FMRI analysis using GNN

Members:

1. Manishkumar Alagumalai
2. Rishit Puri
3. Aathirai Senthilkumar Thamaraiselvi
4. Aravind Sivakumar
5. Aaditya Dharne

Overview

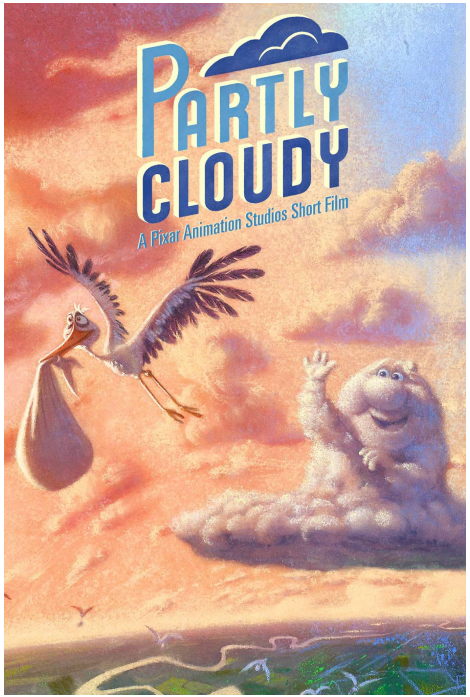
1. Project explanation.
2. Dataset overview.
3. Dataset preparation.
4. Models run:
    i. GCN
    ii. Edge Conv Layers
    iii. GAT
    iv. Graph Sage
    v. Graph Neural ODE
5. Result comparison

Project explanation

- The brain can be conceptualized as a complex network where each brain region is a node and connections between regions represent functional interactions.
- MRI data of 3–12 year old children and adults during viewing of a short animated film, Pixar “Partly Cloudy”.
- Binary prediction problem.

Goal:

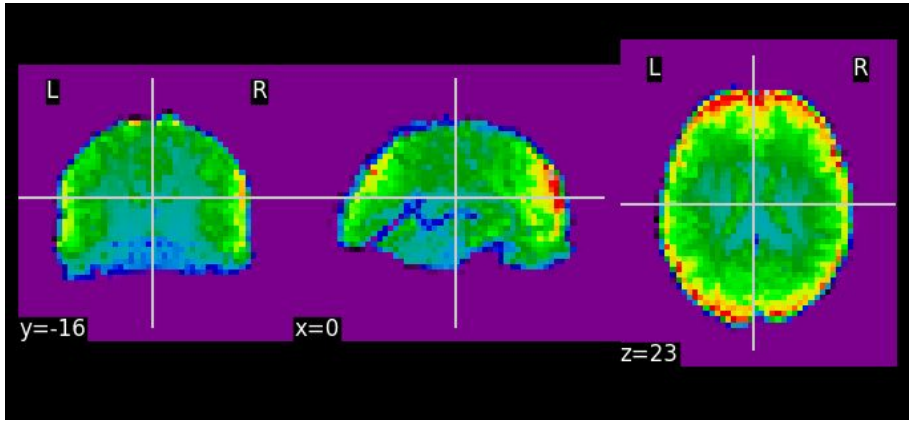
- To predict if the participant watching the animated film is adult or a child.



Dataset

- Dataset is taken from a open platform - OpenNeuro.
- An atlas that parcellates the brain into ROIs is used.
- This 155-subject dataset has already had a popular preprocessing pipeline run on it known as fMRIPrep.
- A confounds file is used to regress out noise. This confounds file, is generated by fMRIPrep.

