NeuroGraph: Benchmarks for Graph Machine Learning in Brain Connectomics

Anwar Said, Roza G. Bayrak, Tyler Derr, Mudassir Shabbir,
Daniel Moyer, Catie Chang, Xenofon Koutsoukos
Department of Computer Science
Vanderbilt University, USA
{anwar.said, roza.g.bayrak, tyler.derr, mudassir.shabbir}@vanderbilt.edu
{daniel.moyer, catie.chang, xenofon.koutsouks}@vanderbilt.edu

Abstract

Machine learning provides a valuable tool for analyzing high-dimensional functional neuroimaging data, and is proving effective in predicting various neurological conditions, psychiatric disorders, and cognitive patterns. In functional Magnetic Resonance Imaging (MRI) research, interactions between brain regions are commonly modeled using graph-based representations. The potency of graph machine learning methods has been established across myriad domains, marking a transformative step in data interpretation and predictive modeling. Yet, despite their promise, the transposition of these techniques to the neuroimaging domain remains surprisingly under-explored due to the expansive preprocessing pipeline and large parameter search space for graph-based datasets construction. In this paper, we introduce NeuroGraph, a collection of graph-based neuroimaging datasets that span multiple categories of behavioral and cognitive traits. We delve deeply into the dataset generation search space by crafting 35 datasets within both static and dynamic contexts, running in excess of 15 baseline methods for benchmarking. Additionally, we provide generic frameworks for learning on dynamic as well as static graphs. Our extensive experiments lead to several key observations. Notably, using correlation vectors as node features, incorporating larger number of regions of interest, and employing sparser graphs lead to improved performance. To foster further advancements in graph-based data driven Neuroimaging, we offer a comprehensive open source Python package that includes the datasets, baseline implementations, model training, and standard evaluation. The package is publicly accessible at `https://anwar-said.github.io/anwarsaid/neurograph.html`.

1 Introduction

Graph Neural Networks (GNNs) have demonstrated remarkable efficacy in a variety of domains including recommendations, forecasting and the analysis of functional Magnetic Resonance Imaging (fMRI) data [56, 25, 32]. In human neuroimaging research, GNNs have proven valuable in capturing the complex connectivity patterns within the brain's functional networks [13, 32]. By examining the spontaneous and synchronized fluctuations of the magnetic resonance signals, fMRI provides a useful means of measuring functional network connectivity [17].

Neuroimaging and Graph Machine Learning (GML) are two rapidly evolving fields with immense potential for mutual collaboration. However, a significant challenge lies in bridging the gap between these domains and enabling seamless integration of neuroimaging data into state-of-the-art GML approaches [27]. This gap is primarily attributed to the expansive fMRI data preprocessing pipeline, the absence of interface for creating articulate graph representation datasets, and a limited understand-

Preprint. Under review.

Table 1: Dataset statistics. d indicates degree, K global clustering coefficient, $|X|$ represents number of node features and $|Y|$ indicates the number of classes.

	Dataset	Statistics						$\|X\|$	$\|Y\|$	Task
		$\|G\|$	$\|N\|_{avg}$	$\|E\|_{avg}$	d_{max}	d_{avg}	K			
Static	HCP-Activity	7443	400	7029.18	153	19.40	0.41	400	7	Graph Classification
	HCP-Gender	1078	1000	45578.61	413	45.78	0.46	1000	2	Graph Classification
	HCP-Age	1065	1000	45588.40	413	45.78	0.46	1000	3	Graph Classification
	HCP-FI	1071	1000	45573.67	413	45.78	0.46	1000	-	Graph Regression
	HCP-WM	1078	1000	45578.61	413	45.78	0.46	1000	-	Graph Regression
Dynamic	DynHCP-Activity	7443	100	843.04	992	6.22	0.427	100	7	Graph Classification
	DynHCP-Gender	1080	100	874.88	992	9.26	0.439	100	2	Graph Classification
	DynHCP-Age	1067	100	875.42	992	9.26	0.439	100	3	Graph Classification
	DynHCP-FI	1073	100	874.82	992	9.26	0.438	100	-	Graph Regression
	DynHCP-WM	1080	100	874.88	992	9.26	0.439	100	-	Graph Regression

ing of the practical applications of graph machine learning to neuroimaging [25]. To address these challenges, the principal objectives of this study include a careful exploration of the graph-based dataset generation, with the goal of formulating a strategic road map for transitioning from fMRI data to a graph-based representation paradigm. Secondly, we conduct a rigorous evaluation of graph machine learning methodologies, with a special emphasis on GNNs, examining their efficacy when applied to diverse fMRI data configurations.

The human brain, a complex network of interconnected regions, can be represented as a graph, wherein nodes correspond to contiguous segments known as Regions of Interest (ROIs), and edges represent their relationships [10, 7]. Features of the functional connectome, such as correlations between the BOLD (Blood Oxygen Level Dependent) signals between different brain regions, typically employed for downstream machine learning tasks [27, 1], can be re-envisioned as node features within attributed graph representations. These representations pave the way for a rich assortment of graph-based data representations, wherein GNNs are exceptionally well-suited [21]. Yet, the vast potential offered by the intersection of fMRI datasets and GNNs remains untapped, due primarily to the expansive search space for data generation and the multifaceted nature of hyperparameters. In this study, we pioneer a rigorous exploration and benchmarking for GNNs, with the following primary contributions:

- We introduce NeuroGraph, a collection of static and dynamic brain connectome datasets tailored for benchmarking GNNs in classification and regression tasks including gender and age classification, mental state decoding, and prediction of fluid intelligence and working memory scores. This enables an extensive exploration of brain connectivity and its associations with various cognitive, behavioral, and demographic variables. Details of the proposed datasets are provided in Table 1.
- We perform an extensive exploratory study in search of optimal graph-based data representations for Neuroimaging data, implementing 15 baseline models on 35 different datasets. Additionally, we provide detailed benchmarking for the datasets we propose.

By offering NeuroGraph, we create an essential road map between the neuroimaging and graph machine learning communities. Researchers in the neuroimaging field can now tap into the power of cutting-edge GNNs. Our datasets generation pipeline serves as a road map, guiding researchers on how to effectively transform neuroimaging data into a standard graph representation suitable for graph machine learning. This integration facilitates the adoption of state-of-the-art graph-based techniques, unlocking new insights and accelerating discoveries in the field of Neuroimaging.

2 Related Work

While functional brain connectomes have long been recognized as a rich source of information in neuroscience and neuroinformatics [44], their value has become increasingly evident in recent years [13]. Propelled by growth in data availability and methodological breakthroughs, ML has shown remarkable efficacy on tasks such as prediction of cognitive function [37], identification of mental health disorders [15] , and understanding of brain aging [16]. However, these methods utilize the functional connectivity of the matrix while ignoring the relational information among the brain regions which could potentially aid to the modelling process.

