NeuroGraph: Benchmarks for Graph Machine Learning in Brain Connectomics

A Benchmarks Availability and Licensing

The fMRI data utilized in this research was sourced from the Human Connectome Project [18]. The proposed graph-based benchmark datasets can be accessed for download at `https://anwar-said.github.io/anwarsaid/neurograph.html`. These datasets are provided in PyG[1] format, optimized for use with Graph Neural Networks (GNNs). However, they can also be conveniently incorporated into other platforms. Additionally, the associated code for downloading, preprocessing, and benchmarking is open to the public at `https://github.com/Anwar-Said/NeuroGraph`, complete with comprehensive documentation.

B NeuroGraph and Neuroimaging Data

Neuroimaging, a powerful field of study, enables researchers to delve into the complexities of the human brain by capturing detailed images and measurements. Recent advancements in technology have resulted in an abundance of neuroimaging data, particularly functional magnetic resonance imaging (fMRI), which offers invaluable insights into brain activity. However, understanding and analyzing fMRI data pose several challenges. Firstly, the high dimensionality of fMRI data presents a significant hurdle. Additionally, inherent noise and variability in fMRI signals can obscure underlying neural activity. Complex spatial and temporal dependencies further complicate fMRI data analysis, demanding advanced modeling techniques. Furthermore, the interpretation and analysis of fMRI data can be time-consuming and subjective. The graphical representation of fMRI data offers a plethora of opportunities to tackle these challenges. For instance, network science and graph theoretical approaches provide a diverse range of tools to explore brain regions and their connectivity patterns [17]. Furthermore, the application of graph machine learning techniques, such as GNNs are particularly well-suited for analyzing neuroimaging data and have the potential to provide valuable insights. The provision of graph-based neuroimaging benchmarks and computational tools play a crucial role to enhance the field, which is the main focus of this study.

B.1 fMRI Data Sources

Several initiatives have been undertaken in the past decade to assemble comprehensive fMRI datasets. One notable source is the Human Connectome Project (HCP) dataset [18]. The HCP dataset offers an extensive collection of multimodal neuroimaging data, including resting-state fMRI, task-based fMRI, and structural MRI scans, from a large cohort of healthy individuals. In addition to large neuroimaging datasets curated by institutions or projects, some notable resources are OpenNeuro, OpenfMRI and fcon_1000[2] platforms, which host a diverse range of publicly available fMRI datasets contributed by researchers worldwide [13, 12]. These datasets cover various experimental paradigms (see Table 1), clinical populations, and research domains, providing researchers with a wealth of data for analysis and investigation.

[1] `https://pyg.org/`

[2] `http://fcon_1000.projects.nitrc.org/`

37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks.

Table 1: Description of fMRI paradigms in HCP Young Adult dataset.

Dataset	Volumes per run (TR)	Run duration (min)	Duration of task blocks (sec)	Description
Rest	1200	14 : 24	-	no stimuli
Working Memory	405	5 : 01	25	0-back, 2-back
Gambling	253	3 : 12	28	win, loss
Motor	284	3 : 34	12	various body parts
Language	316	3 : 57	approx. 30	story, math
Relational Processing	232	2 : 56	16	relational, control
Social Cognition	274	3 : 27	23	interaction, no interaction
Emotion Processing	176	2 : 16	18	face, shape

Table 2: fMRI scans required disk storage. The storage information is obtained from Human Connectome Project website.

Task	Storage (GB)
Rest	1260.95
Working Memory	527.70
Gambling	387.38
Motor	415.81
Language	426.72
Relational	343.40
Social	386.76
Emotion	295.91

We have chosen to utilize the HCP S1200 dataset from the Brain Connectome as a primary resource for our graph-based benchmarking [18]. This dataset is well-suited for graph-based benchmarking due to its extensive coverage of brain regions and their interconnections. Additionally, the HCP S1200 dataset provides valuable demographic and behavioral information, enabling comprehensive analyses that consider various factors influencing brain connectivity. Its wide availability and standardized processing pipelines further contribute to its suitability for graph-based benchmarking, ensuring consistency and comparability across studies. Thus, the HCP S1200 dataset from the Brain Connectome represents a robust choice for conducting graph-based benchmarking studies in the field of neuroimaging.