Plots of images in the dataset



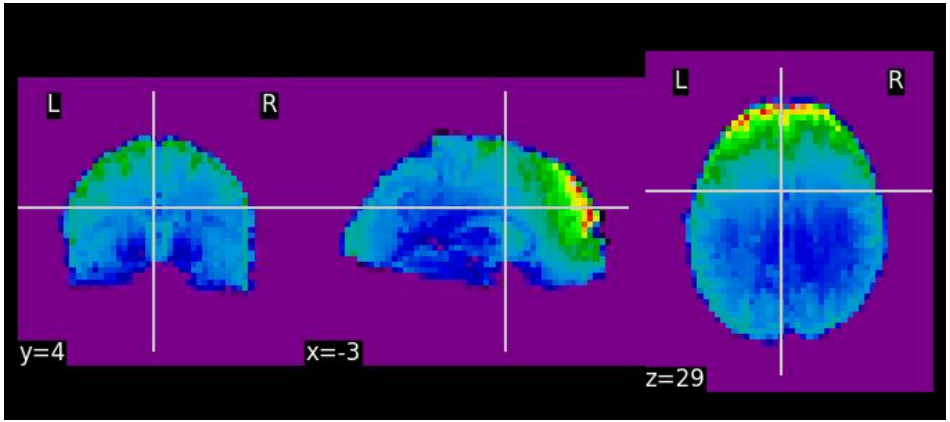
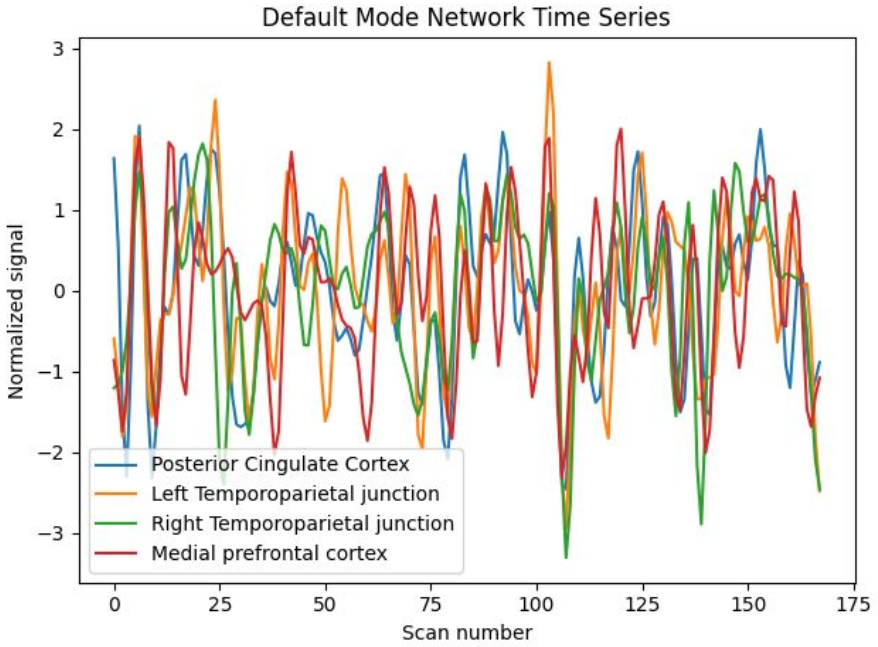
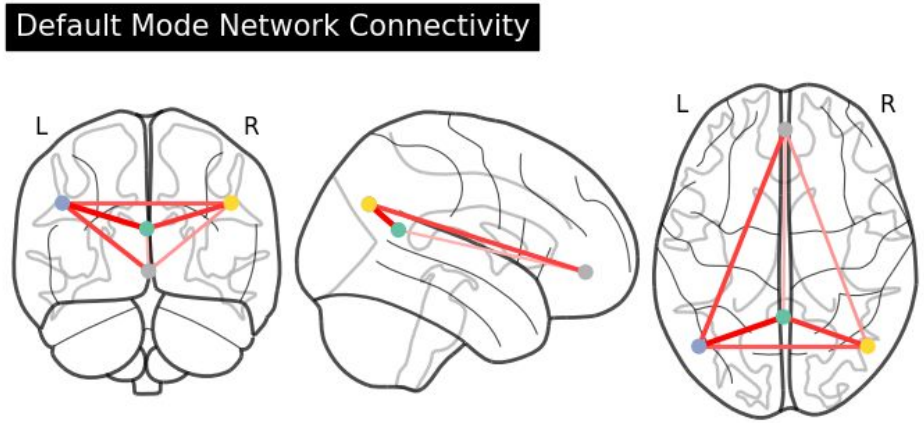