GNNS for static graphs: GNNs have significantly evolved as a major field of exploration, offering an intuitive approach in learning from graph-structured data [26, 20, 14, 6, 36]. In a static setting, where individual data points are represented by single graphs, a variety of methods have been introduced

[49, 6, 12, 36, 43, 40]. Recent studies have demonstrated the effectiveness of various approaches when applied to functional connectome data, which can be represented as different types of graphs, including weighted graphs [32, 9, 22], and attributed graphs [25, 2], among others. By leveraging the structured and relational nature of the data, GNNs not only enable learning from the functional connectivity matrix but also enhance the overall capabilities of the models [1, 11, 36, 56].

Dynamic graph representations: The field of learning dynamic graph representations in a graph classification setting remains relatively unexplored, especially in the realm of brain imaging [51]. In neuroimaging, dynamic graphs are constructed to capture the time-varying interactions and connectivity patterns in the brain [25, 41, 39]. Despite this relative lack of exploration, recent years have witnessed the emergence several methods that have demonstrated remarkable results when applied to brain graphs [25, 29, 55, 8]. These methods have showcased the potential of effectively capturing and analyzing the dynamic nature of brain connectivity, opening up new avenues for advancements in our understanding of brain function and neurological processes.

3 NeuroGraph

A few recent efforts have been made to utilize GNNs for predictive modeling on Neuroimaging data. However, there is no consensus on the preprocessing pipeline and hyper-parameter configuration for deriving expressive graph-based brain datasets [25, 32, 18, 22]. In addition, although there are a multitude of GNNs models, no benchmark datasets have been created to evaluate GML approaches on brain connectome data. To fill this gap and provide a common ground, we use publicly available datasets and only minimally preprocess the data using standard fMRI preprocessing steps.

3.1 From fMRI to Graph Representations

fMRI data is typically represented in four dimensions, where the blood-oxygen level-dependent (BOLD) signal is captured over time in a series of 3-dimensional volumes. These volumes display the intensity of the BOLD signal for different spatial locations in the brain. However, since brain activity tends to exhibit strong spatial correlations, the BOLD signal is often summarized into a collection of special functional units, *brain parcels*. These units represent *regions of interest* (ROIs) whose constituent "voxels" (a smallest three dimensional resolution) exhibit temporally correlated activity.

The Human Connectome Project (HCP) [47] is a publicly available rich neuroimaging dataset containing not only imaging data but also a battery of behavioral and cognitive data. We select this dataset for benchmarking and utilize the established group level Schaefer [42] atlases to represent the measured BOLD signal. These atlases provide a parcellation of the cerebral cortex into hierarchically organized regions at multiple granularities (resolutions).

We use resting-state and seven task fMRI paradigms from the HCP 1200 dataset. All fMRI scans underwent the HCP minimal preprocessing pipeline [19]. We further regressed out six rigid-body head motion parameters and their derivatives, as well as the low-order trends, from the minimally preprocessed data. The mean fMRI time series was extracted from all voxels within each ROI for different parcellation schemes. Individual (subject-wise) ROI time-series signals were temporally normalized to zero mean and unit variance.

Our study of these datasets encompasses two distinct modes of analysis: *static* and *dynamic* graph construction. We apply different GNNs to both types and perform benchmarking in five unique tasks. In the static graph construction, we investigate multiple parameters to build the graphs from the raw data, taking into consideration variations in node features, the number of nodes or regions of interest (ROIs), and the density of the graph. For node features, we take into account correlations, time-series signals, or a blend of both. For the number of nodes provided by [42] (i.e., ROIs), we examine three different resolutions: $100, 400$, and 1000 nodes. As for graph density, we consider sparse, medium, and dense configurations. For the sparse setup, we choose the top 5% of values from the correlation matrix for edge selection, whereas for the medium and dense setups, we select the top 10% and 20% of values, respectively. We note that there are numerous methods for constructing brain graphs; however, we've opted for those more likely to yield superior performance [32, 25]. Additional details about the complexity of the search space in dataset construction and the rationale behind these parameters are presented in the supplementary material. We test 10 GNN methods to find the suitable combination of parameters and use a total of 15 baseline methods for benchmarking.

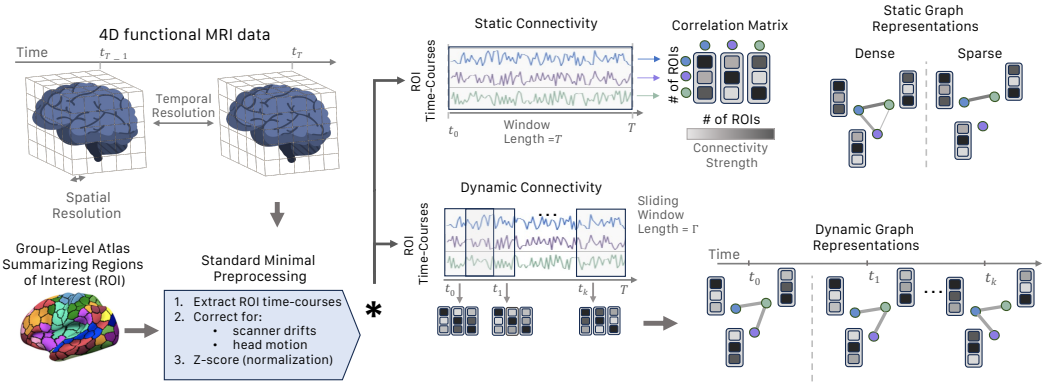


Figure 1: An illustration of the preprocessing pipeline, demonstrating the transition from fMRI data to the construction of both static and dynamic graphs.

Using the optimal combination of parameters in the static setting, we generate benchmark datasets for corresponding tasks in the dynamic setting. In the subsequent sections, we first describe the generation of graph-based datasets, followed by the description of each task.

3.2 Graph Representation

The static graph representation encompasses the conventional methodology of generating a static functional connectome graph from an fMRI scan, see supplementary materials for additional details. We define a connectome graph as $G = (\mathcal{V}, \mathcal{E}, X)$, wherein the node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ represents ROIs, while the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents positive correlations between pairs of ROIs, determined via a defined threshold. The feature matrix is denoted by $X^{n \times d}$, where n signifies the total number of ROIs and d refers to the feature vector's dimension. Subsequently, we define a representation vector $\boldsymbol{h}_G$ for the graph G, obtained via a GNN with an objective to perform the desired downstream machine learning task.

fMRI data comprise numerous timepoints within a scan, permitting the construction of dynamic graphs and thereby emphasizing the temporal information encapsulated within the data. This strategy has been evidenced to be notably effective within the literature [25]. Within the dynamic context, we define a sequence of brain graphs over T timepoints, denoted as $\mathcal{G} = \{G_1, G_2, \ldots G_T\}$, wherein each graph G_t captured at index t to $t + \Gamma$ from the fMRI scan. Here, Γ signifies the window length, set to 50 with a stride of 3 in our experiments. This setup allows us to capture functional connectivity within 36 seconds every 2.16 seconds, adhering to the standard protocol for sliding-window analyses as outlined in [39]. To alleviate computational load and memory during training, we followed the approach from [25], and randomly sliced the time dimension of the ROI-timeseries matrix at each step, maintaining a fixed length 150 and use 100 ROIs for the dynamic datasets. The procedure for constructing a graph for each timepoint parallels the one applied to the static graph. Subsequently, $\mathcal{G}$ can be utilized to procure a dynamic graph representation h_{dyn} to execute the desired downstream ML job. We refer the reader to supplementary material for further details.

