BrainSTEAM: A Practical Pipeline for Connectome-based fMRI Analysis towards Subject Classification

Alexis Li

Hamilton High School,
E-mail: li.alexis1111@gmail.com

Yi Yang

Duke University,
E-mail: owen.yang@duke.edu

Hejie Cui

Department of Computer Science, Emory University,
E-mail: hejie.cui@emory.edu

Carl Yang

Department of Computer Science, Emory University,
E-mail: j.carlyang@emory.edu

Functional brain networks represent dynamic and complex interactions among anatomical regions of interest (ROIs), providing crucial clinical insights for neural pattern discovery and disorder diagnosis. In recent years, graph neural networks (GNNs) have proven immense success and effectiveness in analyzing structured network data. However, due to the high complexity of data acquisition, resulting in limited training resources of neuroimaging data, GNNs, like all deep learning models, suffer from overfitting. Moreover, their capability to capture useful neural patterns for downstream prediction is also adversely affected. To address such challenge, this study proposes BrainSTEAM, an integrated framework featuring a spatio-temporal module that consists of an EdgeConv GNN model, an autoencoder network, and a Mixup strategy. In particular, the spatio-temporal module aims to dynamically segment the time series signals of the ROI features for each subject into chunked sequences. We leverage each sequence to construct correlation networks, thereby increasing the training data. Additionally, we employ the EdgeConv GNN to capture ROI connectivity structures, an autoencoder for data denoising, and mixup for enhancing model training through linear data augmentation. We evaluate our framework on two real-world neuroimaging datasets, ABIDE for Autism prediction and HCP for gender prediction. Extensive experiments demonstrate the superiority and robustness of BrainSTEAM when compared to a variety of existing models, showcasing the strong potential of our proposed mechanisms in generalizing to other studies for connectome-based fMRI analysis.

Keywords: Brain Connectome Analysis; Neuroimaging Studies; Synthetic Data Generation

1. Introduction

Functional brain networks illustrate the dynamic connectivity patterns between anatomical regions of interest (ROIs) for different cognitive states and different responses to disease or injury.[1] The study of functional brain networks provides insights into the underlying mechanisms of human consciousness, developmental processes, and the neural bases of various neurological and psychiatric disorders such as autism, ADHD, depression, and schizophrenia.[2] However, existing computational tools often extract a single static graph structure based on correlations among full BOLD signals, which ignores the dynamic changes of functional connectivity.[3–5]

Compared with other deep learning paradigms such as Convolutional Neural Networks (CNNs),[6] and Recurrent Neural Networks (RNNs),[7] Graph Neural Networks (GNNs)[8,9] provide unique benefits in functional brain network analysis due to its capability in modeling connectivity structures.[10–17] However, most GNN-based frameworks resort to static correlation networks as data instances, and they are prone to unstable performances due to large data noises in the BOLD signals and overfitting due to limited data labels of clinical outcomes. This is especially true for the ABIDE dataset as the images come from 17 international sites with differing imaging protocol, as well as heterogeneity within the dataset.[18]

To address the challenges above, this study proposes BrainSTEAM, an integrated pipeline that features a spatio-temporal module, for brain connectome analysis on dynamic fMRI networks. Specifically, we propose a temporal chunking approach to dynamically segment the BOLD signals of each subject into partitioned sequences based on a tunable sliding window to capture the local connectivity structures at different scales, which are further modeled by EdgeConv. An autoencoder is devised to discover the important connectivity patterns during ROI pooling through learnable dropout, where the objective is to reconstruct the full connectivity patterns only based on the important ones. Mixup is applied to further stabilize and enhance training of the whole framework through linear data augmentation to prevent the model from memorizing certain data points.

Extensive experiments conducted in this study demonstrate that our BrainSTEAM model outperforms state-of-the-art models on both mental disorder prediction and gender classification, indicating its effectiveness in modeling functional brain networks and also highlighting its flexibility and versatility. It is also promising to apply BrainSTEAM to the analysis of functional brain networks for other clinical applications, as well as other dynamic graphs extracted from time-series data. For clinical applications in particular, this model would help to address the limitations of MRI data collection as there are limited scans due to the expensive nature of MRIs and constant exposure for the patient. Decreasing information loss can make the model more robust, providing a more reliable aid for those in a clinical setting.

2. Related Work

2.1. *Data Augmentation*

Mixup utilizes the principles of vicinal risk minimization across different classes, constructing new data as a combination of existing data points.[19] Graph Mixup techniques often involve creating synthetic graphs samples connected subgraphs or reorder the original graph structure.

Previous works such as G-Mixup use probability matrices to predict if an edge exists between two nodes, and Graph Transplant samples the top nodes in a graph and then appends a partial K-hope subgraph to predict edges.[20,21] However, sampling subgraphs and appending them to the original graph becomes problematic when considering the fixed nature of brain ROIs.

Additionally, previous studies have used temporal based augmentation techniques to improve model generalization. STDAC proposed a module using random discontinuous sampling period with a tensor fusion method to combine it with the spatial model.[22] Multi-Head GAGNN modeled both spatio and patterns of functional brain networks simultaneously to fully utilize their characteristics.[23] These methods are still often limited by small sizes, thus there lies potential in combining a spatio-temporal data augmentation technique with mixup.

