices of the brain graphs. As the adjacency matrix is a square matrix, it is simple to convert it to an image on which a CNN can be trained.

The CNN architecture consists of two convolutional blocks and a final classifier, as shown in Fig. 1C. Each convolution block contains two convolutional layers (stride = 3), followed by a maximum pooling layer and a dropout layer. The final classifier consists of two linear layers with a dropout between them. We created several hyper-parameters to control the CNN. The number of convolutional filters within each block is controlled by the Conv1 and Conv2 hyper-parameters. Similarly, the hidden size of the linear layers is controlled by the Linear-hidden hyper-parameter. Additionally, there are hyper-parameters controlling the dropout probability, the choice of the activation function (ReLU or Tanh), and the batch size as shown in Table I.

D. Model evaluation and implementation

The EEG preprocessing, brain graph construction and model evaluation are implemented in R 4.1.2 [32] using in-house scripts, and caret [33] for SVM training. The training of CNN and GNN classifiers is implemented using PyTorch 1.10 [34] and PyTorch Geometric 2.0.2 [35].

The models are trained and evaluated based on repeated 20-fold cross-validation (CV). A 5 times repeated CV is used in order to identify the best combination of hyper-parameters for all models and FC measures. The folds used for CV are created such that samples from the same subject are kept within a single fold in order to prevent information leakage. We use a smaller number of repetitions in order to reduce the computational cost of training CNN and GNN models. Hyper-parameter values are selected using random optimisation, where the values of all hyper-parameters are selected randomly. 200 iterations of random optimisation are performed for each combination of FC measure and model type. The hyper-parameters of all three model types and their possible values are summarised in Table I.

The best-performing models are selected using the area under the sensitivity-specificity curve (AUC), i.e. one model per each combination of FC measure and model type. In order to assess the stability of the selected models, 50 times repeated CV is performed. The performance errors are computed using the maximum difference between the mean and 5th and 95th quantiles. This approach does not assume a normal distribution and results in conservative error estimates.

The CNN and GNN models are trained using an Adam optimiser with an exponential learning rate decay (controlled by the gamma hyper-parameter) and cross-entropy loss function. The models are trained for 300 epochs with an early stopping after 15 epochs if the loss stops decreasing.

IV. RESULTS AND DISCUSSION

Brain graphs were inferred for each 3-second-long EEG segment by using several commonly used FC measures, which aim to quantify the both linear and nonlinear coupling between pairs of brain signals. The brain graphs were then used as an input to train the GNN brain-graph-classifier. Moreover, three baseline models were trained on these brain graphs in order to demonstrate which type of classifier performs the best. AUC is used to select the best model.

Table II reports the AUC values and the 95% confidence intervals of the SVM-NS, SMV-AM and CNN baseline models and GNN across the 8 FC measures. Additionally, the performance of the baseline GNN using Euclidean distance between spatial positions of EEG (GNN-euclid) is reported in Table II as well. The hyper-parameter values of the best models from their respective categories are reported in Table IV. The averaged

sensitivity-specificity curves of these models are shown in Fig. 2.

First, we compared the performances of the baseline models with the highest AUC in their respective categories in order to select the best baseline model. These are the models using MI, PLV and MI for SVM-NS, SVM-AM and CNN, respectively. As the AUC samples collected by 50-times repeated-CV are not normally distributed, non-parametric tests were used, and Bonferroni p-value correction is used to adjust for multiple comparisons. Kruskal-Wallis test shows that there are significant differences in the AUC of the three baseline models ($H = 115.06(df = 2), p - value < 0.001$). Pairwise Mann-Whitney U test is used for posthoc testing. CNN performs significantly better than SVM-NS ($U = 2414, p - value < 0.001$) and SVM-AM ($U = 2475, p - value < 0.001$). SVM-NS performs significantly better than SVM-AM ($U = 208.5, p - value < 0.001$). Thus, the CNN using MI to infer the brain graphs is selected as the best performing baseline model. However, the GNN-euclid baseline model performs significantly better than the CNN model ($U = 6, p - value < 0.001$). These results demonstrate the superior ability of GNN to learn on graph-structured objects despite the fact that the GNN-euclid utilises a fixed graph-structure. In other words, the GNN-euclid cannot learn from the information embedded in the graph structure as the graph structure is identical for all brain graphs.

