Graph Neural Networks for FC Fingerprinting

Thomas Muha

Advisor: Gowtham Atluri

Abstract

Functional connectomes are representations of brain connectivity, and classifying them has many uses in precision psychiatry. This project applies two kinds of graph neural networks to the task of FC fingerprinting: **graph matching networks** and **Siamese graph convolutional networks**. Graph neural networks can be used for matching and classifying functional connectomes due to their graph structure.
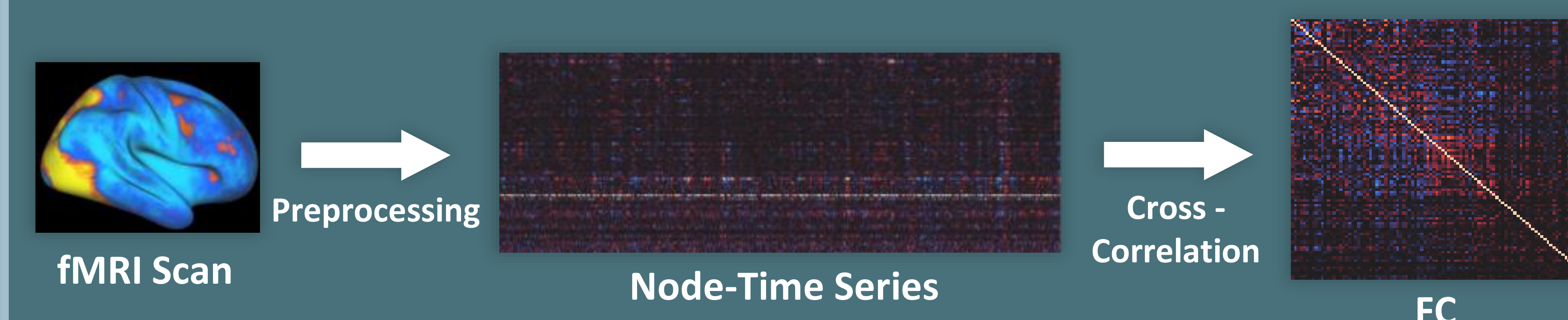
FC Fingerprinting

FC fingerprinting is the task of determining whether two FCs belong to the same subject or to different subjects. A model trained on this task can learn distinguishing structural features.