2.2. *Graph Pooling*

Previous Graph Pooling methods use hierarchical graph clustering methods, following the principle of local neighborhoods with nodes.[24] This has extended to deterministic clustering algorithms and attention based mechanisms to increase the quality of assigning the clusters.[25,26] Other methods include node drop pooling to decrease the time and space required for the process by simply selecting a subset of nodes to construct the coarsened graph. Traditional pooling methods include selecting the top-k nodes, using self-attention networks, and a gated structured aware approach.[27–29] Yet, these methods are also limited by small sample sizes and are prone to focusing on local structures rather than the graph as a whole.

3. The Proposed Model

3.1. *Capturing Dynamic Connectivity via Temporal Chunking and EdgeConv Analysis*

We define a directed graph as $G = \{V, E\}$ for each brain network subject, where V is the set of nodes with a time series and $\boldsymbol{E}$ represents the weighted connectivity. Temporal Chunking is defined by looking at a smaller window of time in the subject's time series data at any one point in time rather than aggregating it as a whole. The window sizes vary from 128 to 50 to 64 and the starting points of the window are randomly generated for each epoch. For each generated window, the partial correlation matrices are extracted to form the adjacency matrix. This dramatically increases the variety of the data the model has to work with, allowing for more robust model at the end of training. It helps to combat the issue of overfititng that previous models have cited as limitations. The model is relevant for clinical use as it can better adapt to the small sample sizes that are commonly seen in MRI datasets and can better adapt to new patients as well. Its novelty lies in its integration of BrainGMixup which ensures maximal data variation by accounting for both spatial and temporal based data augmentation.

Edge features are defined as $\boldsymbol{e}_{ij} = h_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with $h_\theta = \mathbb{R}^F \times \mathbb{R}^F \rightarrow \mathbb{R}^{F'}$ as the MLP for the model with a nonlinear function parameterized by a set of learnable parameters. $\boldsymbol{x}_i$ represents the embedding of node i and $\boldsymbol{x}_j$ represents the embeddings of all the neighbors of node i, including the node itself. In this case, a sum aggregation operation is performed over all the edge features to get the final embedding for the node' and its neighbor's edges represented by

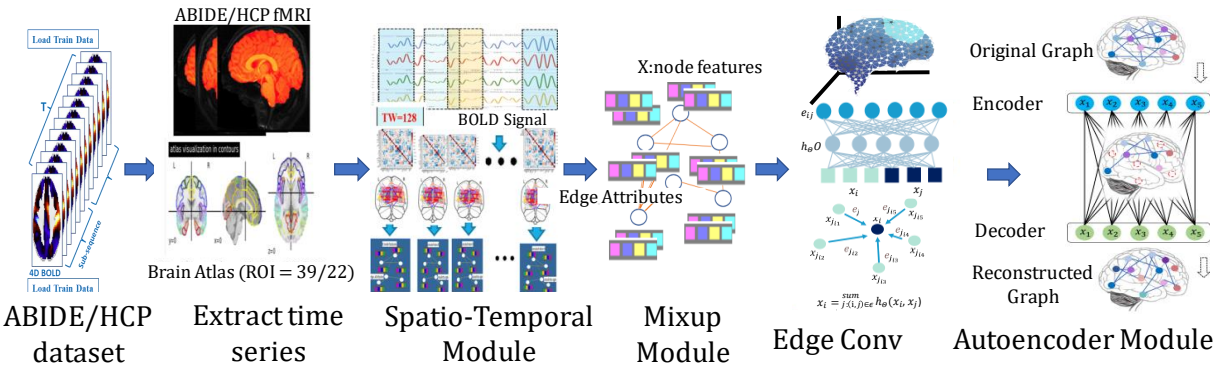

Fig. 1: Overview of the proposed BrainSTEAM architecture.

the following equation:

$$\boldsymbol{x}_i' : \underset{j_i(i,j)\in E}{sum} h_\theta(\boldsymbol{x}_i || \boldsymbol{x}_j - \boldsymbol{x}_i) \tag{1}$$

EdgeConv[30] allows for the extraction of neighborhood-level features within the overall topological structure of the network. Different aggregation methods can be used across the embeddings of the node and the neighbors. By determining the pairwise distance matrices for the characteristics and selecting the k nearest neighbors for each point, the graph is also dynamically updated, where k is a hyper-parameter that can be varied to obtain desirable results.