Using FC measures to infer the brain graphs seems to improve the GNN performance compared to the fixed graph structure used in GNN-euclid as shown in Table II. Mann-Whitney U tests are used to compare the FC-based GNNs to GNN-euclid. All of the FC-based GNNs perform significantly better compared to GNN-euclid: AEC ($U = 204.5, p - value < 0.001$), coh ($U = 446, p - value < 0.001$), corr ($U = 610.5, p - value < 0.001$), iCOH ($U = 333, p - value < 0.001$), MI ($U = 782.5, p - value = 0.01$), PLI ($U = 682, p - value < 0.001$), PLV ($U = 610, p - value < 0.001$), and wPLI ($U = 370, p - value < 0.001$).

This suggests that it might be preferable to utilise FC-based brain graphs rather than the distance-based fixed graphs previously used for EEG-GNN tasks [23, 24]. However, it seems that no FC measure offers clearly superior performance compared to the others. Thus, we suggest that future studies need to carefully consider which FC measure to use based on the type of brain coupling they might wish to focus on. However, we do not claim that the brain graphs inferred by various FC measures are necessarily similar from a graph-theoretic perspective. This is supported by the performance differences of the baseline models where some of the FC measures, such as MI, perform consistently well.

Using edge-filtering methods also seem to improve the GNN performance as the best performing model utilises the brain graphs filtered to only top-20% of the strongest edges in Table IV. However, since we framed the edge-filtering method as a hyper-parameter, it is not possible to perform a fair comparison to elucidate the true effect of using edge-filtering methods to produce sparser brain graphs. Similarly, there seems to be an advantage to using frequency-band-centred brain graphs instead of the entire frequency range. This is especially clear from the baseline models where the δ frequency band was selected for both SVM-NS and SVM-AM IV. CNN also performed best using one of the low-frequency bands, the θ frequency band specifically.

The use of edge-filtering methods instead of a more traditional surrogate-threshold-based edge-filtering might explain the large difference in performances of GNN and the remaining baseline models. Both GNN and CNN contain dropout layers which force the models to learn robust features insensitive to noise. Therefore, the performance difference is smaller compared to the SVM-based baseline models. In contrast, the SVM baseline models were trained on manually extracted features based on the likely-noise-contaminated brain graphs. The SVM algorithm does not contain a mechanism analogous to the dropout of GNN and CNN and thus relies on noisy features.

Finally, the GNN architecture utilised in this study is relatively simple as one of the simplest GCNs was used, and the readout layer merely computes the maximum of the node embeddings. Previous EEG-GNN applications demonstrated the advantages of using more complex graph convolutional layers and edge pooling mechanisms [23]. We hypothesise that exploiting learnable edge-filtering mechanisms akin to those utilised by Zhong et al. [26] instead of relying on the naive edge-filtering methods used in this study might improve the classification.

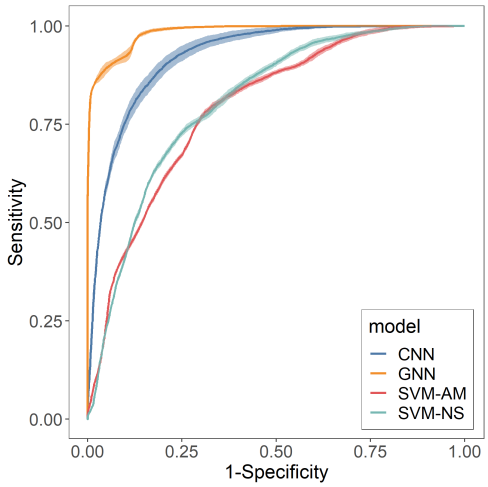

FIG. 2: Averaged Sensitivity-Specificity curves of the best models of their respective categories with 95% confidence intervals (ribbon).