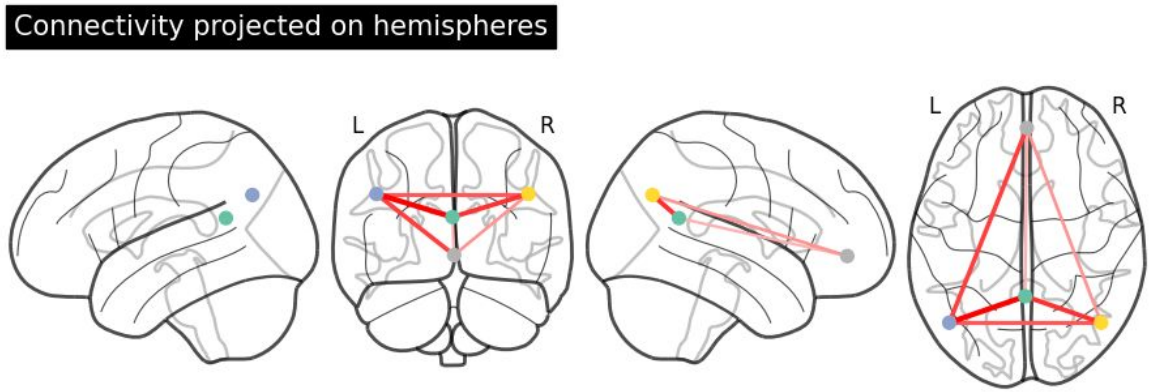
Image converted to Time series data

- BOLD measurements give us information about how active each ROI is within the brain over time.
- ROIs are clusters of brain cells, and our bodies send more oxygen-rich blood to these cells when they become active.
- These local changes in blood-oxygen levels are measured by the MRI machine

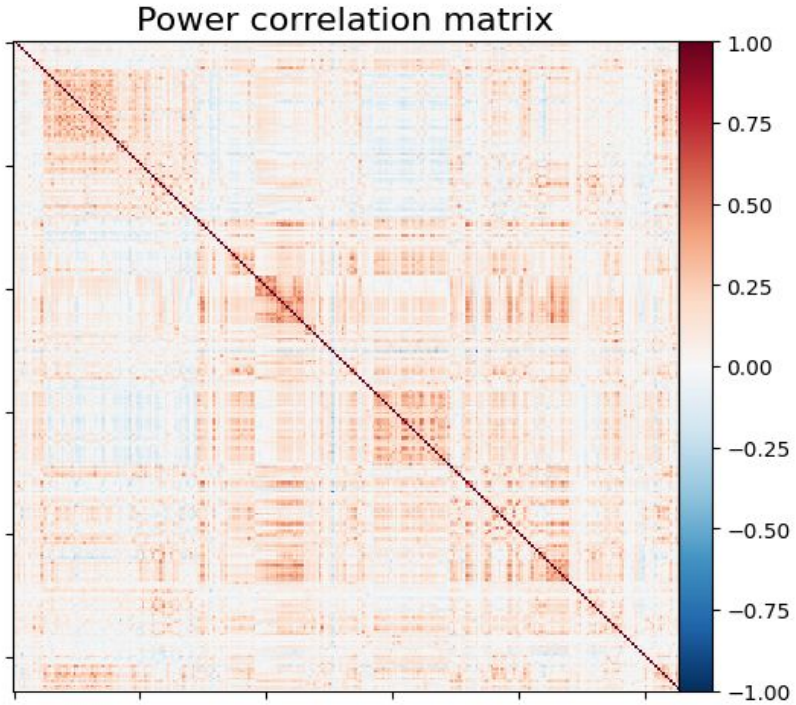


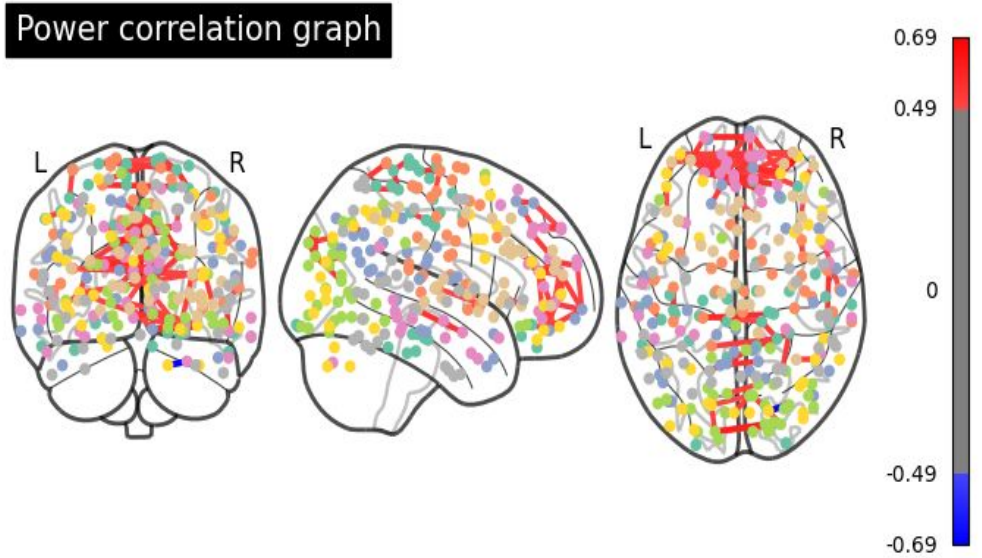
- Compute partial correlation matrix.
- Visualisation of the graph of connection.
- Visualisation of the graph of connection with hemispheric projections.