3.2. *Discovering Important Connectivity via Autoencoder-based Pooling*

Graph pooling is a key component to compress the predictions of multiple nodes into a graph-level classification. To discover important connectivity, we adapted Graph Autoencoder[31] technique where the node dropping is performed to measure the importance of the node for reconstructing the topological structure without labels. The new graph generated by the pooled graph can be defined as:

$$G' = \mathrm{POOL}(G), \tag{2}$$

where the pooling method SAGPool[28] acts as the encoder of the autoencoder. The SAGPool first generates scores for all the nodes from convolution and performs pooling by only taking the top k scoring nodes, with the pooling ratio determined by a hyperparameter k. Those nodes are then used to compose a new coarsened graph by learning the attribute and adjacency matrices:

$$\begin{aligned} \boldsymbol{Z}^{(l+1)} &= \boldsymbol{Z}^{(l)}_{idx^{(l)}} \odot \boldsymbol{S}^{(l)}_{idx^{(l)}} \in \mathbb{R}^{n^{(l+1)}\times 1}, \\ \boldsymbol{A}^{(l+1)} &= \boldsymbol{A}^{(l)}_{(idx^{(l)},idx^{(l)})} \in \{0,1\}^{n^{(l+1)}\times n^{(l+1)}}, \end{aligned} \tag{3}$$

where idx serves at the indexing operator for the top-k significant scoring nodes, $\boldsymbol{Z}^{(l)}_{idx^{(l)}}$ is the row wise indexed embedding matrix, and $\odot$ is the broadcast elementwise product. $\boldsymbol{A}$ is displayed as the row-wise and column-wise adjacency matrix. $\boldsymbol{Z}^{l+1}$ and $\boldsymbol{A}^{(l)}_{(idx^{(l)},idx^{(l)})}$ are respectively the new attribute and adjacency matrices. $\boldsymbol{S}^{(l)}_{idx^{(l)}}$ represents the score matrix of

the top k selected nodes at layer l. The score matrix was calculated by inputting the adjacency matrix and node embedding matrix at the layer l into a Graph Convolution Network (GCN).

The decoder reconstructs the embeddings of the dropped nodes, which includes the creation of an empty attribute matrix with the pooled node embeddings to reconstruct a new embedding matrix, with zero padding operations performed. To measure the validity of this reconstructed matrix, the Euclidean distance is calculated between the reconstructed attribute matrix and the original input matrix. This becomes a loss function L_f. The Euclidean distance is shown below:

$$L_f^{(l)} = \left\| \boldsymbol{X} - \psi_a(\hat{x}^{(l)}) \right\|_F^2, \tag{4}$$

$$L_d^{(l)} = \left\| \boldsymbol{D}^{(l)} - \psi_d(Z^{(l)}) \right\|_F^2, \tag{5}$$

where $L_f^{(l)}$ represents the loss of the node attributes for the l^{th} layer, $\|\|_F$ is the Frobenius norm, and $\boldsymbol{X}$ represents the node feature matrix. An additional L_d is adopted to regularize the distance between the true degree values and the reconstructed ones. This determines how close the pooling mechanism reconstruction came to the original graph of the subject. $\psi_a(\hat{x}^{(l)})$ represents the the reconstructed node attribute matrix. This method of pooling is preferrable to the typical mean, max, or summation pooling as it identifies the most structurally important nodes and reduces the number of noisy nodes allowing for more focused analysis.

3.3. *Enhancing Model Training and Stability via Mixup*

It is difficult for GNNs to properly analyze the underlying signals in functional brain images with the overfitting and memorization of noise in specific training data.[32] Vicinal risk minimization[33] rather than empirical risk minimization[19] techniques have been applied to improve generalization capability. Vicinal risk minimization referring to creating virtual examples of training data based on their neighborhood of data.

This paper proposes BrainGMixup[34] which utilizes 2D feature vectors from the node and edge features rather than the 1D feature vectors for other forms of data such as CNN networks. This requires interpolation between the rows/ROIs of the graph rather than between individual feature columns. This differs from traditional Graph Mixup approaches as it involves interpolation instead of concatenation of smaller sub graphs. Mixup intends to take two subjects and combine their feature and edge index information to create a new node for the model to train on. The fixed nature of ROIs in the data allows for the mixup to be applied across rows for the node feature matrix and edge index matrix,

$$\begin{aligned} \tilde{\boldsymbol{X}} &= \lambda \boldsymbol{X}_i + (1-\lambda)\boldsymbol{X}_j, \ \textit{where } i,j = 1,...,N, i \neq j, \\ \tilde{\boldsymbol{E}} &= \lambda \boldsymbol{E}_i + (1-\lambda)\boldsymbol{E}_j, \ \textit{where } i,j = 1,...,N, i \neq j, \end{aligned} \tag{6}$$

$$\tilde{y} = \lambda y_i + (1-\lambda) y_j. \tag{7}$$

N represents the number of ROIs defined in the node feature matrix, $\boldsymbol{E}$ is the edge index matrix, and y is the corresponding label. $\tilde{\boldsymbol{X}}$, $\tilde{y}$, and $\tilde{\boldsymbol{E}}$ are the mixup-augmented samples of