FC	GNN	SVM-NS	SVM-AM	CNN
AEC	**0.984 ± 0.002**	0.75 ± 0.022	0.734 ± 0.01	0.886 ± 0.019
coh	0.982 ± 0.002	0.773 ± 0.007	0.784 ± 0.008	0.901 ± 0.014
corr	0.982 ± 0.003	0.764 ± 0.008	0.773 ± 0.008	0.887 ± 0.014
euclid	0.978 ± 0.004	✗	✗	✗
iCOH	**0.984 ± 0.002**	0.656 ± 0.017	0.648 ± 0.014	0.876 ± 0.01
MI	0.981 ± 0.002	**0.807 ± 0.008**	0.779 ± 0.013	**0.924 ± 0.015**
PLI	0.982 ± 0.005	0.761 ± 0.057	0.603 ± 0.036	0.893 ± 0.015
PLV	0.982 ± 0.003	0.766 ± 0.008	**0.793 ± 0.007**	0.9 ± 0.014
wPLI	**0.984 ± 0.003**	0.66 ± 0.069	0.637 ± 0.024	0.869 ± 0.016

TABLE II: AUC of GNN, SVM-NS, SVM-AM and CNN models across different FC measures measured by 50-repeated 20-fold cross-validation. The 'euclid' entry refers to the baseline GNN model with fixed graph structure based on the spatial distance of EEG electrodes.

	AUC	Accuracy	Sensitivity	Specificity
GNN (AEC)	0.984 ± 0.002	0.92 ± 0.004	0.974 ± 0.009	0.867 ± 0.01
CNN (MI)	0.924 ± 0.015	0.847 ± 0.022	0.862 ± 0.036	0.832 ± 0.046
SVM±NS (MI)	0.807 ± 0.008	0.739 ± 0.009	0.735 ± 0.009	0.744 ± 0.009
SVM±AM (PLV)	0.793 ± 0.007	0.729 ± 0.006	0.721 ± 0.007	0.738 ± 0.011

TABLE III: Detailed performance metrics of best performing models (selected based on AUC) of each model type.

Hyper-parameter	GNN	SVM-NS	SVM-AM	CNN
FC measure	AEC	MI	PLV	MI
Cost		0.5	0.5	
Batch size	32			64
Learning rate	1E-3			1E-3
GCN-hidden	1024			
N-GCN	2			
Conv1-hidden				128
Conv2-hidden				64
Linear-hidden				96
Activation/Kernel		Radial	Polynomial (1st)	ReLU
DropEdge	TRUE			
Drop-rate	0.9			0.1
Frequency band	alpha	theta	theta	theta
Graph filter	Top-20%	MST-3	full	full
Gamma	0.9			0.9

TABLE IV: Hyper-parameter values of the best performing GNN, SVM-NS, SVM-AM and CNN measured by AUC.

V. CONCLUSION

GNN is an effective model for learning on graph structured data such as FC-EEG brain graphs. However, in the absence of consent about the ideal FC measure for estimating EEG brain graphs, the effect of an FC measure on the performance of GNN classifiers is unclear. In this paper, we have selected eight common FC measures to investigate this effect.

First, we demonstrated that GNN models are superior to classical machine learning and CNN models for brain graph classification. Unfortunately, the utilised GNN architecture is a black-box model. Thus, future work should focus on implementing interpretable GNN architectures that achieve similar performance but additionally offer interpretability, such as which nodes, i.e. brain regions, drive the prediction. Besides providing an opportunity for experts to validate such models, interpretable predictions might also serve in the development of GNN-informed targeted treatment.

Finally, we showed that utilising FC measures to define the brain graph results in improved performance of GNN models compared to a fixed graph structure (i.e. the Euclidean distance between EEG electrodes). While using an FC measure improves the performance, no concrete FC measure can be recommended as the ideal choice. Thus, in future research, the choice of suitable FC measure should be carefully evaluated in the context of the given research question. Alternatively, focusing on fusion methods might lead to developing a novel composite measure of FC.

ACKNOWLEDGEMENT

The EEG data used in this study was approved by the Yorkshire and The Humber (Leeds West) Research Ethics Committee (reference number 14/YH/1070). All participants gave their informed written consent.

