Applications of Connectivity in the Human Brain Using Functional MRI data: Classification with Graph Neural Networks

Andrew Cheng
apcheng@ucsd.edu

Daphne Fabella
dfabella@ucsd.edu

Terho Koivisto
tkoivisto@ucsd.edu

Daniel Zhang
yiz029@ucsd.edu

Gabriel Riegner *
gariegner@ucsd.edu

Armin Schwartzman *
armins@health.ucsd.edu

Abstract

In our project, we implement a neural network-based application to discern gender difference from functional magnetic resonance imaging (fMRI) resting state data. An important region that activates while the brain is at rest is the default mode network (DMN). While we don't directly measure the activation of the DMN, it proves to be a key differentiator for the classification problem. Our research will walk through steps of statistical methods, simulation, and model description to achieve our task.

Code: https://github.com/AndrewCheng02/DSC180B-Capstone-ProjectA09

*Capstone mentor

1 Introduction

Functional magnetic resonance imaging still proves to be a key tool to deciphering the mysterious architecture of the brain after its first uses in the early nineties. Due to the high spatial resolution (in millimeters) and relative ease of getting a scan, data for fMRI scans has become highly available and useful. These scans still lack temporal resolution due to how fMRI blood-oxygen-level-dependent (BOLD) signals work, where it is the blood flow that is being measured rather than the signals in the brain. This will make our temporal resolution in seconds rather than some other scanning methods (e.g. MEG, EEG, NIRS) that are in milliseconds. Understanding and simulating fMRI data is crucial for understanding the connectivity of the brain.

Our project will focus on data from the Human Connectome Project (HCP) Young Adult data release, which has a collection of 1,200 fMRI scanned adult brains. The data is in the form of voxels, which are pixels that exist in a three-dimensional space of a brain mapping to an intensity of a BOLD signal at a specific point in time. With the amount of different brains, it would be impossible to map one to one comparisons between specific voxel locations in the brain. So we use pre-processed data that shrinks the number of subjects to 1,003 subjects. We will later discuss the data in more detail.

We are not the only ones to be interested in the differences of fMRI data in resting state between genders. A study by Sie et al. (2019) investigated age and gender differences in resting state functional connectivity. They found during rest females will have stronger correlations in the DMN than their male counterparts. While these differences were shown to wear off with age, this still proves there exist possible indicators to be found in fMRI data.

For our neural network based classifier, we must explain some important considerations. For data pre-processing we explain changes we might need to make to data. Simulate an environment to prove the classification is practicable from correlation data. Finally explain our exploratory data analysis to display significant differences in the data between genders.

2 Background

2.1 HCP Data

HCP is a research initiative that aims to map the structure and connections in the brain through fMRI. It started in 2009, and the project has become a joint effort involving various institutions in the United States. Our data comes from the HCP young adult study which released 1200 young adult subjects in 2018. The subject's data has been prepossessed to contain the denoised transformed network matrices and time series for each subject. The data also includes many features such as age and gender that will be useful in building our classification models. The data also includes the ICA decomposition atlas. (A choice of mapping for assigning the brain regions) Having the atlas allows us to train other fMRI scans outside of HCP on our model by transforming any Nifti file (fMRI 4D raw data) into a network matrix with the same preprocessing methods as the HCP data.

2.2 PCA and ICA

Raw fMRI data is a 4D matrix consisting of over 92000 voxels. We need a reduced version of the dataset to efficiently compute statistical methods and analyses and one of the ways we can group fMRI data is by decomposing the data into different components. This is called multivariate decomposition, where the data we observe comes from a number of separate sources that have been mixed together. We can then use factorization methods to identify these components and create regions of the brain that reduce the dimension of our data. The next two methods are commonly used to pinpoint areas of interest in the brain.

The first factorization method is Principal Component Analysis (PCA). PCA finds the orthogonal eigenvectors of the covariance matrix, otherwise known as the principal components. For example, given a set of points in 2D space, the first principal component is the direction along which the data has the most variance. The second principal component is the direction that explains the next greatest amount of variance and is uncorrelated with the first principal component. PCA has the benefit of being easy to understand and to implement. It can also be used for data reduction, where we only perform analysis on the principal components instead of thousands of individual data points (voxels) of the fMRI scan. However, as PCA is only sensitive to Gaussian data, it is not useful for many of the signals in fMRI data that do not follow a Gaussian distribution.

On the other hand, Independent Components Analysis (ICA) is able to find the components only if the data is not orthogonal (independent variables of the dataset are correlated) nor Gaussian. This factorization method was created to solve blind source separation problems, where signals from multiple sources have been mixed together, and the goal is to identify the different sources. The model for ICA is defined as $X = As$, where X is the data we observe, s is the set of components we are trying to identify, and A is the mixing matrix that combines the components. By assuming that the individual components are statistically independent, the resulting combination is likely to be non-Gaussian and able to be decomposed with ICA.

The in the algorithm for preprocessing the HCP data, we first apply PCA to reduce the overall dimensionality of the data. PCA performs better than ICA when reducing large datasets since it's computationally faster and simpler with out having a mixing matrix. In addition, a large dataset guarantees the data to become more Gaussian according to the Central Limit Theorem. After reducing the data through batches of PCA, the final step is reduced through ICA. The HCP database contains ICA decomposition for 15, 30, 100, 200, and 300 dimensions. For this report we will focus on using the 100 dimension data.

2.3 Correlation

The input for our model will be the pairwise correlation matrix generated by the fMRI time series also known as the network matrix. A common approach for estimating functional connectivity is finding the correlation values between two brain regions' activity over time. The fMRI scans used to generate the HCP data are pre-processed and analyzed to provide 4950 pairwise correlations for each subject. These pairwise correlations are calculated from the 100 regions found via PCA and ICA.

When analyzing a dataset encompassing more than two brain regions, a correlation matrix can be employed to explore the linear relationships among all possible pairs. We perform a series of analyses on these correlation matrices to explore the possible distinctions between the functional connectivity of the male brain and the female brain.

2.4 Partial Correlation

One major issue with pairwise correlations is that regions can be indirectly correlated with each other. For example, if X and Y are correlated to Z, then X and Y would have correlation with each other in the pairwise correlation matrix despite having no direct connections. To solve this problem we use Partial Correlations. A Partial Correlation matrix is generated by regressing the regions against every other region when before computing the correlation. In this case the instrument or confounding regions are accounted for which reduces the amount of false correlations.

2.5 Fisher Transform

The correlation matrix calculated from the fMRI scans undergo a Fisher Transformation. This Fisher transformation, which is achieved by taking the inverse hyperbolic tangent of the original Pearson correlation coefficients, amplifies pairwise correlations as they approach 1 or -1 and yields a new variable that has an approximately normal distribution.

The variable's normal distribution and amplification of highly correlated regions facilitates parametric testing in later sections that interesting patterns and statistically significant differences in BOLD signals between the male and female brain.

3 Methods

3.1 Data Generation

We decided to first test our classification methods on simulated data. To do this, we need to generate data with simple relations between variables that would clearly show on a correlation matrix.