3.3 Benchmark Datasets

The datasets are primarily divided into three main categories: those constructed for classification of demographics and brain states, and those constructed for predicting cognitive traits. Each category encapsulates distinct aspects of the collected data and serves unique analytical purposes. Detailed discussions of these categories will be provided in the subsequent sections with some basic statistics presented in Table 1. For more detail, readers are referred to the supplementary materials.

Predicting Demographics: The category of demographic estimation in our dataset is comprised of gender, and age estimation [18]. The gender attribute facilitates a binary classification with the categories being male and female. Age is categorized into three distinct groups as in [9]: 22-25, 26-30, and 31-35 years. A fourth category for ages 36 and above was eliminated as it contained only 14 subjects (0.09%), to maintain a reasonably balanced dataset. We introduce four datasets named:

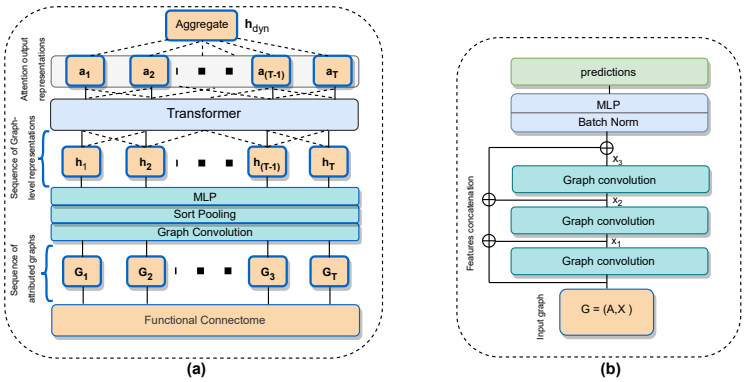


Figure 2: (a). Illustration of the architecture for learning dynamic graph representations. (b). Visualization of the GNN* architecture featuring residual connections and concatenated features.

HCP-Gender, HCP-Age, DynHCP-Gender, and DynHCP-Age under this category. The first two are static graph datasets while the last two are the corresponding dynamic graph datasets.

Predicting Mental States: The mental state decoding involves seven tasks: Emotion Processing, Gambling, Language, Motor, Relational Processing, Social Cognition, and Working Memory. Each task is designed to help delineate a core set of functions relevant to different facets of the relation between human brain, cognition and behavior [3]. Under this category, we present two datasets: HCP-Activity, a static representation, and DynHCP-Activity, its dynamic counterpart.

Estimating Cognitive Traits: The cognitive traits category of our dataset comprises two significant traits: working memory (List Sorting) [46] and fluid intelligence evaluation with PMAT24 [35]. Working memory refers to an individual's capacity to temporarily hold and manipulate information, a crucial aspect that influences higher cognitive functions such as reasoning, comprehension, and learning [37]. Fluid intelligence represents the ability to solve novel problems, independent of any knowledge from the past. It demonstrates the capacity to analyze complex relationships, identify patterns, and derive solutions in dynamic situations [14, 11]. The prediction of both these traits, quantified as continuous variables in our dataset, are treated as regression problem. We aim to predict the performance or scores related to these cognitive traits based on the functional connectome graphs. We generate four datasets under cognitive traits: HCP Fluid Intelligence (HCP-FI), HCP Working Memory (HCP-WM), DynHCP-FI and DynHCP-WM.

3.4 Learning models

The functional connectome, which effectively captures the network structure of brain activity, has proven to be a valuable representation of fMRI data for machine learning, as demonstrated in numerous previous studies and our own experiments [13, 1]. Recognizing its significance in the learning process, we sought a suitable GNN framework that could effectively leverage the comprehensive functional connectome data through a combination of message passing and neural network. After thorough exploration, we implemented a GNN architecture, denoted as GNN* illustrated in Figure 2 (b), that incorporates residual connections and concatenates hidden representations obtained from message passing at each layer. To further enhance the model's performance, we employ batch normalization and a multi-layer perceptron (MLP) to effectively utilize the combined representations during training. While adaptive residual connections have been extensively explored in GNNs, we present this simple and unique architecture for brain graphs that effectively learns the representations for brain graphs [33].

Recently, a number of dynamic graph representation approaches in conjunction with recurrent neural networks (RNNs) such as GRU, LSTM, and transformers, have been introduced [41, 23]. However, assessing the effectiveness of GNN models in a unified dynamic setting using the existing approaches presents a significant challenge. Therefore, we implement a simple and generalized architecture tailored to process dynamic graphs for the graph classification problem, as illustrated in Figure 2 (a). Our architecture comprises two distinct modules. The first is a GNN-based learning module, responsible for deriving graph-level representations from each of the derived graph snapshot. Following this, a transformer module takes over, applying attention to the learned representations

Table 2: Results of the gender classification using three distinct node feature configurations across three settings, evaluated on 10 GNNs. The configurations include CORRELATIONS (CORR), BOLD signals, and a combination of CORR + BOLD, evaluated across 100ROIs, 400ROIs, and 1000ROIs. Avg. column indicates the average results across the row and numbers under ROIs indicates average results across each ROI. The blue notation highlights the overall best results for each GNN and red indicates average best performance across each ROI. Instances marked with OOM denote an out-of-memory error. Average best results are obtained through 1000 ROIs with sparser graphs.

Dataset		k-GNN	GCN	SAGE	UniMP	ResGCN	GIN	Cheb	GAT	SGC	General	Avg.
100ROIs	CORR	65.65	68.98	68.70	68.33	66.06	68.24	63.94	69.49	68.43	64.95	**67.30**
	BOLD	49.58	50.97	51.67	51.30	51.34	55.09	53.19	49.95	51.90	51.11	51.11
	CORR+BOLD	52.78	51.02	50.28	50.79	50.60	54.91	49.44	50.37	51.57	51.30	51.36
400ROIs	CORR	72.21	74.10	61.66	68.57	70.09	71.89	58.94	69.35	**75.99**	**73.09**	**69.56**
	BOLD	51.16	51.62	53.94	51.39	52.31	55.09	49.07	50.46	53.24	53.94	52.22
	CORR+BOLD	51.53	51.90	52.96	51.57	52.36	55.56	50.63	52.13	52.08	52.61	53.33
1000ROIs	CORR	**78.80**	**75.19**	**71.71**	**75.14**	**78.75**	**77.22**	**64.77**	**71.34**	73.75	OOM	**74.07**
	BOLD	48.15	46.99	49.31	50.93	47.92	56.48	47.22	50.93	49.31	51.62	49.89
	CORR+BOLD	51.30	51.81	51.25	51.11	49.86	54.35	49.66	51.22	51.34	51.37	51.33

from the GNNs. Finally, the outputs are averaged into a single dynamic graph representation vector, $\boldsymbol{h}_{dyn}$. This design offers a universally applicable method for evaluating multiple GNN methods within a dynamic graph setting for the downstream ML classification and regression problem.