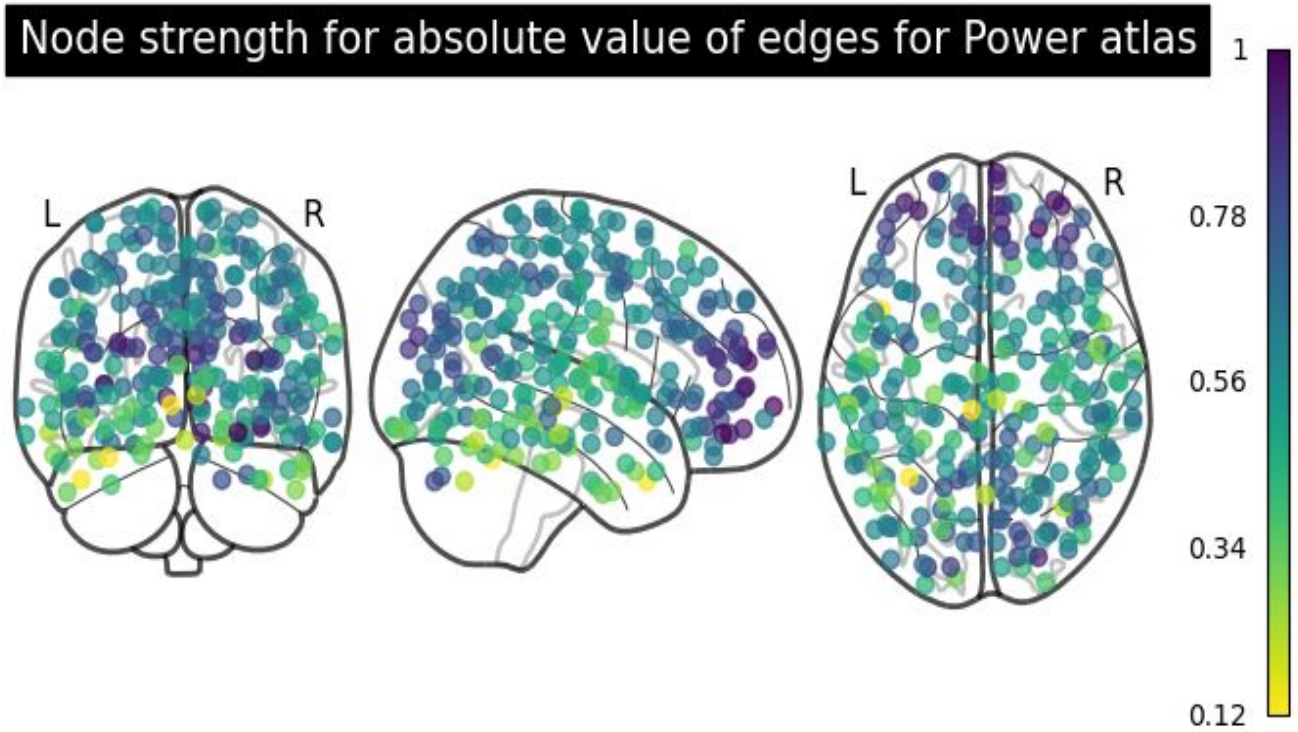


Correlation calculation using Power Atlas

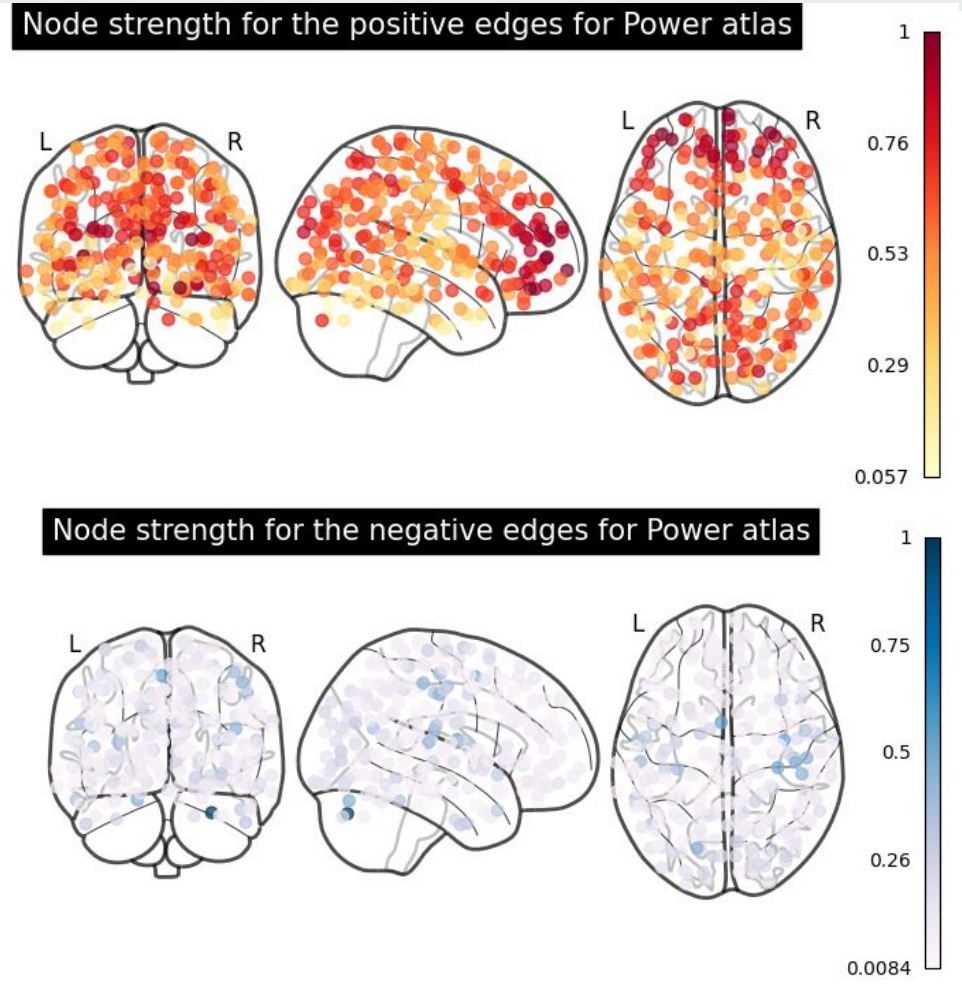




Edge Strength visualisation

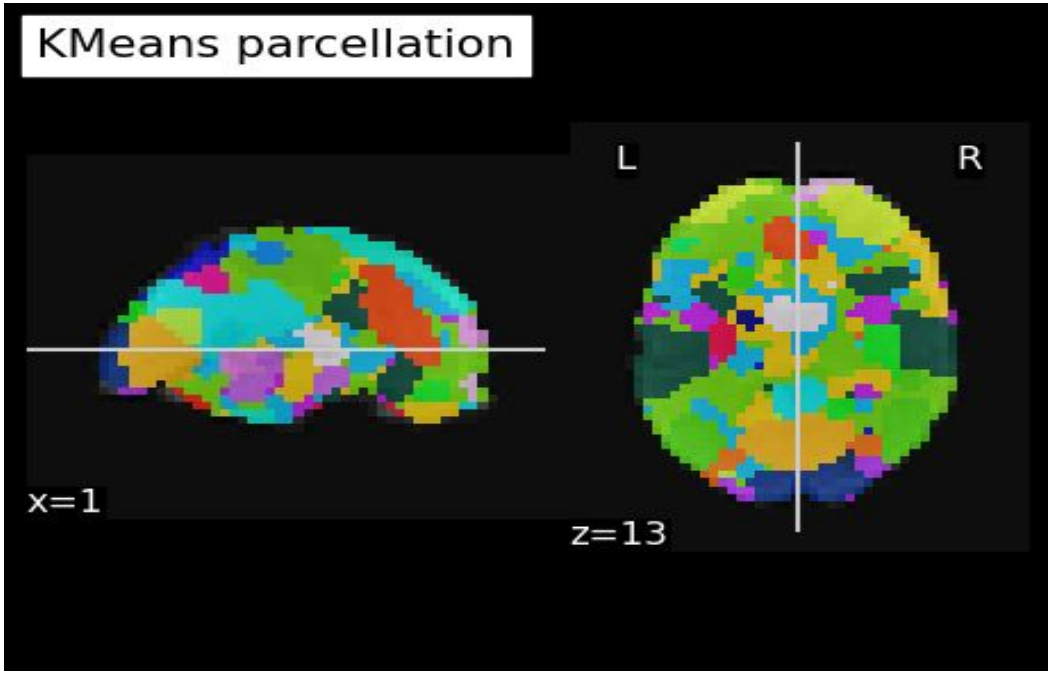


Positive and negative structure of the Edge strength



K means clustering analysis

- 30 clusters.
- Standardization and smoothing performed before clustering analysis.