We start by generating an $n \times m$ matrix A, where n is the number of time points and m is the number of regions in our simplified model of the brain. The greater the number of time points, the less noise will appear in the correlation matrix. Each cell in this matrix is filled with Gaussian noise. This simulates the different regions of the brain and their activity at each time point.

Box 1 *Simulate Brain Activity with Gaussian Noise*

$$A_{n,m} \overset{\text{iid}}{\sim} N\left(\mu, \sigma^2\right)$$

where

A = *generated dataset of brain activity*
μ = *mean of noise*
σ^2 = *variance of noise*

We want to simulate the connections between brain regions. When one region of the brain becomes active, there will also be activity in the other areas of the brain that are connected to it. This connection between the regions can be seen in the correlation matrix.

To mimic this, we first separate the matrix A in to two sets of rows, the active set B and inactive set C. This split is determined by a proportion p, ie. if $p = 0.2$, then B is the first 20 percent of rows and C is the remaining 80 percent.

We then define a $1 \times m$ vector of weights $\overrightarrow{v}$, which is preset beforehand, and define the matrix w as the diagonal matrix of the weights $\overrightarrow{v}$. To get the final result, we multiply the active set B with the matrix w then concatenate the product with the inactive set C.

Box 2 *Correlate Brain Regions with Weights*

$$\overrightarrow{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

$$w = Diag(\overrightarrow{v})$$

$$result = \begin{bmatrix} Bw \\ C \end{bmatrix}$$

where

$\overrightarrow{v}$ = *vector of weights*
w = *diagonal matrix of weights*
$result$ = *final processed generated data*

This process forces regions of the brain to be correlated with each other according to the vector of weights. For example, if we want regions a and b to be connected, we can set their weights to 2 while the other regions have a weight of 1. This means that for some time points, an increase in region a will correspond to an increase in region b, which will cause the two regions to be positively correlated.

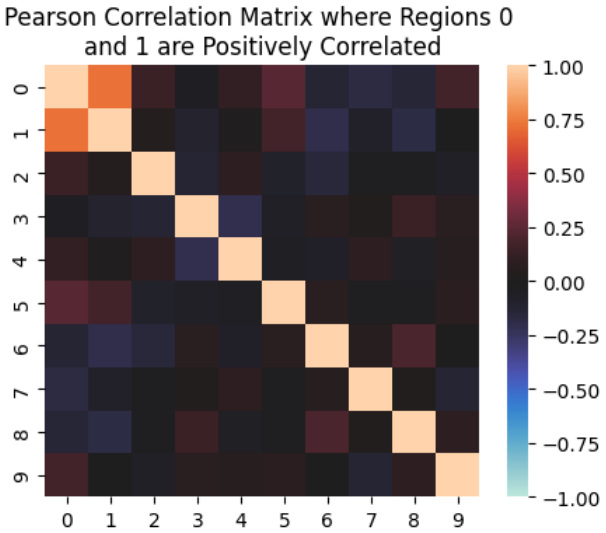

Figure 1: Pearson correlation matrix with weights:
$[2, 2, 1, 1, 1, 1, 1, 1, 1, 1]$

We now need to define the proportion of the initial split p, which determines the amount of active and inactive time points. The goal of this method is to force regions of the brain to correlate with each other, so a value of p that results in a large correlation coefficient is preferable. To find a suitable p, we calculate the average correlation of the resulting matrix for values in its range.

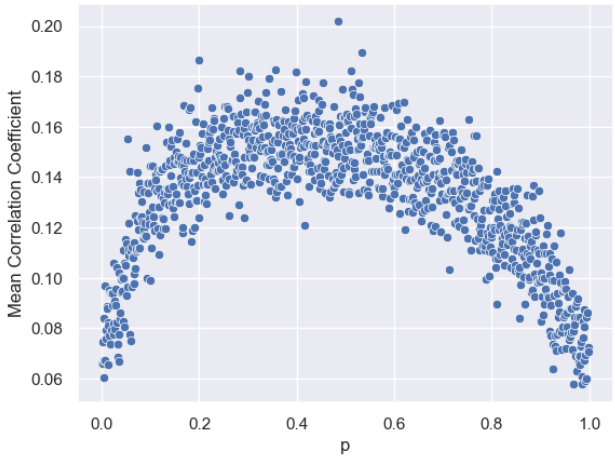

Figure 2: Proportion of the active set vs mean correlation coefficient. Each point is a simulated subject generated with the specified proportion.

From the plot above, we can see that as p increases, the correlation first increases and then decreases, being at its lowest at p values of 0 or 1. This makes intuitive sense, as this is when either all the time points are active or none are. We need both active and inactive sets to calculate the correlation between brain regions. If none of the time points are active, then the data is only Gaussian noise. If all of the time points are active, then the weights simply become inherent features of the brain regions. The value of p for when the correlation is most significant is around 0.4, which is what we ended up choosing for the proportion.

This process works well for generating Pearson Correlation matrices. However, Partial Correlations exclude other features when calculating the correlation between two regions. This means the more regions that are forced to be correlated with the weights in a single vector, the less visible the correlation will be on a Partial Correlation matrix.

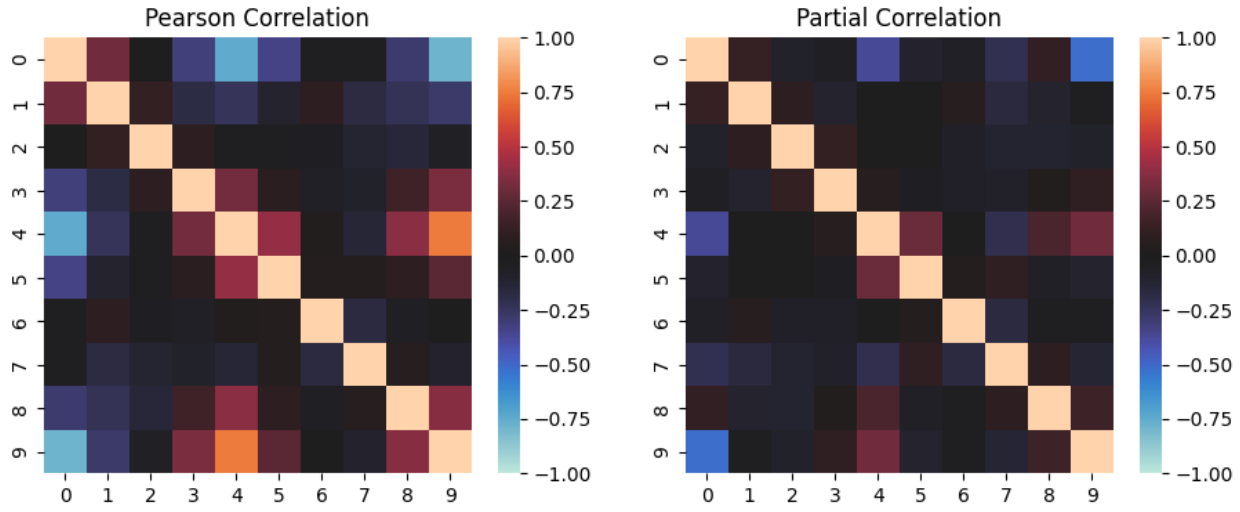

Figure 3: Pearson vs partial correlation with weights:
$[0.4, 0.8, 1, 1.2, 2, 1.2, 1, 1, 1.2, 2]$

To fix this, we repeat this process with multiple different weights, shuffling and recreating the active and inactive sets each time. This way, we can model more complex relationships than we could with a single vector of weights.