4 Benchmarking Setup

In order to thoroughly evaluate the performance of brain graphs generated through different hyper-parameters, we propose a series of research questions. These questions seek to identify the optimal setting for our graph-based neuroimaging analysis and ultimately enhance the performance of the predictive models derived from it.

Research Question 4.1 *What are the optimal node feature configurations?*

The first question aims to identify the best configurations for node features. This involves an exploration and comparison of various feature representations to discern their effectiveness on the performance of the derived predictive models. In assessing node feature configurations, our analysis encompasses the correlation matrix, the time-series BOLD signals, as well as their combination. The correlation matrix is generated by calculating the correlation values amongst all ROIs. On the other hand, the BOLD signals are derived post the preprocessing of the input fMRI image, adhering to the preprocessing pipeline outlined in Section 3.1.

Research Question 4.2 *To what extent does the number of ROIs impact the performance of predictive modeling on graphs?*

The second question delves into the influence of varying the number of ROIs on the performance of predictive modeling. The objective is to assess how the granularity of ROIs affects the quality and the performance of the predictive models. We evaluate $100, 400$ and 1000 number of ROIs.

Research Question 4.3 *To what degree does sparsifying brain functional connectome graphs impact the performance of predictive modeling? What threshold yields optimal performance?*

Our third question investigates the impact of sparsifying brain functional connectome graphs on the performance of the predictive models. It aims to establish an optimal threshold that leads to optimal model performance in graph machine learning setting. In our exploration, we consider the top 20%, 10%, and 5% percentile values from the correlation matrices to construct the graph edges.

Research Question 4.4 *Which graph convolution approaches are preferable for the predictive modeling on brain graphs?*

Our fourth and final question delves into the exploration of various graph convolution methods, assessing their suitability for predictive modeling on brain graphs. The aim here is not only to identify, but also to recommend the most effective techniques, considering the specific features and intricacies of neuroimaging data. In this endeavor, we have put to test over 12 GNNs, which include two of our own implemented frameworks, to gauge their comparative performance.

Table 3: Performance comparison in terms of accuracy of 10 GNNs with different ROIs and varying graph densities on gender and activity classification problems. The blue highlights the overall best results for each GNN. Instances marked with OOM denote an out-of-memory error.

Dataset			$k-$GNN	GCN	SAGE	UniMP	ResGCN	GIN	Cheb	GAT	SGC	General
Gender Classification	100ROIs	Sparse	63.33	72.96	69.35	69.72	68.06	69.72	63.70	70.28	70.37	67.22
		Medium	65.65	68.98	68.70	68.33	66.06	68.24	63.94	69.49	68.43	64.95
		Dense	64.44	68.52	65.00	68.06	63.70	66.39	64.26	69.72	68.43	61.76
	400ROIs	Sparse	69.95	**77.14**	69.86	67.56	71.43	69.4	**66.45**	72.72	**78.25**	76.13
		Medium	65.65	68.98	68.70	68.33	66.06	68.24	63.94	69.49	68.43	64.95
		Dense	71.61	76.13	62.58	61.20	69.77	73.27	61.84	67.83	74.19	72.44
	1000ROIs	Sparse	**82.13**	75.46	**77.69**	**76.67**	78.33	75.56	59.07	**76.2**	76.48	**78.89**
		Medium	78.80	75.19	71.71	75.14	78.75	77.22	64.77	71.34	73.75	OOM
		Dense	OOM	73.80	OOM	72.50	**78.89**	**78.70**	OOM	71.67	OOM	OOM
Activity Classification	100ROIs	Sparse	91.50	91.56	91.43	92.73	92.14	88.31	92.55	92.91	91.40	91.52
		Medium	90.91	90.80	91.81	92.75	92.25	88.01	93.06	93.15	91.40	91.22
		Dense	90.30	91.15	93.15	93.28	93.02	87.12	93.18	93.08	90.49	89.47
	400ROIs	Sparse	93.23	**94.21**	94.78	94.72	94.61	**89.79**	94.45	95.2	**94.17**	93.62
		Medium	92.26	93.93	93.89	95.02	94.33	89.44	79.03	94.67	93.39	93.58
		Dense	90.64	93.36	**95.76**	94.48	**94.64**	88.22	47.24	94.78	93.18	90.84
	1000ROIs	Sparse	93.50	93.80	94.09	93.59	94.23	85.14	93.82	94.66	93.2	**94.17**
		Medium	92.65	90.87	94.39	**95.79**	92.04	85.40	**94.88**	94.00	91.37	91.87
		Dense	**93.77**	93.12	94.12	94.54	93.59	81.59	92.92	**95.35**	93.76	93.76

By addressing these questions, we aim to set a robust benchmarking framework for graph-based machine learning methods in neuroimaging, providing invaluable insights into their optimal application.

5 Benchmarking Results

In this section, we introduce the baseline models, describe our experimental setup, and present the results from our preliminary exploration study. Following this, we lay out our approach to benchmarking and showcase the performance of various baseline methods on each dataset.

5.1 Baselines and Experimental Setup

This section outlines the specifics of our unique, generalized experimental setup designed to evaluate a range of GNN models. We consider 10 well-established GNN models: $k-$GNN [36], GCN [26], GraphSAGE [20], Unified Message Passing denoted as UniMP [43], Residual GCN (ResGCN) [6], Graph Isomorphism Network (GIN), Chebyshev Convolution [12], Graph Attention Network (GAT) [49], Simplified GCN (SGC) [50], and General Convolution (General) [52][1]. We also consider 3-layered Neural Network (NN), two dimensional Convolutional Neural Network (CNN) and Random Forest for the comparison.

In our experimental setup, we devise a graph classification architecture comprising three layers of GNNs, followed by a sort pooling aggregator [54]. Sort pooling sorts the node features based on the last channel, selecting only the first k representations. Subsequently, sort pooling is advanced through two one-dimensional convolution layers, which are then succeeded by a two-layer Multi-Layer Perceptron (MLP). This architecture has been consistently utilized across all GNNs throughout the entire experimental setup. For the dynamic datasets, we utilize our baseline method with five different GNNs. For NN, we utilized 512, 256, and 128 hidden units in each layer, respectively. For the CNN, we utilized a four-layer model with a stride of 2, 64 kernels of size 5, and padding set to 2. This was complemented by three fully connected layers [28]. For the Random Forest (RF) [5], we opted for 100 estimators, leaving the remaining parameters at their Scikit-learn defaults. All of our experiments were carried out 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.