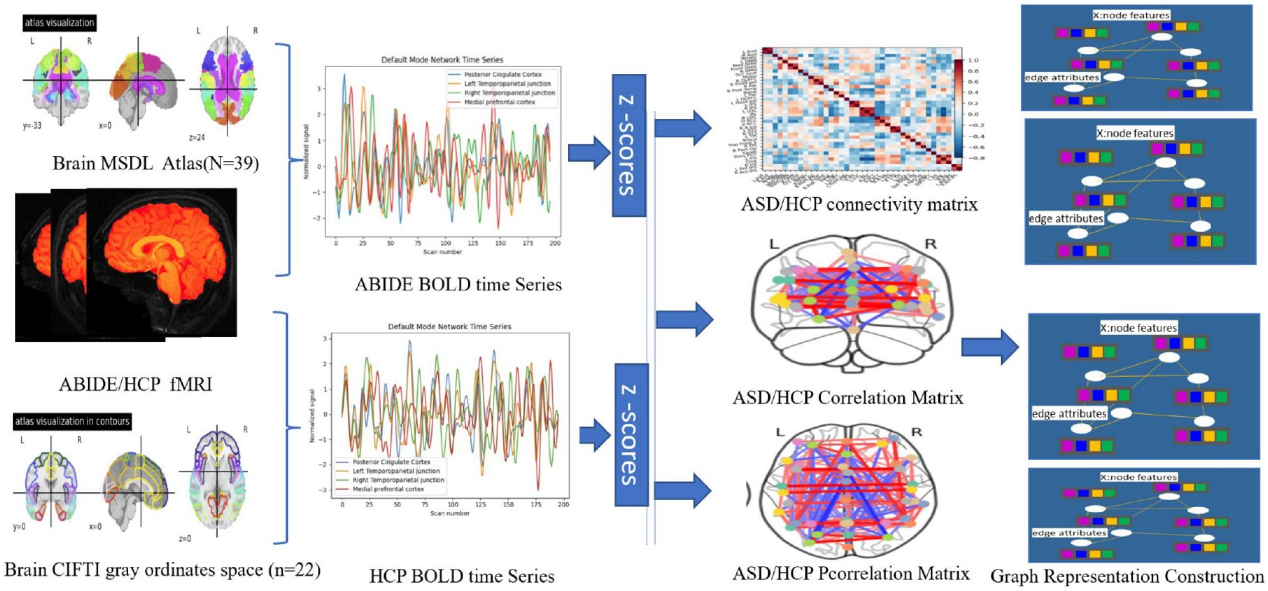

Fig. 2: The process of brain network construction.

the corresponding matrices. It involves the interpolation of the previous graph samples to cover for in-between brain network variations

$$l = \lambda \cdot c(p, y_a) + (1 - \lambda) \cdot c(p, y_b). \tag{8}$$

The mixup loss criterion L_m utilizes the Vicinity distribution[33] to find the chance that a particular feature target is in the near area of that graph to generate L_m. The differentianting lambda values ensures that even if data is taken at a similar timepoint, the resulting graph will not be the same. The hyperparameter a is used to determine the degree of interpolation between the different ROI regions and edge connectives. This serves as an efficient and effective way of accounting for the heterogeneous, scarce, and noisy nature of brain networks. The total overall loss is $L_{all} = \alpha * L_f + \beta * L_d + L_m$. With, *alpha* and *beta* serving as hyperparameters to determine the weight of the feature loss and degree loss. Mixup serves as an additional sample size increase alongside temporal chunking to provide the model with more training modules.

4. Experiments

Datasets. We evaluate our framework using two publicly available real-world neuroimaging datasets, the Autism Brain Imaging Data Exchange (ABIDE)[35]on the ASD prediction task and the Human Connectome Project (HCP)[36] on the gender classification task. The CPAC[37] preprocessed ABIDE dataset is a collection of 4D resting-state functional MRI scans from a total of 1,112 individuals with 539 Autism Spectrum Disorder (ASD) and 573 typical health controls. The preprocessed HCP dataset, on the other hand, is a large-scale dataset that includes resting-state fMRI scans from 1095 subjects with the gender split being 595 females and 500 males with about 1200 frames in each scan.

Brain network construction. The brain Blood Oxygenation Level Dependent (BOLD) signal time series is then extracted from those fMRI subset data with MSDL brain Probabilistic atlas which defines soft parcellations of the brain to 39 ROI on ABIDE and CIFTI(Connectivity Informatics Technology Initiative) (ROI=22) on HCP to produce ABIDE time series matrix (196 $\times$ 39) and HCP time series matrix (1200 $\times$ 22). These were determined from previous experimentation and papers on the appropriate number depending on the condition.[38]

Then, the brain connectivity matrices among ROI are calculated from time series data with partial correlation and correlation matrix, followed by z-scores normalization . Non-zero adjacency matrices mean a pair of ROI nodes share an edge, and the values of adjacency matrices indicate edge weights between nodes. The sparse partial correlation matrix can help to avoid the over-smoothing issue commonly seen in GNN applications. Node features are initialized with the corresponding rows in the edge weight matrix.

The temporal windows/chunks of each subject is constructed using a graph representation object as seen in Figure 2's third step. Each graph representation object then goes through mixup to create a new graph that is a interpolation of two different subjects via the BrainMixup module. This data is then fed into the EdgeConv model to train, and pooling is conducted by the AutoEncoder.