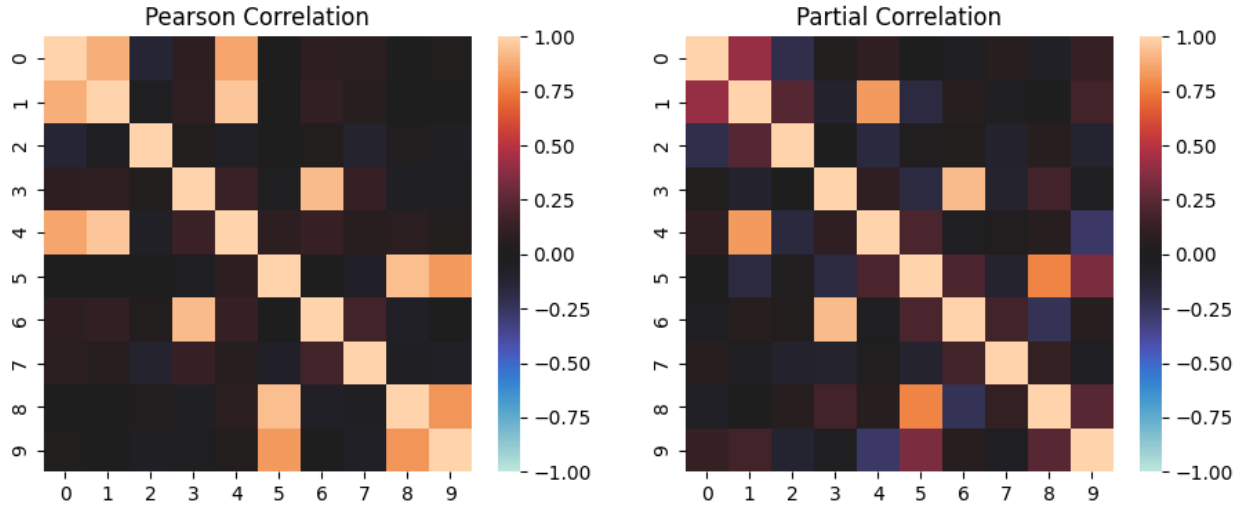

Figure 4: Pearson vs partial correlation with weights:
$[2, 10, 1, 1, 10, 1, 1, 1, 1, 1]$
$[1, 1, 1, 1, 1, 10, 1, 1, 10, 2]$
$[1, 1, 1, 10, 1, 1, 10, 1, 1, 1]$

We will be simulating a brain with 100 regions. The 100 weights for each brain are drawn

from a weight function, which is created using the product of a sinusoidal function and the normal distribution.

Box 3 *Generate weights from Sinusoidal Function*

$$W(x) = amp \cdot (f_1 sin(s_1 x) + f_2 \sin(s_2 e x) + f_3 \sin(s_3 \pi x)) \cdot (2\pi)^{\left(-\frac{1}{2}\right)} \cdot e^{-\frac{1}{2}\left(\frac{(x-b)}{w}\right)^2} + 1$$

where

$W(x)$ = *weight function to draw weights from*
amp = *amplitude scale parameter for the entire function*
f_i = *amplitude scale parameter for a sub-function*
s_i = *x scale parameter for a sub-function*
w = *width scale parameter of the normal distribution*
b = *translation parameter of the normal distribution*

The parameters of the weight function can be tweaked to adjust its amplitude and frequency, as well as the proportion contributed by each sub-function. The parameters we chose were amp: 0.1, f_1: 1.9, s_1: 3, f_2: 6, s_2: 5, f_3: 4.5, s_3: 1.3, and w: 7.

The parameter b translates the normal distribution along the x-axis and is used as the measure of separability between two weight functions. Functions with values for b that are close together would have low separability, while functions with values for b that are far from each other would be easy to separate.

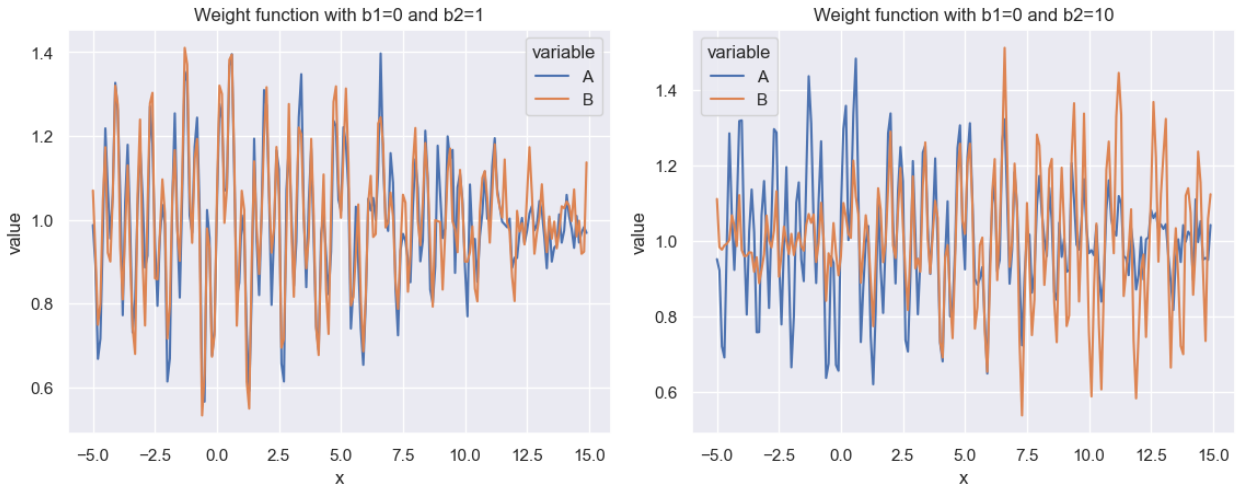

Figure 5: Comparison of weight functions at various levels of separability.

For this simulation, the b parameter of "male" brains will be situated at 0, while the b parameter of "female" brains will vary depending on the level of separability we are looking for.