FC 1 → Model ← FC 2

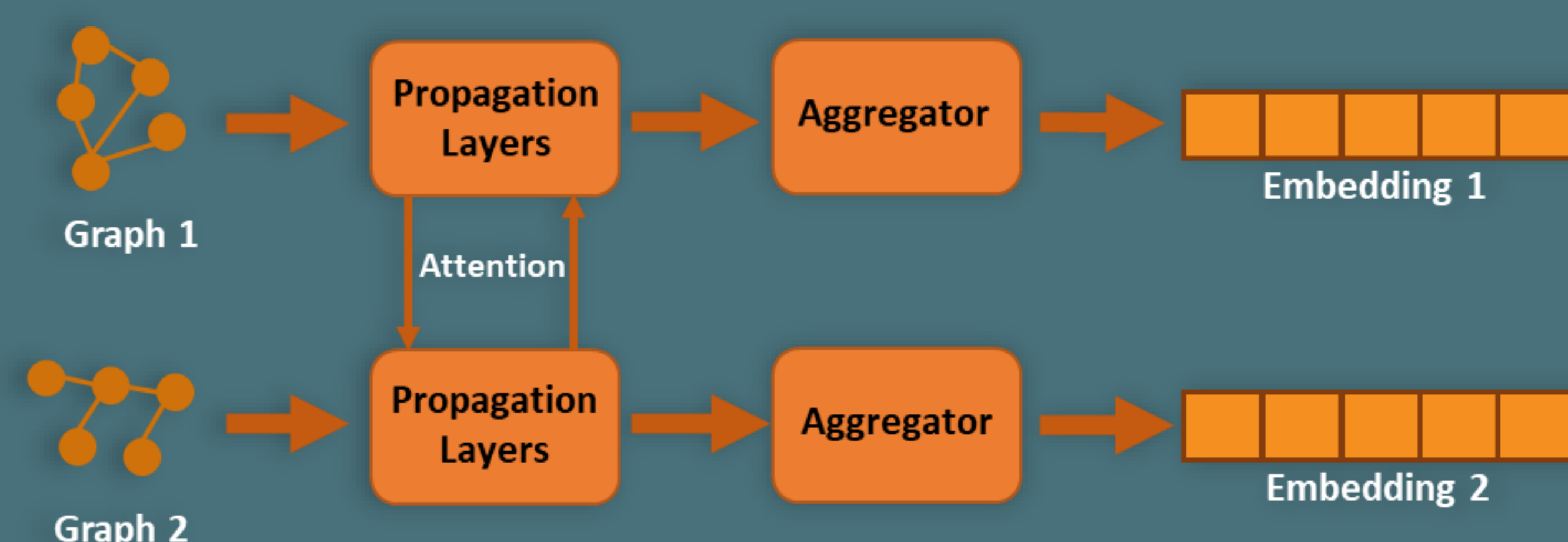**Prediction: Same Subject or Different Subjects**
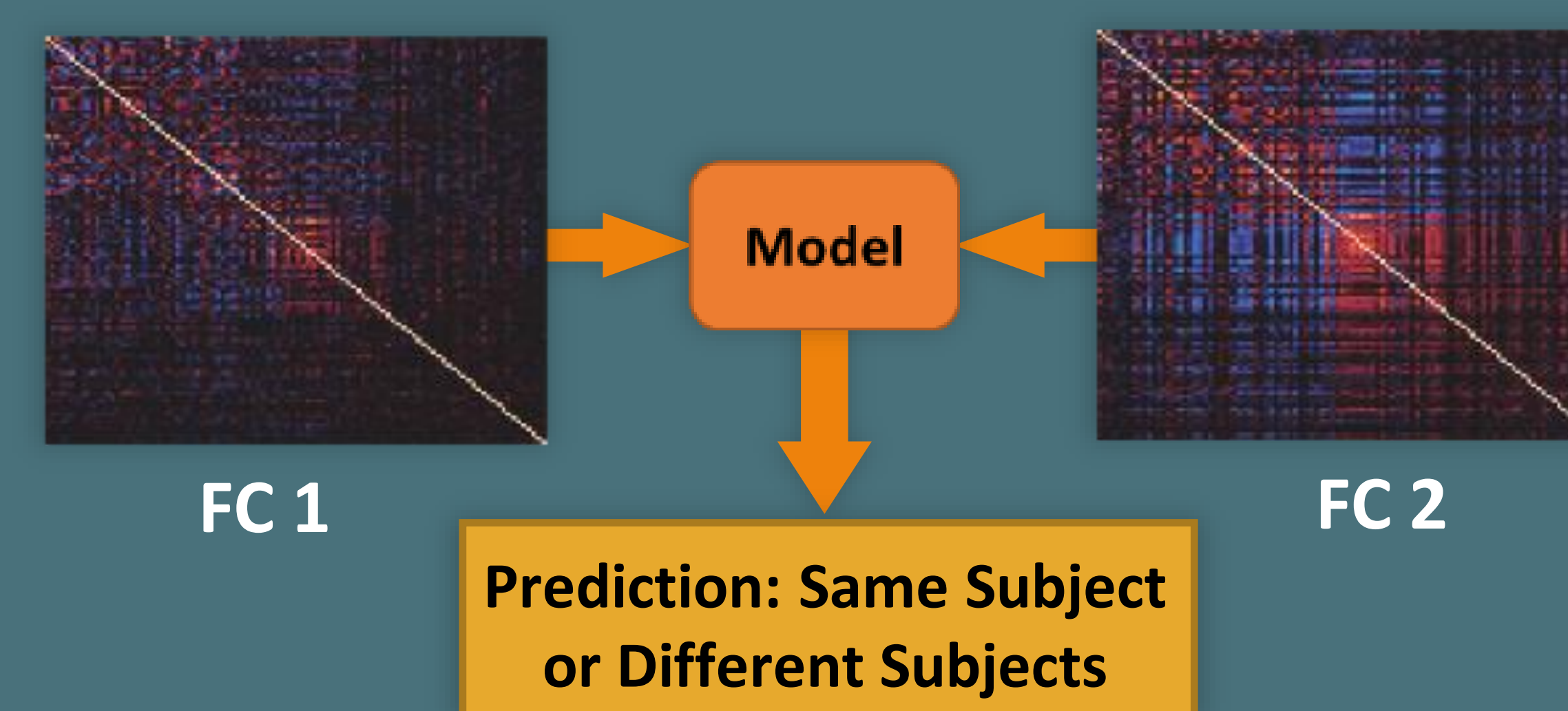
Computing FCs