Dataset preparation

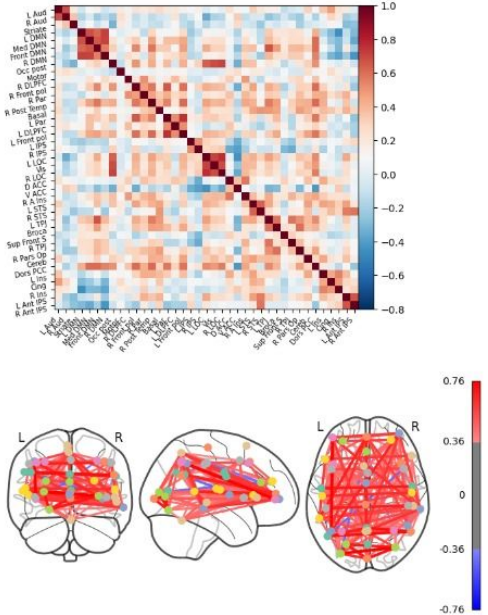
NiLearn is a Python package that performs machine learning on neuroimaging data.

Step 1: Download dataset and Atlas from OpenNeuro. (Preprocessing pipeline fMRIPrep already run on the dataset)

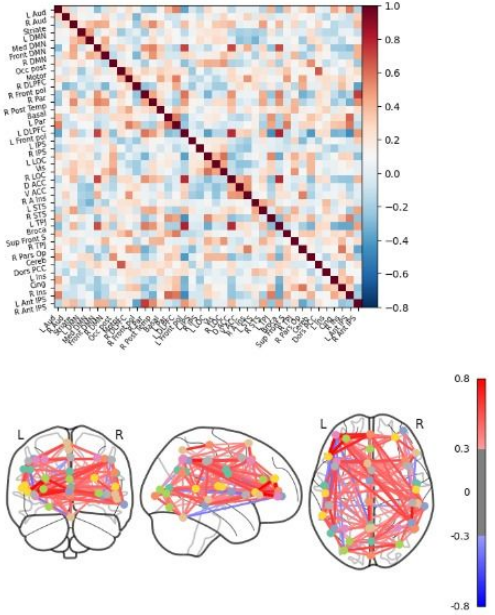
Step 2: Get brain signal time series data from the ROIs defined by the atlas. Regress out noise using confonds file generated by fMRIPrep.

Step 3: Calculate correlation matrices using the time series data.

Step 3: Correlation Graph samples



Adult



Child

How train and test dataset is created

- We create graph matrix using the correlated features and pass that matrix to networkx function to create a dataset format that can be used in GNN algorithms.
- We split the entire dataset by 80% for training and 20% for test.
- We also maintain a label list with values 1(child) and 0(adult). We use the list to compare the prediction from the model with the original label list.

Models trained

Hyperparameters

- Epochs trained: 30
- Activation function - Relu, Softmax(last layer)
- Optimizer - Adam
- Learning rate - 0.005, 0.01
- Loss function - NLLLoss