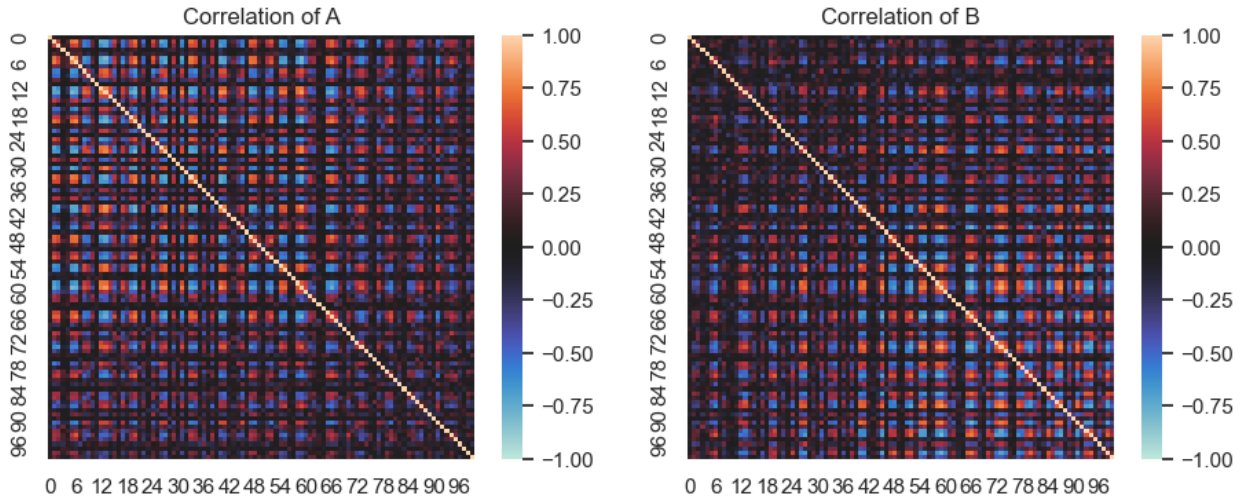

Figure 6: Pearson correlation of simulated "male" and "female" brains situated at $b = 0$ and $b = 10$ respectively.

In total, we generated correlation matrices for 11 levels of separability. Every level has 1000 subjects each with 100 time points and 100 brain regions. 50% of the subjects are "male" and 50% are "female". We can use KNN to check how separation accuracy increases with a greater difference in the b parameter.

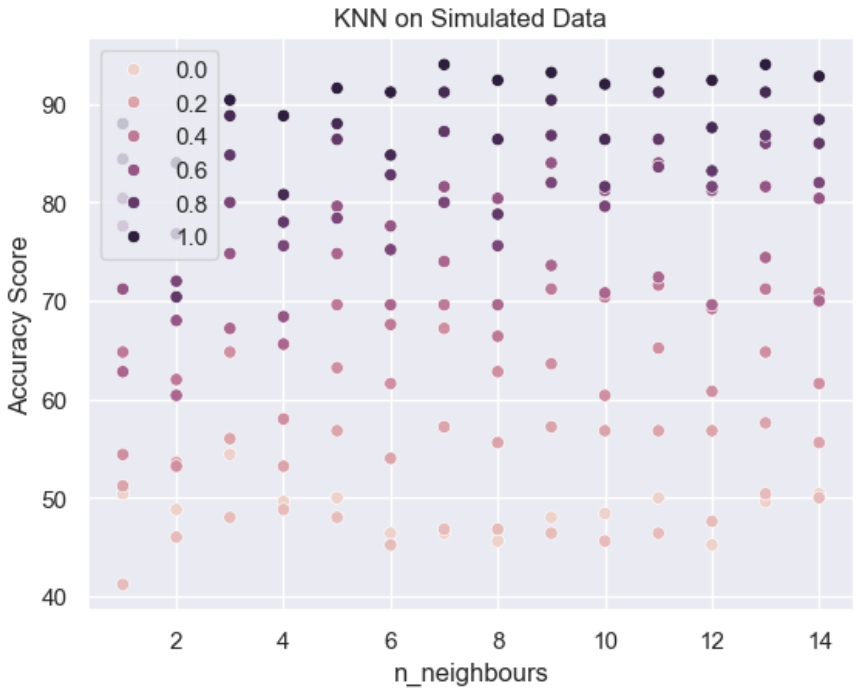

Figure 7: KNN accuracy at 11 levels of separability, indicated by color.

3.2 Exploratory Data Analysis

3.2.1 Independent Samples t -Test with Unequal Variances (Welch's t-Test)

In statistics, an independent samples t-test is a parametric test which is used to compare the populations of two variables. The t-test can examine two types of comparisons: whether two populations have equal means, a two-sided test, or if one of the population means is greater than or equal to the other, a one-sided test.

Here, we use the two-sided independent samples t-test to test whether two populations have equal mean pairwise correlations:

- **Null Hypothesis (H_0)**: Population 1 and Population 2 have the same mean.

$$\mu_1 - \mu_2 = 0$$

- **Alternative Hypothesis (H_1)**: Population 1 and Population 2 *do not* have the same mean.

$$\mu_1 - \mu_2 \neq 0$$

The t-test follows three assumptions: the populations are independent, each population's data is normally distributed, and the data within each population are independent.

We generated two datasets, x_1 and x_2, of sample size j under these assumptions by randomly sampling from two different normal distributions with unequal variances, each distribution described as follows:

Box 4 *Simulated Normal Distribution*

$$X_i \sim N\left(\mu_i, \sigma_i^2\right)$$

where

X_i = *dataset of random variable from population* i
μ_i = *mean of population* i
σ_i^2 = *variance of population* i

The resulting sample distributions had minimal overlap, and each with their own distinct centers:

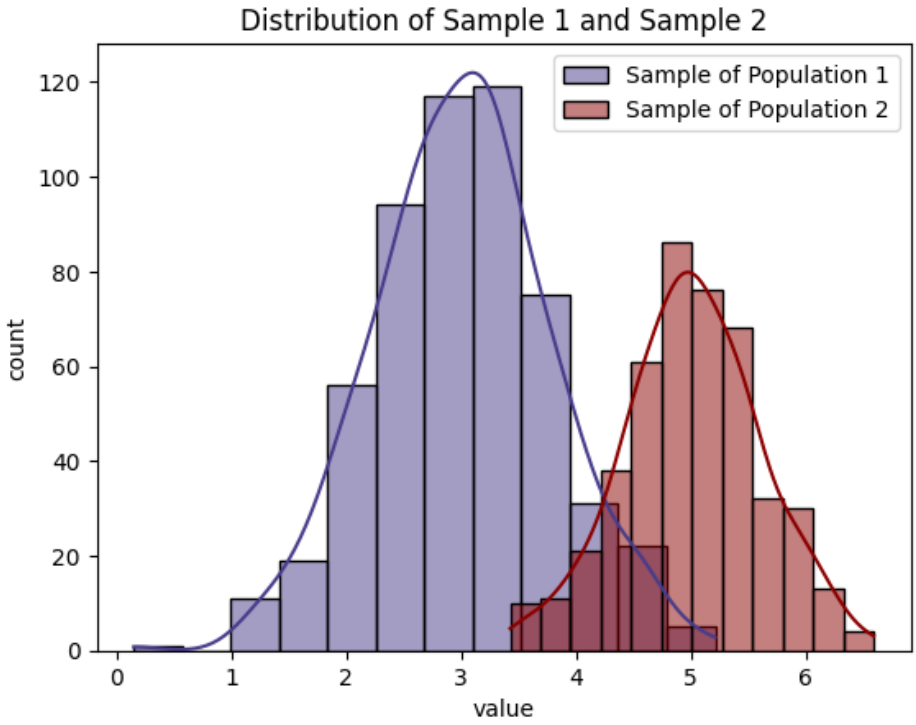

Figure 8: Distribution of Two Samples Generated From Distinct Populations with Different Centers

By looking at the distributions, we can see the that their centers are different. We can test the statistical significance of this observed difference by calculating the t-statistic. To calculate the t-statistic, we compare the sample means of each distribution (μ_i) and calculate the estimated overall standard error of the difference between the sample means ($\sigma_{\Delta\hat{\mu}}$), as follows:

Box 5 *t-Statistic*

$$t = \frac{\Delta\hat{\mu}}{\sigma_{\Delta\hat{\mu}}} = \frac{\hat{\mu_1} - \hat{\mu_2}}{\sqrt{\sigma^2_{\hat{\mu_1}} - \sigma^2_{\hat{\mu_2}}}}$$

$$\sigma_{\hat{\mu_i}} = \frac{\hat{\sigma}_i}{\sqrt{j_i}}$$

where

$\hat{\mu_1}, \hat{\mu_1}$ = *sample means of population* 1 *and population* 2
$\sigma_{\hat{\mu_i}}$ = *variance of sample means of population* i
$\hat{\sigma}_i$ = *sample variance of population* i
j_i = *sample size of population* i

Using the t-distribution, the t-statistic yields a p-value that, under a given significance level (α), allows us to reject or fail to reject the null hypothesis.