Models training: We have carefully carried out the training and evaluation of each dataset in our study. Each dataset was partitioned randomly with 70% training, 20% testing, and 10% for validation. To ensure reproducibility and balance across the datasets, we employed a fixed seed, 123, for the split in a stratified setting. This stratified approach facilitated an equitable distribution of classes in each partition. Each model underwent training for 100 epochs with a learning rate of $1e^{-5}$ for classification, and for 50 epochs with a learning rate of $1e^{-3}$ for regression problem. Across all experiments, we set dropout to 0.5, weight decay to $5e^{-4}$, and designated 64 hidden dimensions for both the GNN convolution and MLP layers. Furthermore, for loss functions, we utilized cross entropy for classification and mean absolute error for regression problems.

[1]We use PyG implementations and default settings for running all these models.

5.2 Exploratory Experiments and Results

Here we address the research questions outlined earlier by conducting a series of experiments including the evaluation of different node feature configurations, the influence of varying numbers of ROIs, the implications of sparsity in brain graphs, and the effectiveness of diverse graph convolution approaches. Each experiment aligns with a research question, thereby paving the way for comprehensive analysis and definitive conclusions.

Performance enhancement with correlations as node features: Our first step involves evaluating the interplay between the number of ROIs and the configuration of node features, with an aim to streamline the overall search space. For this purpose, we engage in the gender classification problem using 10 different GNNs. The results of these experiments are presented in Table 2. It is clear that employing correlations as node features consistently enhances the performance across all evaluated numbers of ROIs. However, what caught our attention was the significant variance in the results obtained through correlations and BOLD signals and the number of ROIs. The performance notably declines when correlations and BOLD signals are combined, and the number of ROIs are reduced. This motivates further investigation on how to leverage BOLD signal or perhaps obtain features from the BOLD signals to be used for learning. Furthermore, the performance of different GNNs baselines does not consistently correlate with the number of ROIs or node features.

Performance enhancement through large ROIs and sparse brain graphs: Our analysis extended to evaluating the efficacy of 10 GNNs on gender classification, using a varying number of ROIs and different graph densities. In addition to gender classification, we further incorporated an activity classification problem to strengthen our observations under different settings. For all the experiments, we opted for correlations as node features, a decision driven by the consistent boost they offer in performance from the last experiment. The results are presented in Table 3. An important observation from our findings reveals that larger numbers of ROIs, (1000) demonstrate superior performance in gender classification. Similarly, a significant number of GNNs exhibit improved results with the use of 1000 ROIs for the activity classification problem. An analysis of the graph densities reveals an intriguing trend. We found that most GNNs achieved superior results when deployed on sparse graphs. Therefore, we deduce that the combination of large ROIs, sparse graphs, and correlation features contribute significantly to enhancing the performance of GNNs.

Table 4: Classification results in terms of accuracy on benchmark static datasets constructed with optimal setting. Blue indicates overall best results for each dataset.

Dataset	NN	CNN	RF	$k-$GNN	GCN	SAGE	UniMP	ResGCN	GIN	Cheb	GAT	SGC	General	GNN*
HCP-Activity	97.78	95.88	88.98	93.23	94.21	94.78	94.72	94.61	89.79	94.45	95.2	94.17	93.62	**98.20**
HCP-Gender	86.67	76.39	69.9	82.13	75.46	77.69	76.67	78.33	75.56	59.07	76.20	76.48	78.89	**89.07**
HCP-Age	44.23	43.38	40.84	42.72	43.66	40.94	43.85	40.00	44.98	41.97	42.25	43.47	41.03	**50.23**

Table 5: Results for HCP-FI and HCP-WM dataset using mean absolute error (MAE). Blue indicates overall best results for each dataset.

Dataset	k-GNN	GCN	SAGE	UniMP	ResGCN	GIN	Cheb	GAT	SGC	General	GNN*
HCP-FI	0.284	0.288	0.283	0.287	0.281	6.548	0.278	0.290	0.282	0.283	**0.264**
HCP-WM	0.818	0.825	0.810	0.812	0.830	1.032	0.789	0.804	0.828	0.819	**0.751**

5.3 Benchmarking with Optimal Settings

Considering the optimal setting obtained through exploring search space presented in the previous section, here we present the experimental setup and benchmarking results on the proposed 10 datasets.

The classification accuracy of all baseline models is detailed in Table 4. It is evident from the results that the GNN* stands out as the leading performer. However, the Neural Network's performance is also notably impressive. Similarly, the results pertaining to the regression problems have been outlined in Table 5. The leading performer on the regression problems is again GNN*.

In Table 6, we lay out the classification and regression results obtained on the dynamic datasets. Given the consideration of a basic dynamic baseline and the construction of dynamic datasets using limited dynamic lengths and number of ROIs, the performance does not quite match up to the static datasets. Nonetheless, it's worth noting that UniMP, despite the constraints, consistently demonstrates respectable performance.

Table 6: Models' performance in terms of accuracy and MAE across five dynamic datasets.

Dataset	Accuracy ↑					Dataset	MAE ↓				
	UniMP	$k-$GNN	GAT	SAGE	General		UniMP	$k-$GNN	GAT	SAGE	General
DynHCP-Activity	89.66	73.03	89.67	**90.93**	68.84	DynHCP-FI	**3.839**	3.841	3.861	3.842	3.862
DynHCP-Gender	**72.3**	68.45	67.13	66.20	62.04	DynHCP-WM	10.589	10.596	10.592	10.597	**10.571**
DynHCP-Age	**44.41**	44.25	44.39	40.65	42.99						

6 Conclusion

In this work, we introduce novel brain connectome benchmark datasets specifically tailored for graph machine learning, representing a promising avenue for addressing various challenges in neuroimaging. The inherent symmetries and complex higher-level patterns found in brain graphs make them well-suited for graph machine learning techniques. To advance this vision, we present NeuroGraph, a comprehensive suite encompassing benchmark datasets and computational tools.

In our comprehensive exploratory study encompassing 35 datasets, we conducted a thorough analysis by running multiple machine learning models. Our key observations are as follows: Firstly, utilizing correlation as node features shows promising potential for enhancing models' performance. Secondly, we observed that increasing the number of ROIs or employing large-scale brain graphs leads to improved performance compared to datasets with fewer ROIs. Thirdly, we demonstrated that employing sparser graph setting resulted in enhanced models' performance. Through a range of experiments across various learning objectives, we further highlight that GNNs exhibit superior performance compared to traditional NNs and 2D CNNs. These findings underscore the significant potential of GNNs in achieving improved performance across diverse tasks and underscore their suitability for graph-based Neuroimaging data analysis.

Based on these insightful observations, we have developed NeuroGraph, a meticulously curated and comprehensive benchmark dataset specifically designed for graph-based neuroimaging. Additionally, we provide computational tools to explore the design space of graph representation coming from Neuroimaging data, to facilitate the transformation of fMRI data into graph representations and showcase the potential of GNNs in this context. NeuroGraph serves as a valuable resource, offering a road map for researchers interested in leveraging graph-based approaches for fMRI analysis and demonstrating the effective utilization of GNNs in this domain.