B.2 Reading HCP Dataset

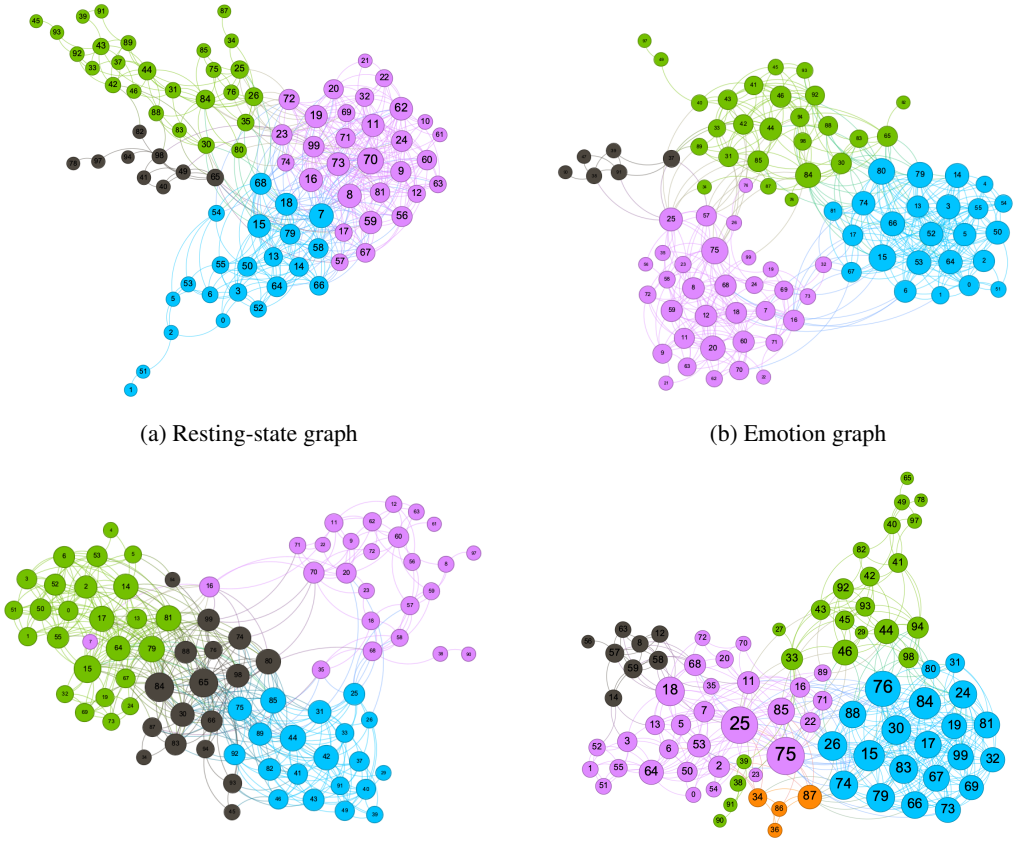
Storing and reading fMRI datasets presents a formidable challenge due to their substantial storage requirements, necessitating significant disk space allocation, e.g., each subject of HCP S1200 requires 1.1 GB of space on disk. Moreover, the preprocessing of fMRI data calls for tools that are not only user-friendly but also highly efficient. Fortunately, the Human Connectome Database (HCP) offers an AWS instance (s3 bucket) that allows for seamless data crawling. NeuroGraph, with its implementation utilizing the boto3 Python package, provides an efficient solution for crawling the dataset. Boto3, a widely used Python package, enables seamless interaction with AWS services, facilitating efficient data retrieval and preprocessing in the NeuroGraph framework. Our implementation offers users the flexibility to either store the datasets or preprocess them on the fly if storage space is limited (see Table 2 for disk storage). To access the HCP data, users are required to obtain credentials from HCP[3] and provide them to NeuroGraph. Moreover, NeuroGraph also provides a Python class for preprocessing data from the local storage.

B.3 Data Preprocessing

In close collaboration with domain experts from both the neuroimaging and graph machine learning fields, NeuroGraph's preprocessing pipeline is divided into five stages. These stages ensure the quality and reliability of the fMRI data. Initially, we utilize data that has already been processed using the HCP minimal processing pipeline [6].