Here, our t-statistic (≈ -46.436) yielded a p-value of $0.0 < \alpha = 0.01$, which means we reject the null hypothesis that our two populations have the equal mean pairwise correlations.

t-tests allow us to compare the region-wise pairwise correlations in male and female BOLD signal activity, revealing interesting insights about the overall difference in the Default Mode Network for males and females.

3.2.2 Multiple Hypothesis t-Test

When comparing the male and female brain, we examine and compare the whole pairwise correlation matrix, rather than a single pairwise correlation, as we did in Section 3.2.1. In the HCP dataset, there exists 100 regions, yielding 4950 total pairwise correlations.

We will analyze and compare each of the 4950 pairwise correlations observed in the male brain with those in the female brain. This results in a total of 4950 t-tests. This statistical testing of many variables is referred to as the multiple hypothesis or multiple comparisons independent samples t-test. We can see the null hypothesis below:

- **Null Hypothesis (H_0)**: All of the population means are equal.

$$\mu_1 = \mu_2 = \mu_3 = \ldots = \mu_{4950}$$

- **Alternative Hypothesis (H_1)**: At least one of the population means is not equal.

Although similar to an independent samples t-test, a multiple comparisons test yields slightly different results due to the Familywise Error Rate, which we will demonstrate by simulating the HCP dataset.

We simulate the HCP dataset by generating $N = 1000$ random samples (one for each subject), each with a sample size of $j = 4950$ (each subject has 4950 pairwise correlations). 450 of these random samples are drawn from one normal distribution (to represent males) and the remaining 550 random samples are drawn from another normal distribution (to

represent females). Refer to Box 4 for a mathematical description of the normal distributions.

Below we can see the distribution of our simulated pairwise correlations corresponding to region 1:

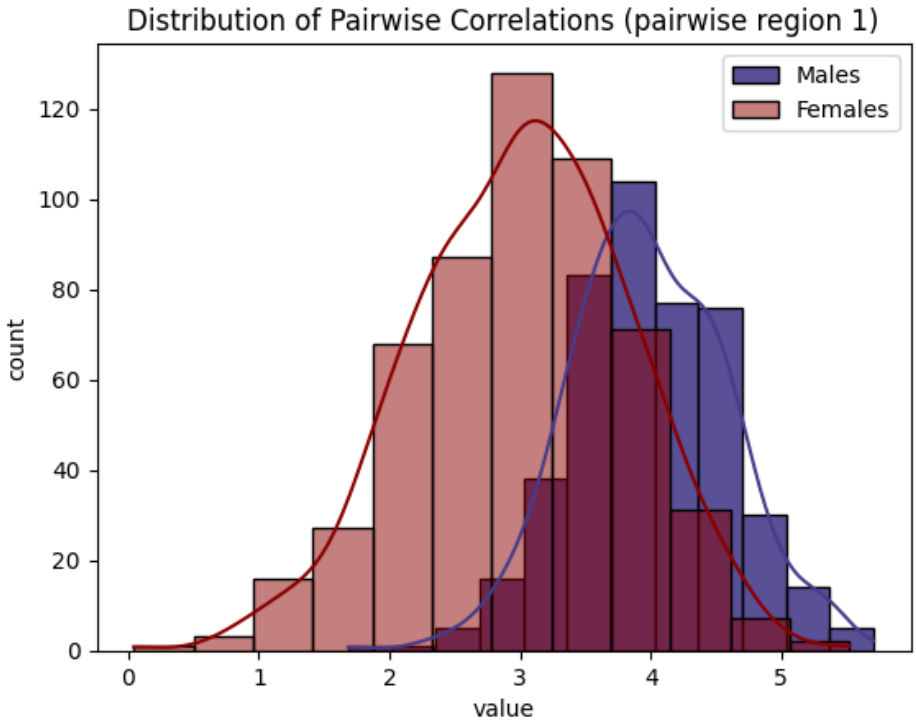

Figure 9: Distribution of Simulated Pairwise Correlations Generated From Distinct Normal Distributions

The simulated female and male distributions of pairwise correlations have visually distinct centers. To validate this observation, we find the t-statistic for each of our 4950 pairwise correlations using the formula for the t-statistic given in Box 5.

The resultant p-values for each of the 4950 pairwise correlations is captured below:

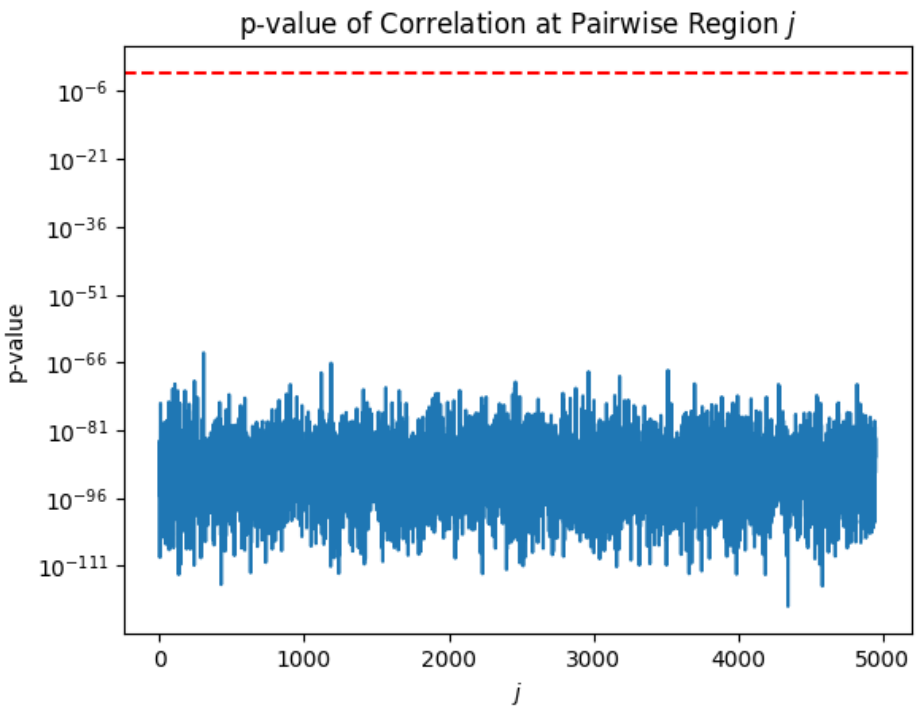

Figure 10: p-values of each Pairwise Correlation

As we can see, the t-test yielded a p-value $< \alpha = 0.01$ for all of the pairwise correlations, so we reject the null hypothesis. This is supported by the parameters of our population data, but when working with the real-life HCP dataset, each of the pairwise correlations' distributions may not be as distinct between nor consistent within the males and females.