The EdgeConv model contains three block which each block containing a dyanmic EdgeConv layer, a batch normalization layer, and a relu activation layer. Each block also includes the feature decoder and degree decoder layers as a part of the AutoEncoder module. The loss is calculated as seen in the methods section with different weights applied to the loss of the model and Autoencoder loss in regards to the reconstructed feature and degrees in comparison to their ground truth values.

Baselines. We compare our proposed BrainSTEAM with baseline model MAGE,[39] SVM-MTFS,[40] MISO-DNN,[41] e-STAGIN,[41] MAGIN,[42] IMAGIN[42] on the ABIDE dataset, and with ST-GCN,[38] LTSM,[43] GCN,[8] GC-LSTM,[43] STAGIN-SERO[44] and DECENNT[45] on the HCP dataset.

Experimental settings. This study performs training and testing in 5-fold cross-validation, and dynamically construct graph data object for each sub-sequences of different window sizes with fixed optimum W as 128. The learning rate is set as 10^{-4}, epochs as 10000 for ABIDE and 30000 for HCP. All reported results are averaged of five runs of five-fold cross-validation. Additional details regarding the experiment settings can be found in the supplementary materials.

Prediction performance. The overall prediction results presented in Table 2 show that BrainSTEAM outperformed the baseline model MAGE by 9.38%, IMAGIN by 8.25% on the ABIDE dataset, and achieves 7.71% improvements over ST-GCN and 3.21% improvements over STAGIN-SERO on the HCP dataset. The results demonstrate the superiority of BrainSTEAM in neuropsychiatric disorder prediction and gender classification compared to other state-of-the-art models.

	ABIDE					HCP			
	Accuracy	AUC	Precision	Recall		Accuracy	AUC	Precision	Recall
MAGE	75.86	83.14	71.53	79.24	ST-GCN	83.7	-	-	-
SVM+MTFS	$76.7_{\pm 2.7}$	$81_{\pm 0.31}$	$72.5_{\pm 3.2}$	$76.7_{\pm 2.7}$	LTSM	81.7	-	-	-
MISO-DNN	$77.73_{\pm 4.26}$	-	$76.73_{\pm 4.11}$	$77.16_{\pm 3.72}$	GCN	83.98	-	84.59	87.78
e-STAGIN(Sch	$75.81_{\pm 1.70}$	$81.12_{\pm 0.30}$	$78.03_{\pm 2.34}$	$79.06_{\pm 0.89}$	GC-LSTM	81.50	-	-	-
MAGIN	$78.12_{\pm 1.91}$	$85.72_{\pm 0.2}$	$78.37_{\pm 2.11}$	$79.55_{\pm 1.62}$	STAGIN-SERO	$88.20_{\pm 1.33}$	$92.96_{\pm 1.87}$	-	-
IMAGIN	$79.25_{\pm 2.33}$	$86.44_{\pm 0.24}$	$81.03_{\pm 3.47}$	$79.06_{\pm 0.89}$	DECENNT	86.00	93.6	87.2	88.6
BrainSTEAM	$\mathbf{87.5_{\pm 0.99}}$	$89.23_{\pm 0.88}$	$82.24_{\pm 2.48}$	$96.11_{\pm 2.47}$	BrainSTEAM	$\mathbf{91.41_{\pm 0.02}}$	$93.67_{\pm 0.01}$	$100_{\pm 0.00}$	$78.78_{\pm 0.04}$

Table 1: Overall performance (%) comparison on two datasets. Results with - were not provided in the original work.

	ABIDE					HCP			
	Accuracy	AUC	Precision	Recall		Accuracy	AUC	Precision	Recall
BrainSTEAM	$\mathbf{87.5_{\pm 0.99}}$	$89.23_{\pm 0.88}$	$82.24_{\pm 2.48}$	$96.11_{\pm 2.47}$	BrainSTEAM	$\mathbf{91.41_{\pm 0.02}}$	$93.67_{\pm 0.01}$	$100_{\pm 0.00}$	$78.78_{\pm 0.04}$
BrainEAM	$62.86_{\pm 0.87}$	$62.36_{\pm 0.78}$	$67.23_{\pm 0.09}$	$63.95_{\pm 1.70}$	BrainEAM	$77.20_{\pm 1.35}$	$80.15_{\pm 2.27}$	$87.43_{\pm 4.49}$	$66.59_{\pm 5.82}$
BrainEM	$63.66_{\pm 1.45}$	$62.50_{\pm 1.66}$	$68.09_{\pm 2.14}$	$71.24_{\pm 1.69}$	BrainEM	$74.42_{\pm 0.01}$	$74.48_{\pm 0.01}$	$77.11_{\pm 0.01}$	$73.79_{\pm 0.01}$
BrainE	$59.43_{\pm 1.48}$	$59.24_{\pm 1.64}$	$60.22_{\pm 0.61}$	$63.98_{\pm 1.34}$	BrainE	$67.85_{\pm 0.01}$	$68.01_{\pm 0.01}$	$68.97_{\pm 0.01}$	$70.46_{\pm 0.02}$

Table 2: The ablation study with different model variants: BrainSTEAM is the full version with all components, BrainEAM removes the temporal chunking, BrainEM removes both the temporal chunking and autoencoder, and BrainE is only equipped with Edgeconv.