- **Step 1 - Brain Parcellation:** The first phase of our pipeline involves brain parcellation, a process that divides the brain into smaller regions or parcels. This step allows for the analysis of functional connectivity within and between these parcels. In our study, we employ the Schaefer atlases [15], widely used brain parcellation schemes that define neurobiologically meaningful features of brain organization. These atlases provide a parcellation of the

[3] `https://db.humanconnectome.org`



(a) Resting-state graph

(b) Emotion graph

(c) Gambling graph

(d) Language graph

Figure 1: Visualization of the corresponding simple undirected graphs with 100 ROIs for a single subject during both the rest condition and while performing certain tasks. Note that the coloring of the graphs has been applied based on the community structure, but solely for visualization purposes. Isolated nodes were removed.

cerebral cortex into hierarchically organized regions at multiple resolutions. Using the population level atlases, we extract the mean fMRI timeseries for each region of interest (ROI). This provides a representative measure of the average neural activity within each specific brain region, enabling subsequent connectivity analyses.

- **Step 2 - Remove Scanner Drifts and Motion Artifacts:** Next, we remove linear and quadratic trends along with six rigid-body head motion parameters and their derivatives, from the fMRI data. Removal of the trends aims to remove the scanner drifts in the fMRI signals that arise from instrumental factors. Removal of the motion parameters, that capture the movement and rotation of the subject's head during the scanning session, ensures that any potential confounding effects are minimized. By eliminating these artifacts, we enhance the signal-to-noise ratio and increase the sensitivity to neural activity.
- **Step 3 - Subject-Level Signal Normalization:** We perform subject-level normalization of the ROI timeseries signals. Specifically, we temporally normalize all subject signals to zero mean and unit variance. This step allows for fair comparisons and facilitates the identification of meaningful variations in the functional connectivity patterns across subjects.
- **Step 4 - Calculate Correlation Matrix:** We compute the correlation matrices from the ROI timeseries signals. Correlation matrices capture the strength of functional connectivity between different ROIs. By calculating pairwise correlations between the timeseries signals of each ROI, we obtain a matrix that represents the interregional functional connections within the brain. This step allows us to quantify and analyze the patterns of functional

connectivity across the entire brain, and construct a graph. The correlation matrices serve as a valuable tool for investigating the network-level organization of the brain and identifying regions that exhibit synchronous activity [4]. These matrices provide a representation of the functional architecture and can be further utilized for graph-based analyses, such as network characterization and identification of key brain hubs [4, 7]. In Figure 1, we provide the visualizations of the graphs correspond to one subject in certain conditions.

- **Step 5 - Construct Static/Dynamic Attributed Graphs:** Finally, we compute two types of graph-based datasets from the functional connectivity matrix: static and dynamic graphs. As discussed in Section 3 of the paper, the static graph is defined as $G = (\mathcal{V}, \mathcal{E}, X)$. Here, the node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ represents ROIs, while the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes positive correlations between pairs of ROIs, as determined by a predefined threshold. The feature matrix is represented by $X^{n \times d}$, where n symbolizes the total number of ROIs, and d corresponds to the dimension of the feature vector. We explore the dataset generation search space by considering different numbers of ROIs, different thresholds, and node features to identify optimal parameters. The next section provides a comprehensive overview of the dataset construction search space.

Regarding the parameter setup for constructing our benchmark datasets, we opt for a sparse setup (top 5%) with 1000 ROIs for the HCP-Gender, HCP-Age, HCP-WM, and HCP-FI datasets. However, for the HCP-Task dataset, we reduce the number of ROIs to 400 in order to manage memory overhead. In the dynamic setting, we employ a sliding window approach with a fixed window length (Γ) set to 50 and a stride of 3. Considering memory constraints and computational overhead, we fix the dynamic length (l) to 150 and slide over the preprocessed timeseries matrix to construct dynamic graphs. For all dynamic graphs, we consider 100 ROIs and medium sparsity (top 10%). With this setting, the total number of dynamic graphs we obtain for each subject is $((l - \Gamma)/stride) + 1$.