FCs are constructed from fMRI scans, which contain brain activity over time measured by blood activity. The scan is preprocessed with the HCP functional pipeline and mapped to nodes found using group-PCA and group-spatial ICA to get a node-time series of activity. The FC is constructed from the cross-correlation between every node in this time series. In this project, the node-time series were from the HCP1200 PTN dataset released under the Human Connectome Project.


fMRI Scan → Preprocessing → Node-Time Series → Cross-Correlation → FC

Graph Matching Networks

Graph 1 → Propagation Layers → Aggregator → Embedding 1

Attention

Graph 2 → Propagation Layers → Aggregator → Embedding 2

Each graph is embedded to a vector, where similar graphs are closer in the embedding space. The GMN utilizes an attention mechanism in its propagation layers: when a node is updated, the difference with the same node in the other graph is incorporated.

Siamese Graph Convolutional Networks

Graph 1 → GCN

Graph 2 → GCN

→ Dot Product Layer → Similarity Layer → Similarity Value
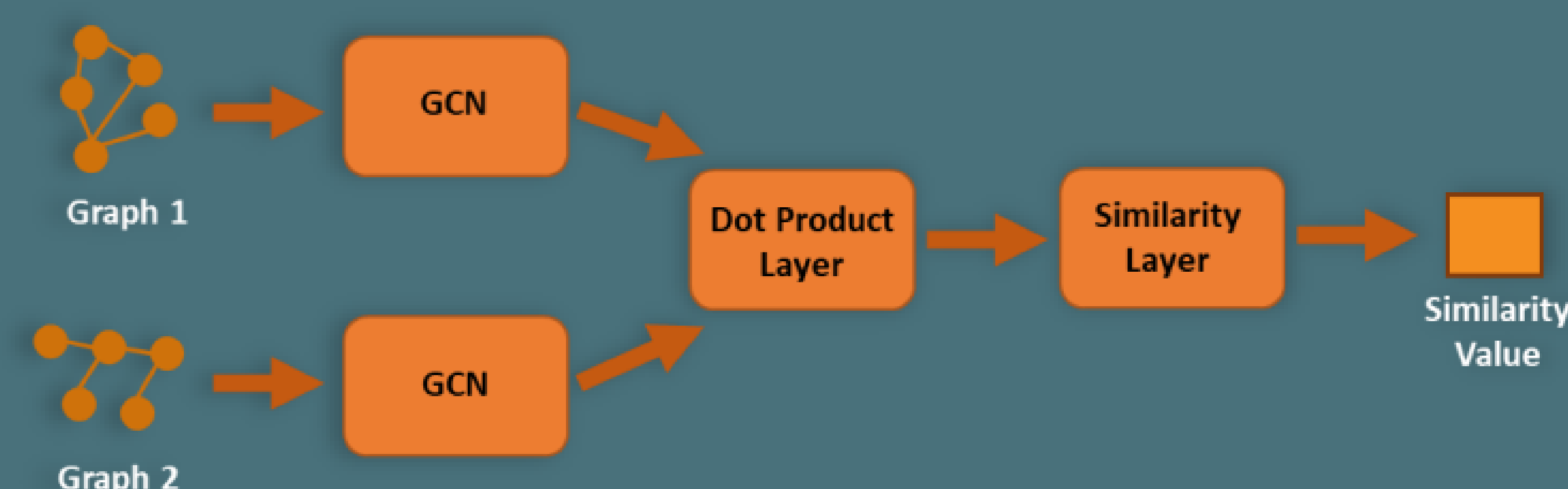
Each graph is processed by a GCN, which share weights. In GCNs, spectral filter layers apply convolutions to map input features to output features. Each GCN output is combined and mapped to a similarity value. Higher-order S-GCNs utilize community structure through a co-occurrence frequency matrix of random walks.

## Results

Models were evaluated with 5-fold cross-validation.