[1] Clive Ballard, Serge Gauthier, Anne Corbett, Carol Brayne, Dag Aarsland, and Emma Jones. Alzheimer's disease. *The Lancet*, 377(9770):1019–1031, 2011.

[2] Xavier Delbeuck, Martial Van der Linden, and Fabienne Collette. Alzheimer'disease as a disconnection syndrome? *Neuropsychology review*, 13(2):79–92, 2003.

[3] Willem de Haan, Wiesje M van der Flier, T Koene, Lieke L Smits, Philip Scheltens, and Cornelis J Stam. Disrupted modular brain dynamics reflect cognitive dysfunction in Alzheimer's disease. *Neuroimage*, 59(4):3085–3093, 2012.

[4] Matthew R Brier, Jewell B Thomas, and Beau M Ances. Network dysfunction in Alzheimer's disease: refining the disconnection hypothesis. *Brain connectivity*, 4(5):299–311, 2014.

[5] Daniel J. Blackburn, Yifan Zhao, Matteo De Marco, Simon M. Bell, Fei He, Hua-Liang Wei, Sarah Lawrence, Zoe C. Unwin, Michelle Blyth, and Jenna Angel. 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*, 8(7):134, 2018.

[6] Dominik Klepl, Fei He, Min Wu, Matteo De Marco, Daniel Blackburn, and Ptolemaios Georgios Sarrigiannis. Characterising Alzheimer's disease with EEG-based energy landscape analysis. *IEEE Journal of Biomedical and Health Informatics*, 2021.

[7] Raymundo Cassani, Mar Estarellas, Rodrigo San-Martin, Francisco J. Fraga, and Tiago H. Falk. Systematic review on resting-state EEG for Alzheimer's disease diagnosis and progression assessment. *Disease markers*, 2018, 2018.

[8] Justin Dauwels, François Vialatte, Toshimitsu Musha, and Andrzej Cichocki. A comparative study of synchrony measures for the early diagnosis of Alzheimer's disease based on EEG. *NeuroImage*, 49(1):668–693, 2010.

[9] O. Vyšata, M. Vališ, A. Procházka, R. Rusina, and L. Pazdera. Linear and nonlinear EEG synchronization in Alzheimer's disease. *Neurophysiology*, 47(1):46–52, 2015.

[10] Claudio Babiloni, Roberta Lizio, Nicola Marzano, Paolo Capotosto, Andrea Soricelli, Antonio Ivano Triggiani, Susanna Cordone, Loreto Gesualdo, and Claudio Del Percio. Brain neural synchronization and functional coupling in Alzheimer's disease as revealed by resting state EEG rhythms. *International Journal of Psychophysiology*, 103:88–102, 2016.

[11] Kaustubh Supekar, Vinod Menon, Daniel Rubin, Mark Musen, and Michael D Greicius. Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. *PLoS Comput Biol*, 4(6):e1000100, 2008.

[12] A Kabbara, H Eid, W El Falou, M Khalil, F Wendling, and M Hassan. Reduced integration and improved segregation of functional brain networks in Alzheimer's disease. *Journal of neural engineering*, 15(2):026023, 2018.

[13] Ruofan Wang, Jiang Wang, Haitao Yu, Xile Wei, Chen Yang, and Bin Deng. Decreased coherence and functional connectivity of electroencephalograph in Alzheimer's disease. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 24(3):033136, 2014.

[14] Fabrizio Vecchio, Francesca Miraglia, Francesca Alú, Alessandro Orticoni, Elda Judica, Maria Cotelli, and Paolo Maria Rossini. Contribution of graph theory applied to EEG data analysis for Alzheimer's disease versus vascular dementia diagnosis. *Journal of Alzheimer's Disease*, 82(2):871–879, 2021.

[15] Casper T Briels, Deborah N Schoonhoven, Cornelis J Stam, Hanneke de Waal, Philip Scheltens, and Alida A Gouw. Reproducibility of EEG functional connectivity in Alzheimer's disease. *Alzheimer's research & therapy*, 12(1):1–14, 2020.