We can repeat this simulation, but with two distributions that have the same population mean. This will help demonstrate how the multiple comparison independent samples t-test performs on distributions with less distinct means.

To generate two datasets with the same population mean, we randomly split the previously generated female samples into two parts.

As captured in the figure below, the two sets of 4950 pairwise correlations share the same population mean.

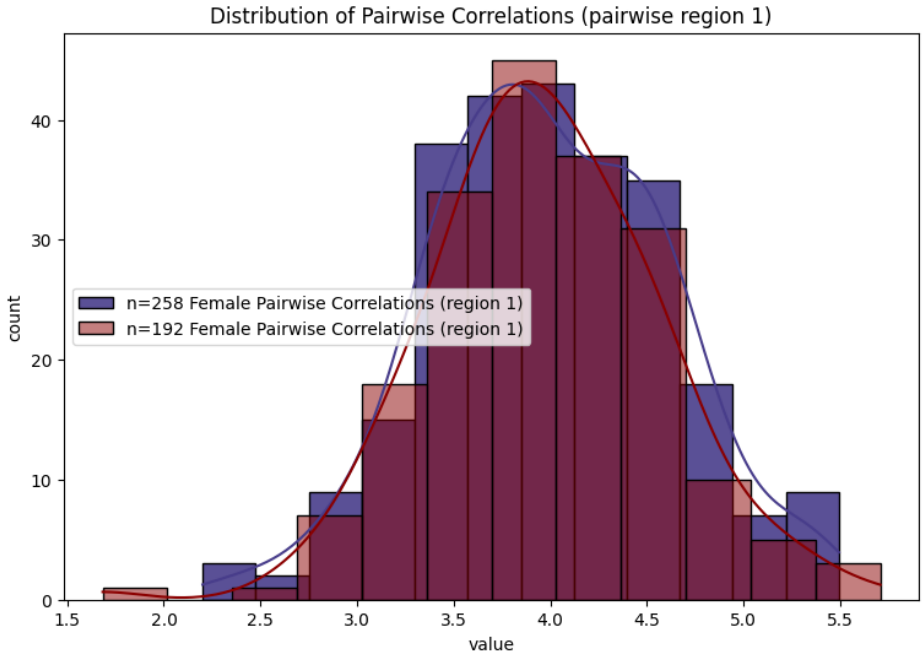

Figure 11: Distribution of Two Samples Generated From the Same Population

We calculated the t-statistics and their resultant p-values to observe a false positive rate approximately equal to $\alpha = 0.01$. When performing any multiple comparisons two-sided independent samples t-test on variables with unknown population means, we can expect to observe a similar false positive rate.

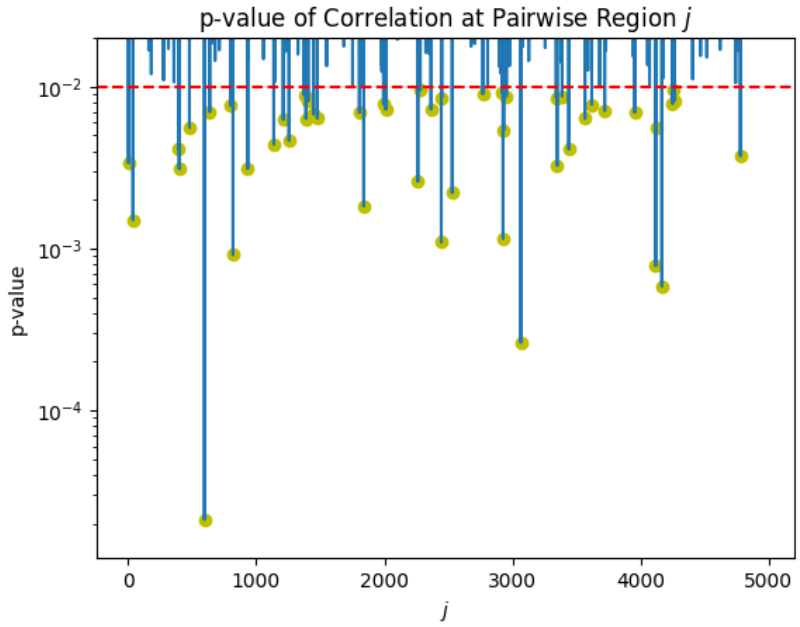

Figure 12: p-values of each Pairwise Correlation

For our exploratory data analysis, we are interested in conducting a more conservative statistical test to ensure that our pairwise correlations of interest are truly different between males and females.

3.2.3 Bonferroni-Corrected Significance Level

When considering multiple hypothesis testing, the probability of incorrectly rejecting the null hypothesis, also referred to as the false positive rate, is equal to the significance level (α) for *each* hypothesis. The *total* false positive rate, however, is also dependent on j, or the number of hypotheses:

Box 6 *Familywise Error Rate*

$$P_{FWER} = 1-(1-\alpha)^j$$

where
j = *the number of hypotheses*

This is recognized as the Familywise Error Rate ($FWER$), or the probability of at least one false positive. Using this formula, the probability of incorrectly rejecting the null hypothesis at least once ($FWER$) during our multiple comparisons test is approximately equal to 1; incorrectly labeling a pairwise correlation as having a statistically significant difference between males and females is almost guaranteed. We can reduce this Familywise Error rate by applying the Bonferroni correction, which divides α by the number of hypotheses, or in our case, pairwise correlations:

Box 7 *Bonferroni Corrected Significance Level*

$$\frac{\alpha}{j}$$

After plugging in the Bonferroni-corrected α into the Familywise Error Rate equation, assuming that j is large and α is small, our new total probability of incorrectly rejecting the null hypothesis is reduced from 1 to approximately equal to α:

Box 8 *Familywise Error Rate with Bonferroni-Correction*

$$P_{FWER} = 1-(1-\frac{\alpha}{j})^j \approx \alpha$$

We can apply this Bonferroni correction to the $\alpha = 0.01$ for our simulated-data multiple comparisons t-test, which yields the following p-values:

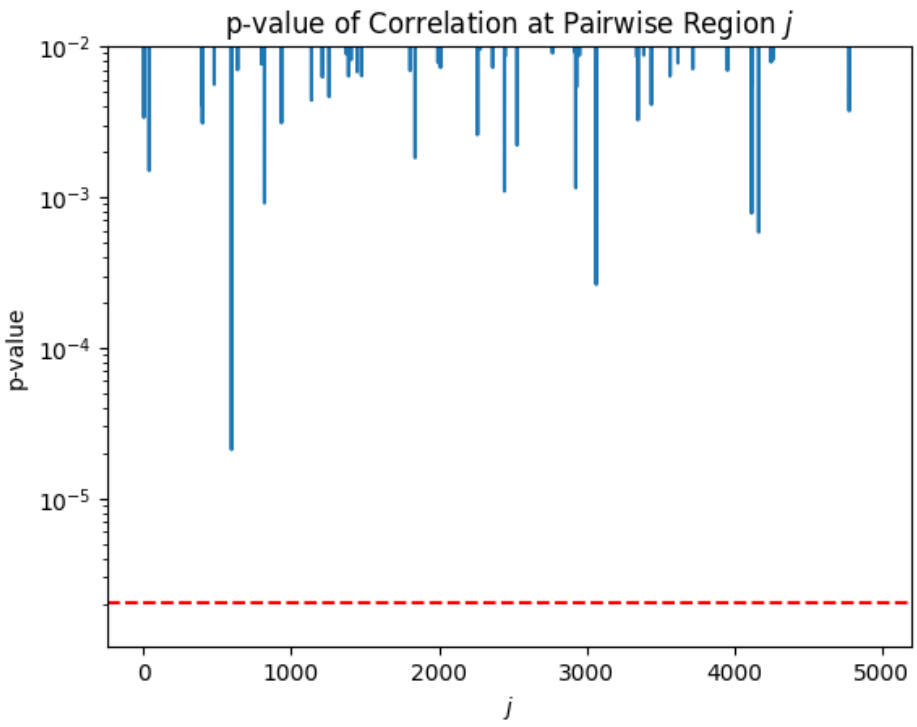

Figure 13: p-values of each Pairwise Correlation

As we can see, the observed false positive rate reduced from approximately 0.01 to 0 as a result of the new Familywise Error Rate ($P_{FWER} \approx (\alpha = 0.01)$).

Thus, when comparing the pairwise correlation matrices of males and females, applying the Bonferroni-correction on our significance level (α) helps yield a lower Familywise Error Rate. With a lower Familywise Error Rate, our statistical test will reveal a sparser matrix of statistically significant pairwise correlations, minimizing the number of features that will be used in our Graph Neural Network.

3.3 Modeling

3.3.1 KNN

This section will describe some of our findings and task related processes found on training the K nearest neighbors model or KNN. For the KNN we, vectorize the correlation matrix get it's position in the 100th dimension.

3.3.2 GNN

This section will describe some of our findings and task related processes found on training the GNN.

3.3.2.1 GNN input The input to our Graph Neural Network (GNN) consists of pre-processed ICA components provided by the previously discussed HCP fMRI data releases for 1,003 young adults. Each subject is associated with metadata including age range and gender. Gender information is extracted and used as a label for our classification task. From this data, meaningful edges are extracted through EDA significant edges, indicating important connections between components for distinguishing gender within the graph.

3.3.2.2 GNN implementation The GNN processes the graph data for each subject iteratively examining the relations of neighboring significantly connected components to a specific component in the graph. By running the network by subject, over time it will recognize important indicators in the structure found by edges as well as correlation weights on specific components. It summarizes these relationships and updates the state of each component per subject iteratively using GNN operations. The process is repeated until the network achieves satisfactory accuracy on both the training and validation sets. GNNs leverage parallel computing methods, making them suitable for efficient training on cloud computing platforms. Despite this, most of our training has run on memory of a personal computer.

3.3.2.3 GNN architecture From empirical findings of running with different hyperparameters, general rules of thumb for neural networks have been useful. This mainly includes having a tuned learning rate, using activation functions on each hidden layer, changing the shape of the data to a higher dimension then reducing the dimension to, and finally having a linear layer before output. Some changes from usual neural networks are using the tanh activation function since our data needs to be analyzed for positive and negative correlation values. Utilizing batchnorm on each hidden layer proved to be useful, which was inspired from a paper with a similar task. Article by Hough (2022) describes the usefulness of batchnorm which is most likely due to its benefits to accelerate training per epochs. Overall there is no single architecture that will work for any given task. As tested by using same architecture on simulated fMRI data vs. real fMRI data, the results vary.

3.3.2.4 Output of GNN Our GNN belongs to the class of graph classification GNNs, where the objective is to predict the class label of the input graph. In our case, the output is binary, indicating whether the subject is female (true) or not (false). We use a linear layer, mean across the layer, then a sigmoid to map the GNN's output to the binary classification task. The model's accuracy is evaluated on a test set that is held out from both training and validation. We also have access to a single UCSD's own fMRI resting state fMRI data which will be interesting to test whether our model is accurate for one of our group members fMRI data. To ensure reproducibility, we save the learned weights of the model. Future directions include exploring the application of the model to detect neurological diseases as data availability allows.

3.3.2.5 Findings Creating a GNN on the simulated fMRI data yielded almost 100 percent accuracy to differentiate if the subject was from a female or male group. Where as, using real data with similar architecture to the simulated one was only able to reach about 70 percent validation accuracy. Some changes are needed to achieve higher classification accuracy, but training the model on simulated data has proved useful to test assumptions about any kind of parameter changes we did. From our findings we believe as we manufacture the neural network more, we will find higher accuracy, due to there being significant differences in fMRI data, and simulated data performing extremely well on correlation data.

4 Appendix

The appendix will include our up-to-date project proposal for comparison. This also offers an alternative description of our plans to use graph neural networks.

4.1 Project Proposal

4.1.1 Problem Statement

For our second quarter project, we plan to highlight differences in female and male fMRI resting-state data. We plan to implement a neural network-based solution to detect the differences in the female and male brains. To detect the differences we look at the default mode network (DMN), due to many papers finding changes of the DMN in age(Sie et al. 2019), disease (Lee, Smyser and Shimony 2012), and gender (Kovalev, Priimenko and Ponomareva 2017). The DMN is interesting because it is highly active when a brain is comprehending something or thinking about the self. To see DMN in fMRI data, the patient is asked to be at rest and think about nothing.

Overall our project is engaged in the mysteries of our brain which cannot only be solved by only by statistical approaches. Neural networks are beneficial for finding relationships in the data that would not otherwise be easily discovered. Once we have a trained model, we are able to release the architecture and trained parameters of the neural network so our work will be easily reproducible. To achieve our task, we need to take consideration of findings from our previous quarter, and create a pipeline utilizing Graph Neural Networks (GNN) to easily classify gender from resting state fMRI data. Specifically, we want to know if a male or female in a resting state.

Another interesting factor about the subject of researching the DMN is that we can measure the connectivity of the network to show evidence of neurodegenerative diseases such as Alzheimer's (Fox et al. 2005). While our project doesn't focus on detecting neurological decay, it is interested in finding a method of detecting the connectivity of the brain, as well as exploring regions of the brain that might be active during a resting state. The DMN is not the only network, but it might be the most mysterious one since it is active when the brain is not focused on a specific task such as hearing or learning.

4.1.2 Previous Work

The following section will cover previous research into the differences of the female and male brains, related works that have research the usage of GNNs on fMRI data, and background on the Human Connectome Project (HCP) data.

4.1.2.1 Gender Difference in Resting State fMRI During our exploratory data analysis we found some differences in the averages of male and female ICA component data, that

included sign-vise location differences in the data. A similar study by Sie et al. (2019) on gender and age difference on functional connectivity found stronger correlations with the DMN than their male counter parts. They also found the correlation differences ware off in older subjects, which has unfavorable effects for the task.