We further investigate the influence of each proposed component by removing each at a time. The results are shown in Table 2. Results show the temporal module contributes to the greatest increase in model prediction accuracy performance, improving about 23.84% on ABIDE, and about 14.21% on HCP. The autoencoder module provides more stability to the network as seen by the decrease in the standard deviation.

Key hyperparameter studies are shown in Fig. 3. (a) shows performance is about 1.87% higher when k=10 than k=15; (b) shows performance is about 3.12% higher when window=128 than window=50; (c) shows performance is 7.03% higher when loss alpha and loss beta is set to 0.3 vs 0.1; (d) and (e) show performance increase dramatically when epoch increases from 1k, 5k to 10k/30k with BrainSTEAM, on the contrary, performance stays flat for BrainEAM when epoch increase accordingly both on ABIDE and HCP.

5. Interpretation Analysis

As summarized above, the proposed BrainSTEAM is shown to significantly outperform baseline models. We claim the fundamental reason is that the other baselines only obtain one graph from the subject full range of time series thus only resulting in 1112 graphs for ABIDE and 1095 graphs for HCP. With our proposed time series temporal chunk combined with the mixup, an exponential increase in the number of new graphs can be generated. Specifically, the model is trained on the same 1000 subjects but the generation of time series chunks with 5-fold cross-validation for 30,000 epochs leads to 150,000 different graphs. Hyperparameter tuning with epochs reveals that the BrainEAM model hits a training accuracy of 99% in 200 epochs,

Fig. 3: The hyperparameter study for BrainSTEAM on the ABIDE dataset.

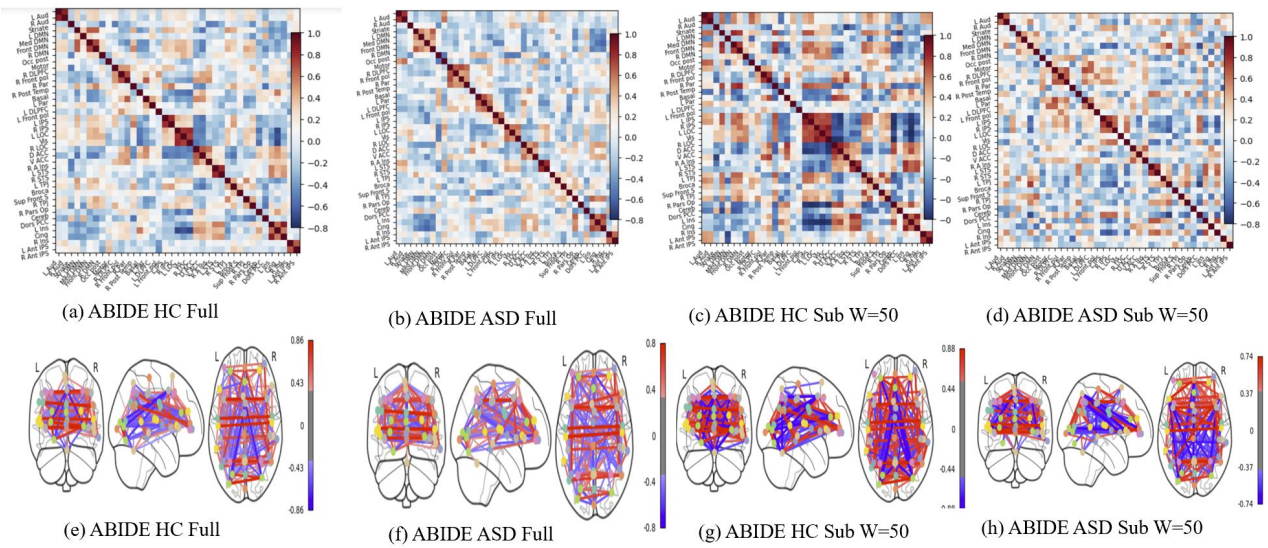

Fig. 4: The visualization of brain connectome, where the subfigure (a) & (e) represent the connectome of ABIDE Health Control (HC) with the full sequence of time series; (b) & (f) represent that of Autism; (c) & (g) represent HC with time series subsequence of window size 50; and (d) & (h) represent Autism with time series subsequence of window size 50.

but the validation accuracy stays in the low 60s indicating a typical sign of overfitting. When applying temporal chunking, the training and validation accuracy scale more evenly with an 18.13% increase of validation accuracy as the number of epochs varies from 1000 to 10000. The temporal chunking results as visualized in Fig. 4 demonstrates that graphs generated at different time windows have significantly different levels of connectivity between ROIs. This

stood true for both health control and patients diagnosed with Autism. The level of interactions for the health control is far more pronounced, as noted with the increase in deep red boxes, than for the Autism patient. This fine-grained interaction difference is not expressed within the graph generated from an average of the entire time series. This demonstrates that the proposed temporal chunking method is able to better capture time specific interactions in the brain and will generate more robust generalization patterns.