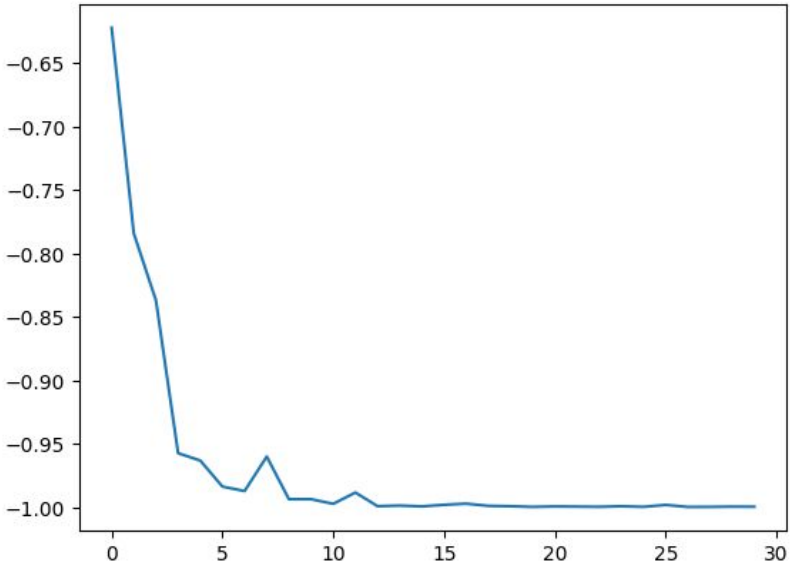
Models

1. GCN Conv Layers
2. EDGE Conv Layers
3. GAT
4. Graph Sage
5. Graph Neural ODE

GCN Conv layers

Train: 100.00%,

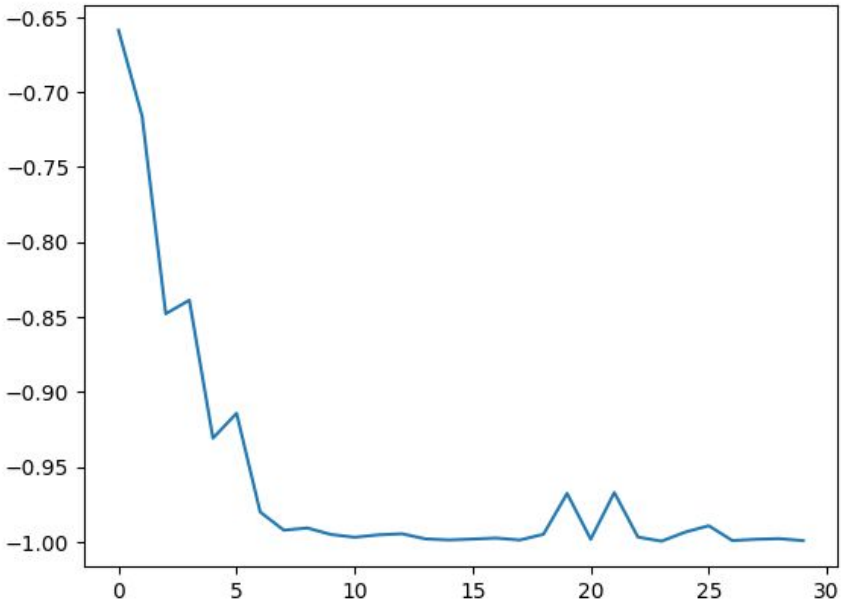
Test: 77.42%



EDGE Conv layers

Train: 100.00%

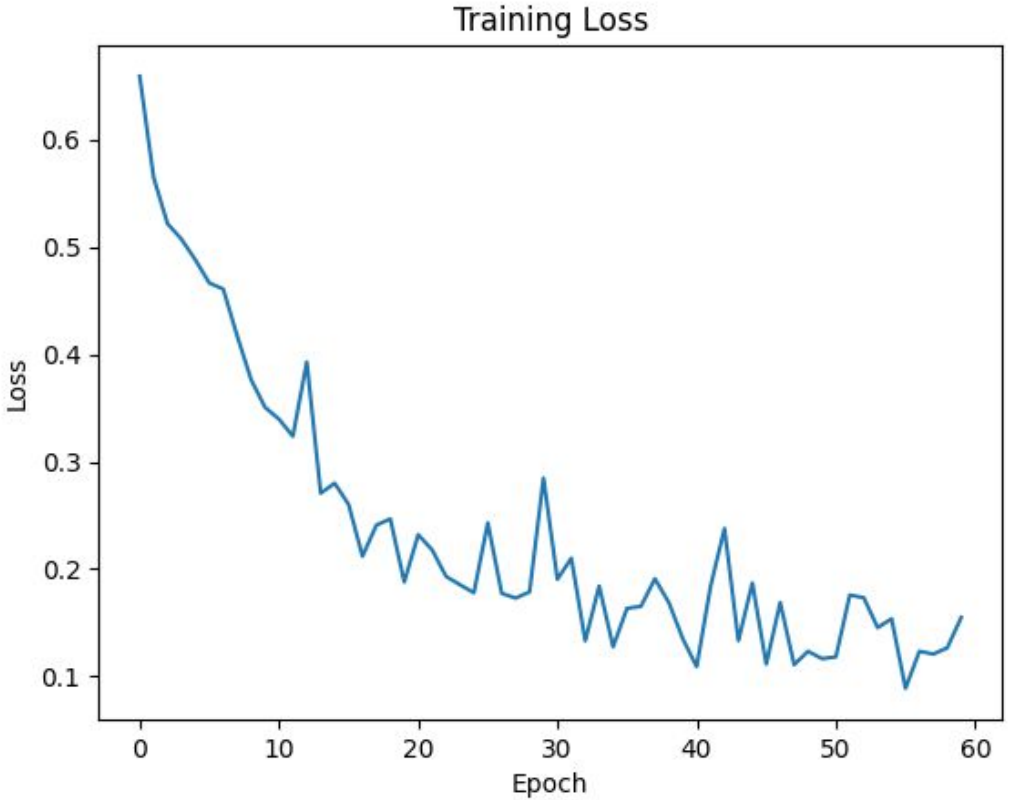
Test: 87.10%



GAT

Train Acc: 98.39%

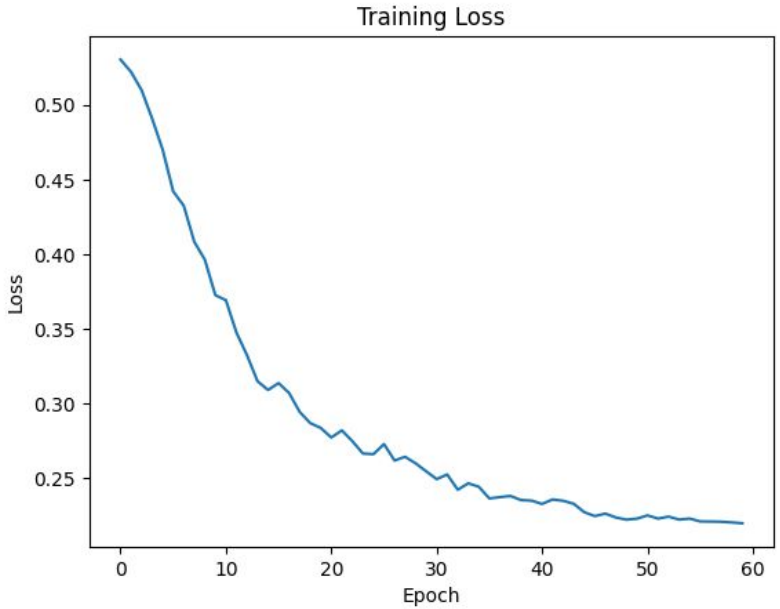
Test Acc: 93.55 %



Graph Sage

Train Acc: 100%

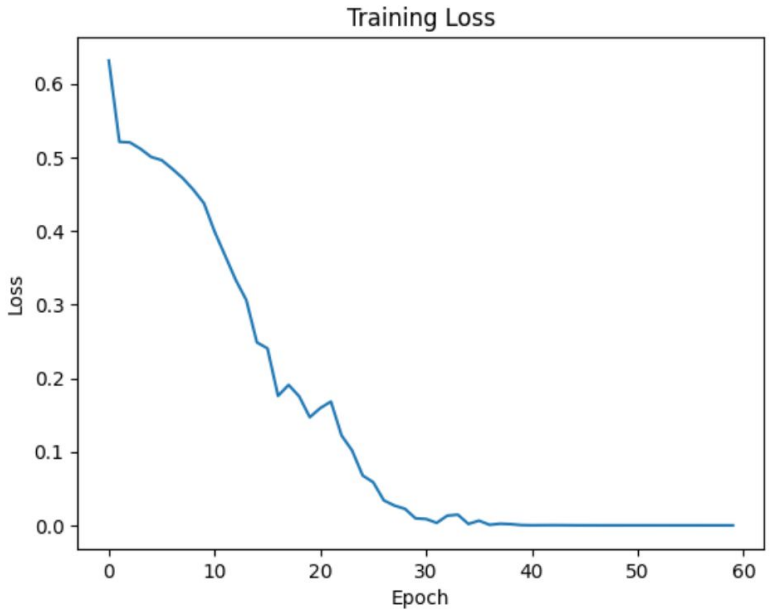
Test Acc: 96.77%



Graph Neural ODE

Train Acc: 100%

Test Acc: 93%



Comparison table

Model	No of epochs trained	Test accuracy(%)
GCN Conv Layers	30	77.42
EDGE Conv Layers	30	87.10
GAT	30	93.55
Graph Neural ODE	30	93.55
Graph Sage	30	96.77

Summary

References

[1] Richardson, H., Lisandrelli, G., Riobueno-Naylor, A. et al. Development of the social brain from age three to twelve years. Nat Commun 9, 1027 (2018). https://doi.org/10.1038/s41467-018-03399-2

[2]https://medium.com/stanford-cs224w/gnns-in-neuroscience-graph-convolutional-networks-for-fmri-analysis-8a2e933bd802

[3][3]https://nilearn.github.io/stable/auto_examples/03_connectivity/plot_multi_subject_connectome.html#sphx-glr-auto-examples-03-connectivity-plot-multi-subject-connectome-py

[4]https://nilearn.github.io/dev/auto_examples/03_connectivity/plot_sphere_based_connectome.html

[5]https://nilearn.github.io/stable/auto_examples/03_connectivity/plot_data_driven_parcellations.html