[16] Saúl J Ruiz-Gómez, Roberto Hornero, Jesús Poza, Aarón Maturana-Candelas, Nádia Pinto, and Carlos Gómez. Computational modeling of the effects of EEG volume conduction on functional connectivity metrics. application to Alzheimer's disease continuum. *Journal of neural engineering*, 16(6):066019, 2019.

[17] Cornelis J Stam, BF Jones, G Nolte, M Breakspear, and Ph Scheltens. Small-world networks and functional connectivity in Alzheimer's disease. *Cerebral cortex*, 17(1):92–99, 2007.

[18] Guido Nolte, Ou Bai, Lewis Wheaton, Zoltan Mari, Sherry Vorbach, and Mark Hallett. Identifying true brain interaction from EEG data using the imaginary part of coherency. *Clinical neurophysiology*, 115(10):2292–2307, 2004.

[19] Zhongliang Yin, Jun Li, Yun Zhang, Aifeng Ren, Karen M Von Meneen, and Liyu Huang. Functional brain network analysis of schizophrenic patients with positive and negative syndrome based on mutual information of EEG time series. *Biomedical Signal Processing and Control*, 31:331–338, 2017.

[20] Sou Nobukawa, Teruya Yamanishi, Shinya Kasakawa, Haruhiko Nishimura, Mitsuru Kikuchi, and Tetsuya Takahashi. Classification methods based on complexity and synchronization of electroencephalography signals in Alzheimer's disease. *Frontiers in psychiatry*, 11:255, 2020.

[21] Christian Sandøe Musaeus, Knut Engedal, Peter Høgh, Vesna Jelic, Morten Mørup, Mala Naik, Anne-Rita Oeksengaard, Jon Snaedal, Lars-Olof Wahlund, Gunhild Waldemar, et al. Oscillatory connectivity as a diagnostic marker of dementia due to Alzheimer's disease. *Clinical Neurophysiology*, 130(10):1889–1899, 2019.

[22] Francisco Aparecido Rodrigues, Caroline Alves, Aruane Pineda, Kirstin Roster, and Cristiane Thielemann. Eeg functional connectivity and deep learning for automatic diagnosis of brain disorders: Alzheimer's disease and schizophrenia. *Journal of Physics: Complexity*, 2022.

[23] Andac Demir, Toshiaki Koike-Akino, Ye Wang, Masaki Haruna, and Deniz Erdogmus. EEG-GNN: Graph neural networks for classification of electroencephalogram (EEG) signals. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1061–1067. IEEE, 2021.

[24] Neeraj Wagh and Yogatheesan Varatharajah. EEG-GCNN: Augmenting electroencephalogram-based neurological disease diagnosis using a domain-guided graph convolutional neural network. In *Machine Learning for Health*, pages 367–378. PMLR, 2020.

[25] Hanjie Liu, Jinren Zhang, Qingshan Liu, and Jinde Cao. Minimum spanning tree based graph neural network for emotion classification using EEG. *Neural Networks*, 145:308–318, 2022.

[26] Peixiang Zhong, Di Wang, and Chunyan Miao. EEG-based emotion recognition using regularized graph neural networks. *IEEE Transactions on Affective Computing*, 2020.

[27] Cornelis J Stam, Guido Nolte, and Andreas Daffertshofer. Phase lag index: assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources. *Human brain mapping*, 28(11):1178–1193, 2007.

[28] Martin Vinck, Robert Oostenveld, Marijn Van Wingerden, Franscesco Battaglia, and Cyriel MA Pennartz. An improved index of phase-synchronization for electrophysiological data in the presence of volume-conduction, noise and sample-size bias. *Neuroimage*, 55(4):1548–1565, 2011.

[29] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

[30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[31] Xiaowei Li, Rong La, Ying Wang, Bin Hu, and Xuemin Zhang. A deep learning approach for mild depression recognition based on functional connectivity using electroencephalography. *Frontiers in neuroscience*, 14:192, 2020.

[32] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[33] Max Kuhn. *caret: Classification and Regression Training*, 2021. R package version 6.0-90.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[35] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.