B.4 The Design Space is Vast

The design space for constructing graphs from correlation matrices is substantial, given the multitude of available methods. We can construct diverse graph types employing various strategies. For instance, some of the potential graph types to consider include simple undirected graphs as demonstrated in [8], weighted graphs[11], attributed graphs [3], and minimum spanning trees [1, 20], among others. Similarly, a range of parameters comes into play during this process, further expanding the design space for these constructions. These parameters include the number of ROIs, edge weights, density thresholding for edge selection, and node features, to name a few.

GNNs have shown considerable promise in handling attributed graphs, demonstrating their effectiveness in various domains [11, 14]. Attributed graphs, which include not only the graph topology but also node-level features, represent complex systems more accurately than simple graph. GNNs leverage these attributes to capture both local and global structural information, allowing for the development of more comprehensive graph representations. Considering the importance of attributed graphs, we opted to construct rich, brain attributed graphs.

Node Features: Traditional methods for representing node features in graphs include using coordinates [10], one-hot encoding [7], and mean activation [5, 10]. Coordinates serve to provide spatial information about the nodes, while one-hot encoding are used for categorical features, effectively distinguishing different node types. Mean activation, on the other hand, can give insights about the average level of a node's activity or influence. While these methods provide a base level of information, they may not fully capture the rich complexity inherent in many data structures, such as brain graphs. To address this, we explore more powerful ways of representing node features, including using correlation vectors, BOLD signals and the combination of both. Correlation vectors can encapsulate the relationship between different nodes, providing insight into the connectivity and interaction within the graph. BOLD signals, give information about changes in blood flow in the brain, which can be an indicator of neural activity. By combining both of them, we may enrich models with a wealth of information, thereby capturing the intricate details and relationships present in brain graphs.

Number of ROIs: ROIs in brain graph construction may significantly impacts the granularity and overall scope of the resulting graph. Using a smaller number of ROIs, such as 100, can lead to a more generalized and coarser view of brain connectivity. This simplified perspective can be useful for broad overviews and initial exploration but might overlook intricate local interactions or specific

clusters of activity. Conversely, using a larger number of ROIs, such as 400 or 1000, allows for a more detailed and finer representation of the brain's connectivity. With more ROIs, the graph can capture more specific interconnections, potentially revealing sub-networks or localized activity patterns that a coarser graph might miss. However, larger graphs also present a challenge in terms of computational load and complexity, also prone to noise. Interestingly, different methods in the literature have adopted different numbers of ROIs for their analysis [8, 11, 3]. These varying approaches underscore the fact that the choice of ROIs number is not merely a matter of computational convenience, but can significantly influence the outcomes of the study.

In light of this, our research aims to explore these three ROIs sizes: 100, 400, and 1000. Our goal is to understand the impact of different graph granularity levels on the performance of GNNs. By doing so, we hope to provide deeper insights into how different levels of detail in the graph structure affect the GNN's ability to capture and model brain connectivity. This investigation could potentially guide the selection of an optimal ROI size in future brain graph studies, striking a balance between capturing sufficient detail and maintaining computational feasibility.

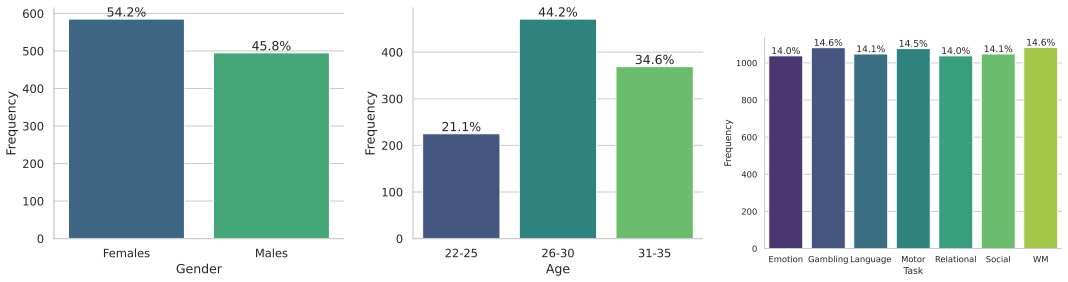
Density Thresholding: Graph density is a fundamental property that may impacts the performance of GNNs. Graph density refers to the proportion of the possible connections in a graph that are actual connections. It influences how information is propagated through the network, may potentially affect the accuracy and efficiency of the GNN. A sparse graph (low-density) might lead to information underflow, with some nodes being poorly connected, which might cause inadequate learning of node representations. On the other hand, a dense graph (high-density) could lead to an information overflow, with a significant amount of information being propagated, possibly causing noise and overfitting [9].