Acknowledgement: "Data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University."

"This research incorporates contributions from the "Modeling and Model Integration" project of the Vanderbilt Institute for Software Integrated Systems. Work on "Modeling and Model Integration" is supported by Wellcome Leap as part of the Multi-Channel Psych Program. "

Benchmarks Availability and Licensing:

The fMRI data utilized in this research was sourced from the Human Connectome Project [47]. The proposed graph-based benchmark datasets can be accessed for download at `https://anwar-said.github.io/anwarsaid/neurograph.html` under the MIT license. These datasets are provided in PyG[2] 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 available at `https://neurograph.readthedocs.io/en/latest/`.

References

[1] Anees Abrol, Zening Fu, Mustafa Salman, Rogers Silva, Yuhui Du, Sergey Plis, and Vince Calhoun. Deep learning encodes robust discriminative neuroimaging representations to outperform standard machine learning. *Nature communications*, 12(1):353, 2021.

[2] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. Graph-based deep learning for medical diagnosis and analysis: past, present and future. *Sensors*, 21(14):4758, 2021.

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

[3] Deanna M Barch, Gregory C Burgess, Michael P Harms, Steven E Petersen, Bradley L Schlaggar, Maurizio Corbetta, Matthew F Glasser, Sandra Curtiss, Sachin Dixit, Cindy Feldt, et al. Function in the human connectome: task-fmri and individual differences in behavior. *Neuroimage*, 80:169–189, 2013.

[4] 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.

[5] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[6] Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

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

[8] Alexander Campbell, Antonio Giuliano Zippo, Luca Passamonti, Nicola Toschi, and Pietro Lio. Dbgsl: Dynamic brain graph structure learning. *arXiv preprint arXiv:2209.13513*, 2022.

[9] 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.

[10] Hejie Cui, Zijie Lu, Pan Li, and Carl Yang. On positional and structural node features for graph neural networks on non-attributed graphs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3898–3902, 2022.

[11] Simon Dahan, Logan ZJ Williams, Daniel Rueckert, and Emma C Robinson. Improving phenotype prediction using long-range spatio-temporal dynamics of functional connectivity. In *Machine Learning in Clinical Neuroimaging: 4th International Workshop, MLCN 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 4*, pages 145–154. Springer, 2021.

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[13] 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.

[14] Mónica Emch, Claudia C Von Bastian, and Kathrin Koch. Neural correlates of verbal working memory: An fmri meta-analysis. *Frontiers in Human Neuroscience*, 13:180, 2019.

[15] Taban Eslami, Vahid Mirjalili, Alvis Fong, Angela R Laird, and Fahad Saeed. Asd-diagnet: a hybrid learning approach for detection of autism spectrum disorder using fmri data. *Frontiers in neuroinformatics*, 13:70, 2019.

[16] Farzad V Farahani, Waldemar Karwowski, and Nichole R Lighthall. Application of graph theory for identifying connectivity patterns in human brain networks: a systematic review. *frontiers in Neuroscience*, 13:585, 2019.

[17] Michael D Fox and Marcus E Raichle. Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nature reviews neuroscience*, 8(9):700–711, 2007.

[18] 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.

[19] 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.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[21] Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. *arXiv preprint arXiv:2210.06681*, 2022.

[22] Jeremy Kawahara, Colin J Brown, Steven P Miller, Brian G Booth, Vann Chau, Ruth E Grunau, Jill G Zwicker, and Ghassan Hamarneh. Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146:1038–1049, 2017.

[23] Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *The Journal of Machine Learning Research*, 21(1):2648–2720, 2020.

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

[25] 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.

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[27] Lada Kohoutová, Juyeon Heo, Sungmin Cha, Sungwoo Lee, Taesup Moon, Tor D Wager, and Choong-Wan Woo. Toward a unified framework for interpreting machine-learning models in neuroimaging. *Nature protocols*, 15(4):1399–1435, 2020.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[29] Jun Li, Junyu Chen, Yucheng Tang, Ce Wang, Bennett A Landman, and S Kevin Zhou. Transforming medical imaging with transformers? a comparative review of key properties, current progresses, and future perspectives. *Medical image analysis*, page 102762, 2023.

[30] 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.

[31] 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.

[32] 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.

[33] Xiaorui Liu, Jiayuan Ding, Wei Jin, Han Xu, Yao Ma, Zitao Liu, and Jiliang Tang. Graph neural networks with adaptive residual. *Advances in Neural Information Processing Systems*, 34:9720–9733, 2021.

[34] 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.

[35] Tyler M Moore, Steven P Reise, Raquel E Gur, Hakon Hakonarson, and Ruben C Gur. Psychometric properties of the penn computerized neurocognitive battery. *Neuropsychology*, 29(2):235, 2015.

[36] 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.

[37] Guixia Pan, Li Xiao, Yuntong Bai, Tony W Wilson, Julia M Stephen, Vince D Calhoun, and Yu-Ping Wang. Multiview diffusion map improves prediction of fluid intelligence with two paradigms of fmri analysis. *IEEE Transactions on Biomedical Engineering*, 68(8):2529–2539, 2020.

[38] 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.

[39] Maria Giulia Preti, Thomas AW Bolton, and Dimitri Van De Ville. The dynamic functional connectome: State-of-the-art and perspectives. *Neuroimage*, 160:41–54, 2017.

[40] 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.

[41] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dynamic graph representation learning via self-attention networks. *arXiv preprint arXiv:1812.09430*, 2018.

[42] 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.

[43] 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.

[44] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: a structural description of the human brain. *PLoS computational biology*, 1(4):e42, 2005.

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

[46] David S Tulsky, Noelle Carlozzi, Nancy D Chiaravalloti, Jennifer L Beaumont, Pamela A Kisala, Dan Mungas, Kevin Conway, and Richard Gershon. Nih toolbox cognition battery (nihtb-cb): List sorting test to measure working memory. *Journal of the International Neuropsychological Society*, 20(6):599–610, 2014.

[47] 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.

[48] 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.

[49] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[50] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[51] Jiaxuan You, Tianyu Du, and Jure Leskovec. Roland: graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2358–2366, 2022.

[52] Jiaxuan You, Zhitao Ying, and Jure Leskovec. Design space for graph neural networks. *Advances in Neural Information Processing Systems*, 33:17009–17021, 2020.

[53] 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.

[54] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[55] Kanhao Zhao, Boris Duka, Hua Xie, Desmond J Oathes, Vince Calhoun, and Yu Zhang. A dynamic graph convolutional neural network framework reveals new insights into connectome dysfunctions in adhd. *NeuroImage*, 246:118774, 2022.

[56] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

A 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 [45]. 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.

A.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 [47]. 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[3] platforms, which host a diverse range of publicly available fMRI datasets contributed by researchers worldwide [38, 34]. These datasets cover various experimental paradigms, clinical populations, and research domains, providing researchers with a wealth of data for analysis and investigation.