The model outperforms ST-GCN, demonstrating that only the temporal module might not be comprehensive enough to cover all the issues that create overfitting and accuracy deficits. A combination of retaining connectivity information and performing self-supervised node dropping is needed to create the most robust version of the model.

6. Conclusion

This study proposes a dynamic functional brain network analysis framework BrainSTEAM, which integrates the temporal sliding window module with EdgeConv, Autoencoder and Mixup for the first time. Extensive experiments on two real-world neuroimaging datasets exhibit significant performance improvement over the state-of-the-art. This study also shows the contribution of each component to the system, demonstrating the temporal chunking approach as the major contributor to performance improvement, which allows for the representation of functional brain connectivity within smaller time windows to capture unique fine-grained ROI interactions. In the meantime, the study also shows EdgeConv helps in capturing the connectivity structures of the brain networks, autoencoder helps in reducing data noise and identifying the most relevant connectivity patterns, and mixup helps in enhancing the model training through linear interpolation. For future work, we look to improve BrainSTEAM with explainability, such as identifying meaningful biomarkers linked to neuropsychiatric disorders and mental development, understanding which neural systems contribute most to the prediction of a specific disease, applying the model to other datasets and tasks, and exploring its potential applications in clinical settings.

References

1. E. Bullmore and O. Sporns, The economy of brain network organization, *Nat. Rev. Neurosci.* (2012).
2. S. L. Bressler and V. Menon, Large-scale brain networks in cognition: emerging methods and principles, *Trends Cogn. Sci.* (2010).
3. S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey and M. W. Woolrich, Network modelling methods for FMRI, *NeuroImage* **54** (2011).
4. S. L. Simpson, F. D. Bowman and P. J. Laurienti, Analyzing complex functional brain networks: fusing statistics and network science to understand the brain, *Stat. Surv.* **7**, p. 1 (2013).
5. X. Kan, Y. Kong, T. Yu and Y. Guo, Bracenet: Graph-embedded neural network for brain network analysis, in *IEEE Big Data*, 2022.
6. J. Kawahara, C. Brown, S. Miller, B. Booth, V. Chau, R. Grunau, J. Zwicker and G. Hamarneh, Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment, *NeuroImage* **146**, 1038 (2017).
7. Y. Cui, S. Zhao, H. Wang, L. Xie, Y. Chen, J. Han, L. Guo, F. Zhou and T. Liu, Identifying

brain networks at multiple time scales via deep recurrent neural network, *IEEE J Biomed Health Inform* **23**, 2515 (2018).
8. T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, *ICLR* (2016).
9. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengios, Graph attention networks, *ICLR* (2018).
10. X. Li, Y. Zhou, N. Dvornek, M. Zhang, S. Gao, J. Zhuang, D. Scheinost, L. Staib, P. Ventola and J. Duncan, Braingnn: Interpretable brain graph neural network for fmri analysis, *Med. Image Anal.* **74** (2021).
11. H. Cui, W. Dai, Y. Zhu, X. Li, L. He and C. Yang, Interpretable graph neural networks for connectome-based brain disorder analysis, in *MICCAI*, 2022.
12. X. Kan, H. Cui, J. Lukemire, Y. Guo and C. Yang, Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation, in *MIDL*, 2022.
13. X. Kan, W. Dai, H. Cui, Z. Zhang, Y. Guo and C. Yang, Brain network transformer, in *NeurIPS*, 2022.
14. H. Cui, W. Dai, Y. Zhu, X. Kan, A. A. C. Gu, J. Lukemire, L. Zhan, L. He, Y. Guo and C. Yang, A benchmark for brain network analysis with graph neural networks, *IEEE TMI* (2022).
15. Y. Zhu, H. Cui, L. He, L. Sun and C. Yang, Joint embedding of structural and functional brain networks with graph neural networks for mental illness diagnosis, in *EMBC*, 2022.
16. Y. Yang, Y. Zhu, H. Cui, X. Kan, L. He, Y. Guo and C. Yang, Data-efficient brain connectome analysis via multi-task meta-learning, in *KDD*, 2022.
17. Y. Yang, H. Cui and C. Yang, Ptgb: Pre-train graph neural networks for brain network analysis, in *CHIL*, 2023.
18. A. Abraham, M. P. Milham, A. D. Martino, R. C. Craddock, D. Samaras, B. Thirion and G. Varoquaux, Deriving reproducible biomarkers from multi-site resting-state data: An autism-based example, *NeuroImage* **147**, 736 (2017).
19. H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, mixup: Beyond empirical risk minimization, *ICLR* (2018).
20. X. Han, Z. Jiang, N. L. Liu and X. Hu, G-mixup: Graph data augmentation for graph classification, *ICML* (2022).
21. J. Park, H. Shim and E. Yang, Graph transplant: node saliency-guided graph mixup with local structure preservation, *Proceedings of the First MiniCon Conference* (2022).
22. Q. Liu, Y. Zhang, L. Guo and Z. Wang, Spatial-temporal data-augmentation-based functional brain network analysis for brain disorders identification, *Frontiers in Neuroscience* (2023).
23. J. Yan, Y. Chen, Z. Xiao, S. Zhang, M. Jiang, T. Wang, T. Zhang, J. Lv, B. Becker, R. Zhang, D. Zhu, J. Han, D. Yao, K. M. Kendrick, T. Liu and X. Jiang, Modeling spatio-temporal patterns of holistic functional brain networks via multi-head guided attention graph neural networks (multi-head gagnns), *Medical Image Analysis* **80** (2022).
24. Z. Zhang, J. Bu, M. Ester, J. Zhang, Z. Li, C. Yao, H. Dai, Z. Yu and C. Wang, Hierarchical multi-view graph pooling with structure learning," in ieee transactions on knowledge and data engineering, *IEEE TKDE* **35**, 545 (2023).
25. A. Duval and F. Malliaros, Higher-order clustering and pooling for graph neural networks, in *CIKM*, 2022.
26. Z. Peng, H. Liu, Y. Jia and J. Hou, Attention-driven graph clustering network, *ACM International Conference on Multimedia* , 935 (2021).
27. H. Gao and S. Ji, Graph u-nets, *IEEE TPAMI* **44**, 4948 (2022).
28. J. K. Junhyun Lee, Inyeop Lee, Self-attention graph pooling, *ICML* (2019).
29. H. Yu, J. Yuan, H. Cheng, M. Cao and C. Wang, Graph u-nets, in *IJCNN*, 2021.
30. Y. Wang, Y. Sun, Z. Liu, S. Sarma, M. Bronstein and J. Solomon, Dynamic graph cnn for