On early tests with classifiers using nilearn tutorials, we found using small dataset on children and young adults the classifier was able to differentiate between age classes with about 90 percent accuracy. Where changing the classification to classify gender in younger subjects, it was only able to reach about 60 percent accuracy. This presents some possible road-bumps to make universally working classifier with a GNN, but the data we are working with will come from the HCP Healthy Young Adult data release. We will discuss more about the data later in this section.

4.1.2.2 Related Work We are not the first ones to propose the idea of using GNNs on resting state fMRI data. A paper by Wang, Wu and Hong (2022), outlines an approach of using GNNs with the GraphSage architecture. Our goal for the project is to explore other architectures and compare results to previously tested methods. The 2022 paper will serve as guidance and inspiration for our project of using deep learning algorithms for detecting networks in the brain.

Another tutorial posted on Medium by Hough (2022), published code to a GNN classifier for age from fMRI data. The task is to classify from movie watching fMRI data whether a child or an adult is watching. One component of their pipeline is rather interesting where batch norms, a batch summarizing technique in neural networks training worked well. Other tips that the author used will be commented in our code. The tutorial is a great starting point for us to train a model for a different task but a similar output process.

Overall, there is no single technique that will work for training a model successfully. Neural networks now more than ever, especially GNNs are being researched into every day, but one rule of thumb still applies. Every task will require a unique set of layers of architecture and pre-processing to fit the task. With the mentioned sources it will put our project to a great start, but much more exploration is needed.

4.1.2.3 HCP Data The data is provided by the Human Connectome Project on healthy young adults, which comprises 1003 subjects. The data sequences we are using include the resting state data, which we are interested in on training our model. The data release has published pre-processed ICA component correlation matrices for each subject, which vary in number of components. We chose to work with the ICA100 which means each subjects partial correlation matrix includes a hundred networks found by the ICA. These networks can overlap the regions in the brain between components, but are unique due to being generated from a timeseries, and offer complexity to leverage the neural network's classification task. Our ambitions for future subjects of research is the human disease data releases, as a similar pipeline can be followed to detect neurological diseases from fMRI resting state.

4.1.3 Deliverables

This section explains each step of our experimentation for finding a graph neural network that fits our task. We will walk through the steps of data processing, data representation needed for GNN's, simplified GNN internal functions, and the output of the network. We plan on producing all code with PyTorch, a Python package for machine learning algorithms.

4.1.3.1 Data Our data is comprised of pre-processed ICA components by the HCP, for 1003 young adults. The metadata includes some information about the subject, such as age range and gender. We extracted the gender information as a label for our classification task. From the data we will be able to extract meaningful edges via threshold or mean structure for components that are important in distinguishing gender from the graph.

4.1.3.2 Input to GNN We created a data collating class to handle the data through processes of training, validation, and later testing. This will include the unique or average thresholded edges that will tell the neural network there is a connection between components. We conduct a shuffled 80:10:10 split for training, validation, and testing. Now the data is ready for processing, next is to decide the architecture and output procedures.

4.1.3.3 GNN Implementation The GNN will look at the relations of neighboring connected location to a specific location in the graph, then summarize the relationships the location has, and update the current locations state to the next one with GNN operations. Run this for every location, and for as long as we want until the network reaches a satisfactory accuracy for the output in both training and validation. GNNs work great with parallel computing methods, so cloud computation will be efficient if we need more powerful machines to train our model on.

Our data does not have any node information than the name of the component. This means the nodes of our graph don't hold any information and it is the edges that hold data about the correlation between regions. This won't be a problem as GNNs are perfect for finding relationships from a network of connection values to extract structure in the connections. This will be useful for binary classification, where the network is able to find some higher dimensional indicator for male or female resting state fMRI.

Finally to get our predictions we need a linear layer to reduce our prediction to a true or false if the network predicts its a female subject that is at rest and their DMN active. Later on we will be able to change the structure of the network to attempt better detection with other related models such as, GAT (attention based), GCN (convolution based), and LSTM (gate based) graph neural networks. These three options all have potential to be great candidates for a successful detection tool.

4.1.3.4 GNN Output Our GNN is in the class of graph classification, given a graph, predict which class it belongs in. For our case we only have a true or false output, but it wouldn't be impossible to distinguish age differences in the subjects. In the end we will have accuracy of the model on the test set that has been completely hidden from the algorithm during its training and validation phase. To publish and reproduce our results we will save the weights learned, because each time a neural network is ran you will get a slightly different result. Our plan is to achieve high accuracy and possible extending the topic to neurological diseases as data allows.

5 Contributions

Daphne Fabella developed the report structure and was responsible for writing Background Section 2.5 Fisher Transform and Method Section 3.2 Exploratory Data Analysis along with their accompanying code.

Andrew Cheng was in charge of working with the HCP data. For the report, he wrote the all the background sections except for the Fisher Transform. For the code, Andrew implemented the HCP data which created a clean and usable version of the HCP data for training. I also wrote the Andrew Analysis and KNN notebooks. Andrew worked on the READ.ME and making sure the project was reproducible.

Daniel Zhang was responsible for generating simulated correlation matrices and for writing Methods Section 3.1 Data Generation and the corresponding notebook.

Terho Koivisto wrote the project proposal, abstract, introduction, and GNN modeling section under methods. He was responsible for the majority of the GNN development and research.

References

Fox, Michael D., Abraham Z. Snyder, Justin L. Vincent, Maurizio Corbetta, David C. Van Essen, and Marcus E. Raichle. 2005. "The human brain is intrinsically organized into dynamic, anticorrelated functional networks." *Proceedings of the National Academy of Sciences* 102(27): 9673–9678. [Link]

Hough, Sidney. 2022. "GNNS in neuroscience: Graph convolutional networks for fmri analysis." *Medium*. [Link]

Kovalev, Dmitry, Sergey Priimenko, and Natalya Ponomareva. 2017. "Search for gender difference in functional connectivity of resting state fMRI." In *ANALYTICS AND DATA MANAGEMENT IN DATA-INTENSIVE AREAS (translated)*. [Link]

Lee, M H, C D Smyser, and J S Shimony. 2012. "Resting-state fMRI: a review of methods and clinical applications." *AJNR Am J Neuroradiol* 34(10): 1866–1872. [Link]

Sie, Jia-Hong, Yin-Hua Chen, Yuo-Hsien Shiau, and Woei-Chyn Chu. 2019. "Gender- and Age-Specific Differences in Resting-State Functional Connectivity of the Central Autonomic Network in Adulthood." *Front Hum Neurosci* 13, p. 369. [Link]

Wang, Donglin, Qiang Wu, and Don Hong. 2022. "Extracting default mode network based on graph neural network for resting state fMRI study." *Frontiers in Neuroimaging* 1. [Link]

Acknowledgements

We want to extend our thanks to our mentor, Professor Armin Schwartzman, who graciously provided his extensive domain expertise in statistics, neuroscience, signal processing and machine learning throughout the project timeline.

We also want to express our gratitude to our PhD advisor, Gabriel Riegner, for his thoughtfulness and constant support.

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.