We have chosen to utilize the HCP S1200 dataset from the Brain Connectome as a primary resource for our graph-based benchmarking [47]. 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.

A.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 7 for disk storage). To access the HCP data, users are required to obtain credentials from HCP[4] and provide them to NeuroGraph. Moreover, NeuroGraph also provides a Python class for preprocessing data from the local storage.

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

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

Table 7: fMRI scans required disk storage. The storage information is obtained from Human Connectome Project website.

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

A.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 six 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 [19].

- **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 [42], widely used brain parcellation schemes that define neurobiologically meaningful features of brain organization. These atlases provide a parcellation of the 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:** Next, we remove linear and quadratic trends. This step aims to remove the scanner drifts in the fMRI signals that arise from instrumental factors. By eliminating these trends, we enhance the signal-to-noise ratio and increase the sensitivity to neural activity.
- **Step 3 - Remove Motion Artifacts:** To further improve data quality, we apply regression techniques to mitigate the effects of motion artifacts. Specifically, we regress out six rigid-body head motion parameters, along with their derivatives, from the fMRI data. These parameters capture the movement and rotation of the subject's head during the scanning session, ensuring that any potential confounding effects are minimized.
- **Step 4 - Subject-Level Signal Normalization:** We perform subject-level normalization of the ROI timeseries signals. More specifically, we temporally normalize all signals from a subject 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 5 - 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 [13]. 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 [13, 24]. In Figure 3 and 4, we provide the visualizations of BOLD signals and their corresponding graphs for one subject in certain conditions.
- **Step 6 - 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

(a) Resting state

(b) Emotion task

(c) Gambling task

(d) Language task

Figure 3: Visualization of BOLD signals for a single subject during rest condition and doing tasks.

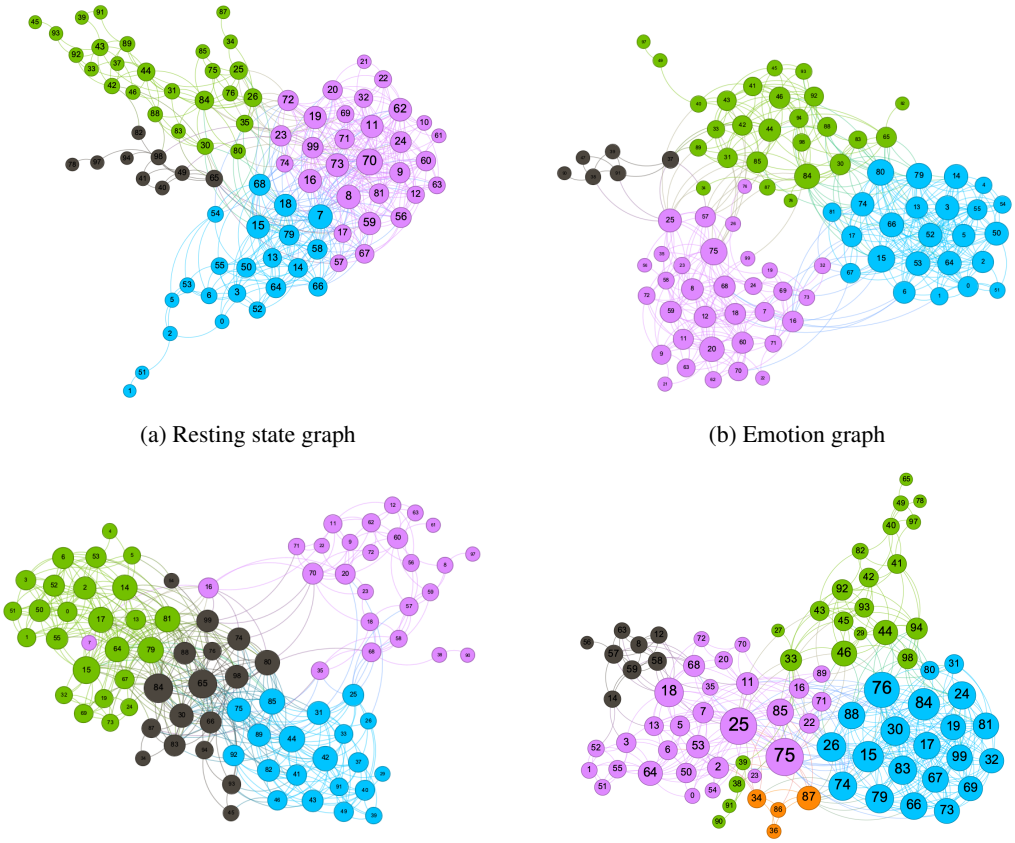
positive correlations between pairs of ROIs, as determined by a predefined threshold. The feature matrix is represented by $X^{n\times d}$, where n symbolizes the total number of ROIs, and d corresponds to the dimension of the feature vector. We explore the dataset generation search space by considering 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-Activity 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 $((l-\Gamma)/stride)+1$.

A.4 The Design Space is Huge

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 [25], weighted graphs[32], attributed graphs [9], and minimum spanning trees [4, 53], 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 [32, 40]. 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.



(a) Resting state graph

(b) Emotion graph

(c) Gambling graph

(d) Language graph

Figure 4: 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.

Node features: Traditional methods for representing node features in graphs include using coordinates [31], one-hot encoding [24], and mean activation [18, 31]. 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 [25, 32, 9]. 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.

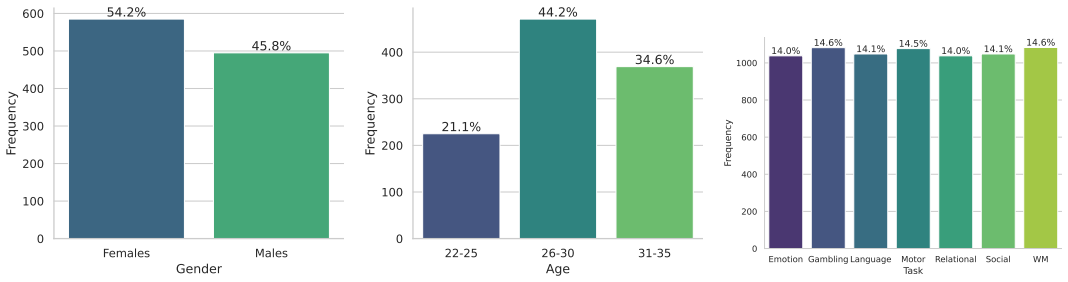
Density thresholding: Graph density is a fundamental property that may impacts the performance of GNNs. Graph density refers to the proportion of the possible connections in a graph that are actual connections. It influences how information is propagated through the network, may potentially affect the 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 high-density graph could lead to an information overflow, with a significant amount of information being propagated, possibly causing noise and overfitting [30].

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 [7]. 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 [48].