Thresholding, on the other hand, is a crucial step in the construction of brain graphs. It's used to determine which correlations are strong enough to be included as edges in the graph. There are several approaches to thresholding. One is absolute thresholding, where a fixed threshold value is selected, and all correlations in the matrix above this threshold are included as edges in the graph. However, the choice of an absolute threshold can be somewhat arbitrary, and may result in graphs of varying sizes and densities. This variability can complicate comparisons between graphs [2]. Proportional thresholding is another method, in which the strongest $x\%$ of correlations are included as edges in the graph. This method ensures that all resultant graphs have the same density of edges, which facilitates comparisons between them. However, it can also result in the inclusion of weak, potentially non-significant correlations in the graph. To avoid this issue, some studies consider only positive correlations, which allows the construction of graphs with various densities and avoids the complications of negative thresholding [19].

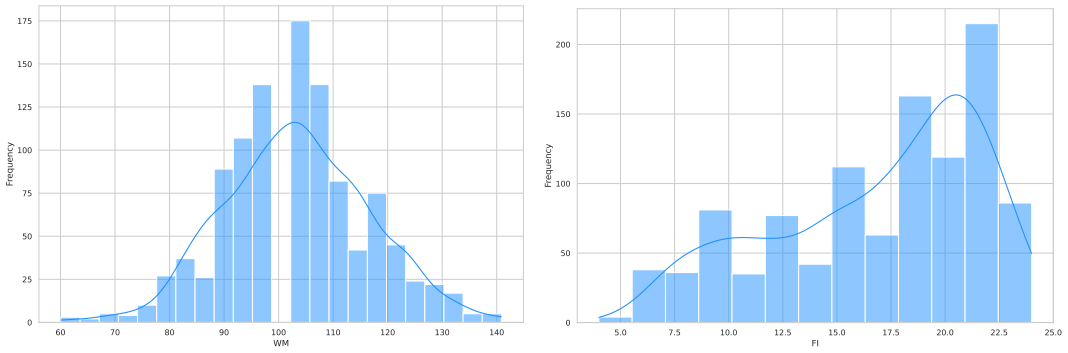
Indeed, there are numerous ways to conduct thresholding in brain graph construction, with several options available within each thresholding approach. Each method and option presents its unique set of advantages and potential limitations. In this context, we focus on proportional thresholding with positive correlations, an approach that has shown encouraging results in previous research [11, 8]. Specifically, we explore three levels of density: those defined by the top 5%, 10%, and 20% percentile values from the correlation matrices. These densities represent different levels of graph sparsity, offering a broad perspective on how the choice of threshold can impact the topology and interpretability of the resulting brain networks. We note that the terms "sparse" (5%) and "dense" (20%) are relative and dependent on the context of feasible ranges. Despite their different percentages of edges, both sparse and dense graphs exhibit a complexity of $O(n^2)$ edges. We observed that even in sparse datasets, the average degree is around 50 for 1000 ROIs, indicating a substantial level of connectivity.

C NeuroGraph Benchmark Datasets

We propose a collection of ten datasets tailored to five distinct tasks, encompassing both static and dynamic contexts. These tasks are identified as HCP-Task, DynHCP-Task, HCP-Gender, DynHCP-Gender, HCP-Age, DynHCP-Age, HCP-WM, DynHCP-WM, HCP-FI, and DynHCP-FI. These datasets are derived from the HCP S1200 dataset, following a sequence of preprocessing operations. For the creation of static datasets, we eliminated two subjects that contained fewer than 1200 scans and then applied the preprocessing as outlined in the previous sections. The resulting datasets are represented as sparse matrices with 1000 ROIs. However, we've tailored the Activity dataset to