learning on point clouds, *ACM Trans. on Graphics* **38**, 1 (2018).
31. C. Liu, Y. Zhan, X. Ma, D. Tao, B. Du and W. Hu, Masked graph auto-encoder constrained graph pooling, *ECML-PKDD* (2022).
32. K. Zhou, Y. Dong, W. S. Lee, B. Hooi, H. Xu and J. Feng, Effective training strategies for deep graph neural networks, *arXiv* (2020).
33. O. Chapelle, J. Weston, L. Bottou and V. Vapnik, Vicinal risk minimization, *NeurIPS* (2000).
34. A. Li, Brainmixup: Data augmentation for gnn-based functional brain network analysis, *IEEE Big Data* (2022).
35. A. D. Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto, B. Deen, S. Delmonte, I. Dinstein, B. Ertl-Wagner, D. A. Fair, L. Gallagher, D. P. Kennedy, C. L. Keown, C. Keysers, J. E. Lainhart, C. Lord, B. Luna, V. Menon, N. J. Minshew, C. S. Monk, S. Mueller, R.-A. Müller, M. B. Nebel, J. T. Nigg, K. O'Hearn, K. A. Pelphrey, S. J. Peltier, J. D. Rudie, S. Sunaert, M. Thioux, J. M. Tyszka, L. Q. Uddin, J. S. Verhoeven, N. Wenderoth, J. L. Wiggins, S. H. Mostofsky and M. P. Milham, The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism., *Mol Psychiatry* (2022).
36. D. C. V. Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub and K. Ugurbil, The wu-minn human connectome project: An overview., *NeuroImage* (2013).
37. C. Craddock, S. Sikka, B. Cheung, R. Khanuja, S. S. Ghosh, C. Yan, Q. Li, D. Lurie, J. Vogelstein, R. Burns *et al.*, Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (c-pac), *INCF Congress of Neuroinformatics.* (2014).
38. S. Gadgil, Q. Zhao, A. Pfefferbaum, E. V. Sullivan, E. Adeli and K. M. Pohl, Spatio-temporal graph convolution for resting-state fmri analysis, *National Library of Medicine* (2020).
39. U. Pervaiz, D. Vidaurre, C. Gohil, S. Smith and M. Woolrich, Multi-dynamic modelling reveals strongly time-varying resting fmri correlations, *Medical Image Analysis* **77** (2022).
40. A. S. Karampasi, A. D. Savva, V. C. Korfiatis, I. Kakkos and G. K. Matsopoulos, Informative biomarkers for autism spectrum disorder diagnosis in functional magnetic resonance imaging data on the default mode networkl networks, *Appl. Sci.* (2021).
41. T. M. Epalle, Y. Song, Z. Liu and Others, Multiatlas classification of autism spectrum disorder with hinge loss trained deep architectures: Abide i results, *Applied Soft Computing* **107**, p. 107375 (2011).
42. A. S. James Orme-Rogers, Spatio-temporal attention in multi-granular brain chronnectomes for detection of autism spectrum disorder, *arXiv* (2022).
43. P. Cao, G. Wen, L. Li, X. Liu, J. Yang and O. Zaiane, Temporal graph representation learning for autism spectrum disorder brain networks, *BIBM* (2022).
44. J.-J. K. Byung-Hoon Kim, Jong Chul Ye, Learning dynamic graph representation of brain connectome with spatio-temporal attention, *AAAI* (2021).
45. U. Mahmood, Z. Fu, V. Calhoun and S. Plis, Deep dynamic effective connectivity estimation from multivariate time series, *IJCNN* (2022).