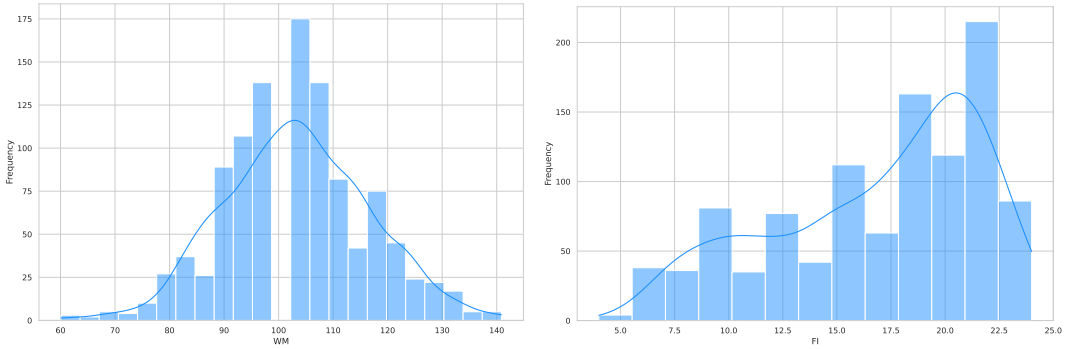
Indeed, there are numerous ways to conduct thresholding in brain graph construction, with several options available within each thresholding approach. Each method and option presents its unique set of advantages and potential limitations. In this context, we focus on proportional thresholding with positive correlations, an approach that has shown encouraging results in previous research [32, 25]. 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.

B 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-Activity, DynHCP-Activity, 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 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 5.



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



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

Figure 5: 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.

B.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.

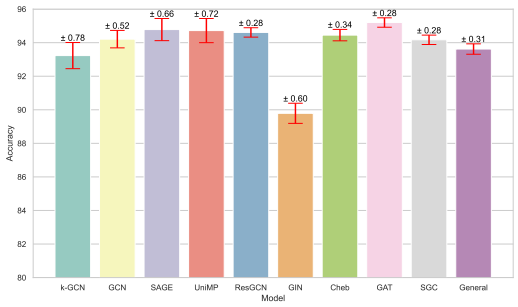
C Memory and Running Time Analysis

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

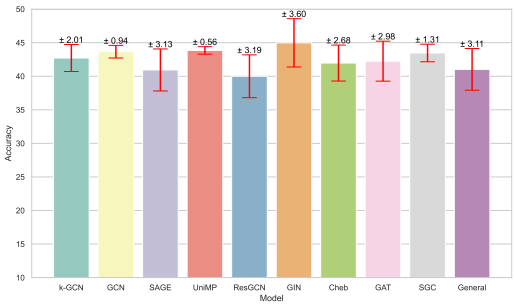
The specifics of this optimization are illustrated in the context of Unified Message Passing (UniMP) model [43], which we use to showcase the efficient resource usage of our datasets and approach. In Table 8, we offer detailed insights into the running times and memory requirements of UniMP model. We 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

Table 8: Resource utilization analysis of UniMP model on all benchmark datasets

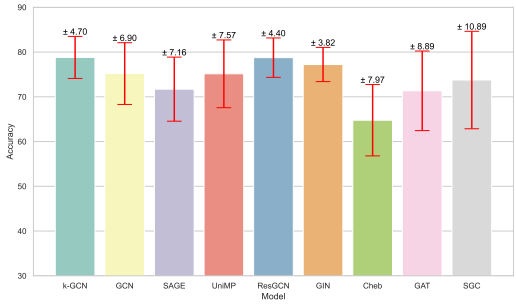
Dataset	Disk storage (GB)	#Parameters	Memory (MB)	Training time (sec)
HCP-Activity	4.0	265035	2463	854
HCP-Gender	3.7	648870	6437	362
HCP-Age	3.6	648903	4293	355
HCP-WM	3.7	803461	6551	696
HCP-FI	3.6	803461	6762	690
DynHCP-Activity	7.3	309575	15881	11200
DynHCP-Gender	1.1	308930	4169	1700
DynHCP-Age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



(a) HCP-Activity



(b) HCP-Age



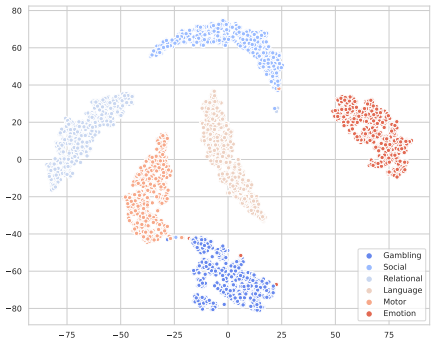
(c) HCP-Gender

Figure 6: Models' performance: Accuracy and standard deviation on 10 runs with different seeds on HCP-Activity, HCP-Age and HCP-Gender datasets.

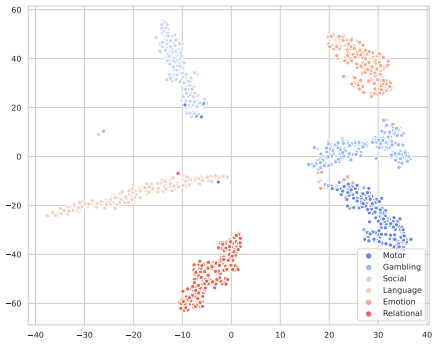
32 and 128 for the MLP layers. These data points provide a tangible representation of the efficiency gains achieved through our dataset size optimization process. Such optimizations are instrumental in ensuring datasets are not only computationally effective using any model but also highly accessible, 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.

D Models Performance and Standard Error

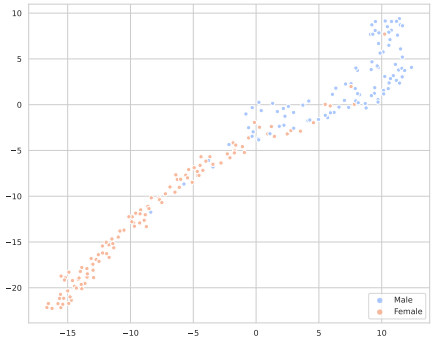
We plot the accuracy along with the standard deviation of 10 runs, each with different seeds, for all the models on three distinct datasets: HCP-Activity, HCP-Age and HCP-Gender in Figure 6. We observed that the results reported a higher level of stability on both HCP-Activity and HCP-Age datasets. This 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-Activity and HCP-Gender datasets in Figure 7. We used TSNE for these visualizations.



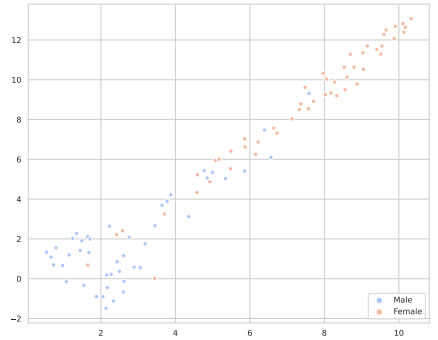
(a) HCP-Activity test set



(b) HCP-Activity validation set



(c) HCP-Gender test set



(d) HCP-Gender validation set

Figure 7: Hidden layer activation on test and validation sets of HCP-Activity and HCP-Gender.