(a) HCP-Gender (b) HCP-Age (c) HCP-Task



(d) HCP-WM (e) HCP-FI

Figure 2: Illustration of class distribution for each dataset. For the regression task, histograms are presented to depict the frequency distributions of both Working Memory (WM) and Fluid Intelligence (FI) scores. In addition to these, Kernel Density Estimates are superimposed on the histograms, providing a smoother representation of the distributions.

include only 400 ROIs owing to its larger size of over 7000 scans, as this adaptation was necessary to overcome memory constraints. As for the dynamic datasets, we've standardized the dynamic length to 150, with a window size of 50 and a stride of 3. Moreover, to alleviate the substantial memory demands, we've limited the dynamic datasets to encompass only 100 ROIs. The distribution of classes for each dataset, as well as the values for regression tasks, have been visualized and are presented in Figure 2.

C.1 GNN* and Dynamic Graph Baselines

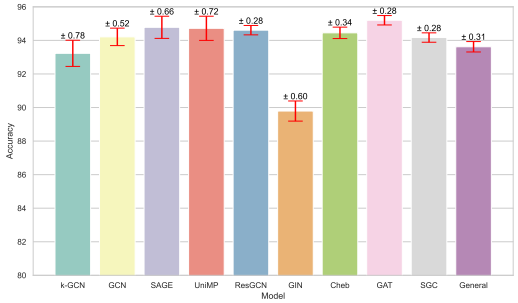
Our study also explores a variation of residual GNNs, we named GNN*, the model that leverages both residual connections and a feature concatenation approach, enhancing the utilization of the functional connectome in the training process. As delineated in Section 3.4 and visualized in Figure 2 of the main paper, GNN* employs a universal graph convolution layer, facilitating the use of any GNN convolution contingent on the project's requirements. Similarly, the dynamic graph baseline (depicted in Figure 2 of the main paper) also uses a general graph convolution, followed by a Transformer module. Throughout our experimentation, we employed UniMP with GNN* and tested five models using the dynamic baseline, the results of which are tabulated in Table 6 of the main paper. All other parameters remain consistent with the detailed exposition in the experimental setup (Section 5.1) of the main paper.

D Memory and Running Time Analysis

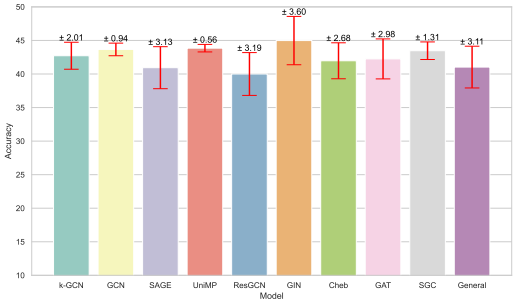
Following a comprehensive and rigorous exploration of the search space, we have identified and established optimal datasets that strike a balance between minimizing memory requirements and maintaining an effective quantity of parameters. The trade-off achieved ensures that models are able to run smoothly on machines with reasonable computing power on our datasets, making them highly accessible to a wide range of users. This optimization also yields the additional benefit of reduced

Table 3: Resource utilization analysis of UniMP model on all benchmark datasets

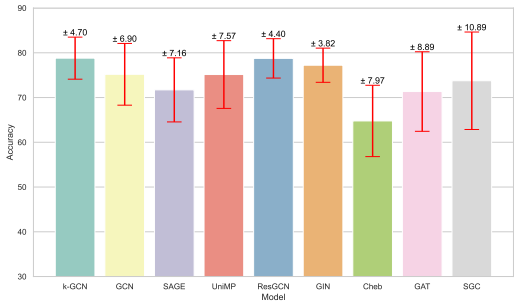
Benchmark Dataset	Disk storage (GB)	#Parameters	Memory (MB)	Training time (sec)
HCP-Task	4.0	265035	2463	854
HCP-Gender	3.7	648870	6437	362
HCP-Age	3.6	648903	4293	355
HCP-WM	3.7	803461	6551	696
HCP-FI	3.6	803461	6762	690
DynHCP-Task	7.3	309575	15881	11200
DynHCP-Gender	1.1	308930	4169	1700
DynHCP-Age	1.0	309059	4113	1709
DynHCP-WM	1.1	308801	4359	1704
DynHCP-FI	1.0	308801	4335	1712