Model	Avg. Accuracy
MLP	0.852 ± 0.005
GMN	0.861 ± 0.010
S-GCN	0.948 ± 0.003
HS-GCN	0.952 ± 0.005

The MLP that was used is a simple network that embeds a graph, making it a baseline model for comparisons. The GMN had similar performance to the MLP.

The Siamese GCN performed very well on the task. However, the addition of the higher-order information in the HS-GCN brought only slight improvement.

Building the Dataset

The dataset was constructed from 1000 subjects with 4 FCs each. For each of the 6 possible same-subject pairs, one different-subject pair was created to achieve a balanced dataset. The 12,000 pairs were then split into training, validation, and test sets.

FCs → Make Pairs → Same-Subject Pairs / Different-Subject Pairs → Shuffle, Split → Training Set / Validation Set / Test Set
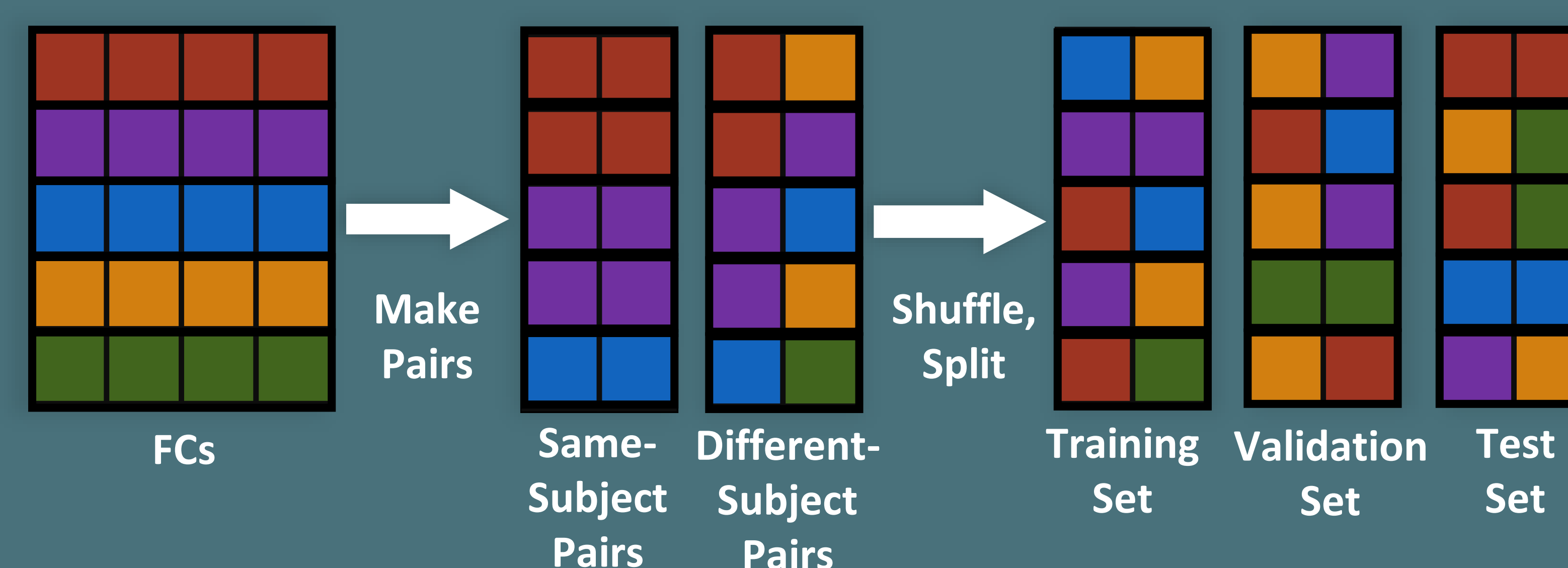
Challenges

Each network was built with PyTorch, in which my previous experience was limited.

Model training was unstable at first. Several methods were used to mitigate this.

References

Deep Graph Similarity Learning for Brain Data Analysis. Ahmed et al. November 2019.

Graph Matching Networks for Learning the Similarity of Graph Structured Objects. Dullien et al. May 2019.

Feature Selection Framework for Functional Connectome Fingerprinting. Atluri G., Li K., Wisner K. February 2021