(a) HCP-Task



(b) HCP-Age



(c) HCP-Gender

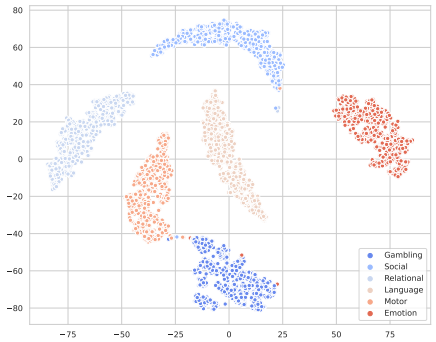
Figure 3: Models' performance: Accuracy and standard deviation on 10 runs with different seeds on HCP-Task, HCP-Age and HCP-Gender datasets.

training times; our models are capable of training in mere minutes, significantly accelerating the model development cycle and promoting rapid iterative progress.

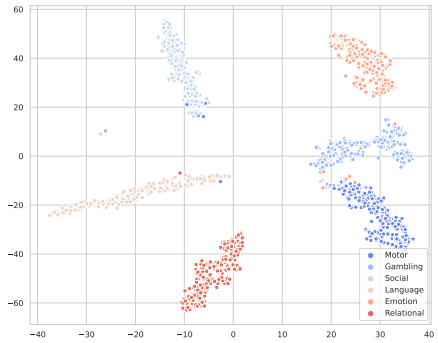
The specifics of this optimization are illustrated in the context of Unified Message Passing (UniMP) model [16], which we use to showcase the efficient resource usage of our datasets and approach. In Table 3, we offer detailed insights into the running times and memory requirements of UniMP model. We executed UniMP on each dataset for 100 epochs and recorded both GPU memory utilization and overall training time, which includes data loading. The number of hidden units for the GNN layer was 32 and 128 for the MLP layers. These data points provide a tangible representation of the efficiency gains achieved through our dataset size optimization process. Such optimizations are instrumental in ensuring datasets are not only computationally effective using any model but also highly accessible, enabling broader applicability for a variety of hardware configurations. All experiments were executed on a system equipped with an Intel(R) Xeon(R) Gold 6238R CPU operating at 2.20GHz with 112 cores, 512 GB of RAM, and an NVIDIA A40 GPU with 48GB of memory.

E Models Performance and Standard Error

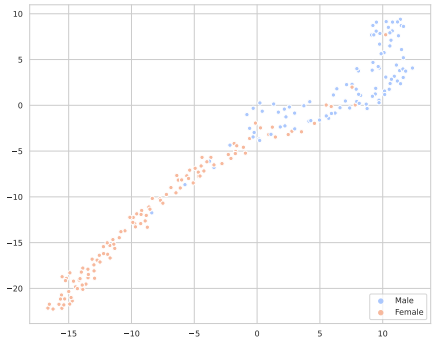
We plot the accuracy along with the standard deviation of 10 runs, each with different seeds, for all the models on three distinct datasets: HCP-Task, HCP-Age and HCP-Gender in Figure 3. We observed that the results reported a higher level of stability on both HCP-Task and HCP-Age datasets.



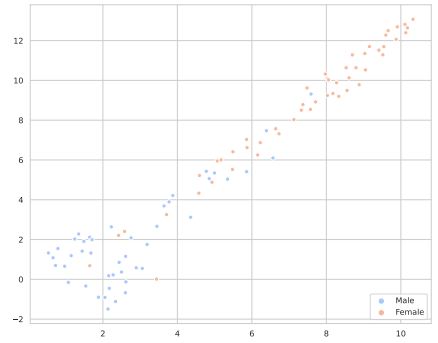
(a) HCP-Task test set



(b) HCP-Task validation set



(c) HCP-Gender test set



(d) HCP-Gender validation set

Figure 4: Hidden layer activation on test and validation sets of HCP-Task and HCP-Gender.

This indicates that the models performed consistently and yielded more reliable results, suggesting a greater degree of confidence in the accuracy measurements. On the HCP-Gender dataset, we observed slightly high standard errors across the models. Moreover, we provide the visualization of the hidden activations obtained from the last layer of GNN^* for the test and validation sets trained on HCP-Task and HCP-Gender datasets in Figure 4. We used TSNE for these visualizations.

References

[1] Giovanni Bonanno, Guido Caldarelli, Fabrizio Lillo, and Rosario N Mantegna. Topology of correlation-based minimal spanning trees in real and model markets. *Physical Review E*, 68(4):046130, 2003.

[2] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3):186–198, 2009.

[3] Hejie Cui, Wei Dai, Yanqiao Zhu, Xuan Kan, Antonio Aodong Chen Gu, Joshua Lukemire, Liang Zhan, Lifang He, Ying Guo, and Carl Yang. Braingb: a benchmark for brain network analysis with graph neural networks. *IEEE Transactions on Medical Imaging*, 2022.

[4] Andrew T Drysdale, Logan Grosenick, Jonathan Downar, Katharine Dunlop, Farrokh Mansouri, Yue Meng, Robert N Fetcho, Benjamin Zebley, Desmond J Oathes, Amit Etkin, et al. Resting-state connectivity biomarkers define neurophysiological subtypes of depression. *Nature medicine*, 23(1):28–38, 2017.

[5] Soham Gadgil, Qingyu Zhao, Adolf Pfefferbaum, Edith V Sullivan, Ehsan Adeli, and Kilian M Pohl. Spatio-temporal graph convolution for resting-state fmri analysis. In *Medical Image*

Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23, pages 528–538. Springer, 2020.

[6] Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.

[7] Byung-Hoon Kim and Jong Chul Ye. Understanding graph isomorphism network for rs-fmri functional connectivity analysis. *Frontiers in neuroscience*, page 630, 2020.

[8] Byung-Hoon Kim, Jong Chul Ye, and Jae-Jin Kim. Learning dynamic graph representation of brain connectome with spatio-temporal attention. *Advances in Neural Information Processing Systems*, 34:4314–4327, 2021.

[9] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[10] Xiaoxiao Li, Nicha C Dvornek, Yuan Zhou, Juntang Zhuang, Pamela Ventola, and James S Duncan. Graph neural network for interpreting task-fmri biomarkers. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part V 22*, pages 485–493. Springer, 2019.

[11] Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021.

[12] Christopher J Markiewicz, Krzysztof J Gorgolewski, Franklin Feingold, Ross Blair, Yaroslav O Halchenko, Eric Miller, Nell Hardcastle, Joe Wexler, Oscar Esteban, Mathias Goncavles, et al. The openneuro resource for sharing of neuroscience data. *Elife*, 10:e71774, 2021.

[13] Russell A Poldrack, Deanna M Barch, Jason P Mitchell, Tor D Wager, Anthony D Wagner, Joseph T Devlin, Chad Cumba, Oluwasanmi Koyejo, and Michael P Milham. Toward open sharing of task-based fmri data: the openfmri project. *Frontiers in neuroinformatics*, 7:12, 2013.

[14] Anwar Said, Mudassir Shabbir, Saeed-Ul Hassan, Zohair Raza Hassan, Ammar Ahmed, and Xenofon Koutsoukos. On augmenting topological graph representations for attributed graphs. *Applied Soft Computing*, 136:110104, 2023.

[15] Alexander Schaefer, Ru Kong, Evan M Gordon, Timothy O Laumann, Xi-Nian Zuo, Avram J Holmes, Simon B Eickhoff, and BT Thomas Yeo. Local-global parcellation of the human cerebral cortex from intrinsic functional connectivity mri. *Cerebral cortex*, 28(9):3095–3114, 2018.

[16] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.

[17] Cornelis J Stam. Modern network science of neurological disorders. *Nature Reviews Neuroscience*, 15(10):683–695, 2014.

[18] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[19] Bernadette CM Van Wijk, Cornelis J Stam, and Andreas Daffertshofer. Comparing brain networks of different size and connectivity density using graph theory. *PloS one*, 5(10):e13701, 2010.

[20] Ke Zeng, Jiannan Kang, Gaoxiang Ouyang, Jingqing Li, Junxia Han, Yao Wang, Estate M Sokhadze, Manuel F Casanova, and Xiaoli Li. Disrupted brain network in children with autism spectrum disorder. *Scientific reports*, 7(1):16253, 2017.