DGL-24 Kaggle Competition: Deleep Graph Based Learning

Adam Jones 01853462
Alfie Chenery 01862873
Avaneesh Deleep 01869362
Luca Mehl 01894664
Oli Callaghan 01508136

I. Introduction

In this project we designed a GNN framework to predict high-resolution brain connectivity graphs from corresponding low-resolution graphs. The aim for our framework was to accurately produce high-resolution data from low-resolution brain scans, allowing lower-resource healthcare facilities to perform low-resolution brain scans and reliably upscale the results using our framework. We expand on the prior proposed AGSR-Net framework (Isallari and Rekik [2021]), to produce **Adversarial Graph Local Super-Resolution Network (AGLSR)**.

II. Datasets

We are provided with two datasets: Low-Resolution (LR) data, and High-Resolution (HR) data. LR data is 160x160 brain connectivity adjacency matrices between the 160 connectomes according to Dosenbach's atlas (Dosenbach et al. [2010]), while HR data is 268x268 brain connectivity adjacency matrices between the 268 connectomes according to Shen's atlas (Shen et al. [2013]).

Each training example is a LR matrix paired with an HR matrix. Since the training data does not include initial node embeddings, we initialise the embedding matrix to the identity matrix. We additionally perform pre-processing of both LR and HR data where we replace NaN values with zeros and negative values with their absolute value. This is because we care about connection strength and not the connection ”direction”. Thus each matrix entry is a value between 0 and 1. We additionally perform data truncation of the LR data by rounding it to 5 decimal places to improve generalisability. Then, we post process our results to ensure a symmetric matrix, and fill the principle diagonal with 1 entries.

The data is split into 167 training examples, and 112 test examples. To test the efficacy of different changes to the architecture we used 3-fold cross validation (3FCV). With 3FCV, we split the training data into thirds and and use one of the thirds as a validation set to test a model trained on the other two thirds. This allows us to make better use of the small dataset by iterating over the whole set 3 times. Additionally, we can ensure that the performance difference is consistent and does not depend solely on the data split. We evaluate performance using a number of metrics detailed in our results, and tune hyperparameters to achieve the best performance during 3FCV.

Once we have found the optimal hyperparameters, we then train a model on the whole training dataset and use it to predict HR graphs from the test data. By training on the whole dataset we are able to achieve better performance.

III. Methods

Our framework is built on top of the AGSR-Net framework (Isallari and Rekik [2021]) with notable improvements to certain areas of the architecture. This architecture consists of a U-net block which is designed to learn useful feature embeddings. Since brain graphs conventionally have no node features, the U-Net block is designed to learn topology-based node embeddings. Specifically, we iteratively pool nodes to extract multi-scale topological features, passing embeddings through our novel Expressive Transformation GCN (ETGCN) layers to aggregate neighborhood information.

A. Expressive Transformation GCN (ETGCN)

Expressive Transformation GCN (ETGCN) layers, inspired by GIN layers, are similar to normal GCN layers except the feature transformation is performed using a 2-layer MLP instead of a single linear layer. This allows the layer to learn a more expressive non-linear feature transformation for each layer while keeping the model depth (and thus aggregation neighbourhood size and space and time complexity) the same. It differs from GIN in that the feature transformation is performed before neighbour aggregation.

This final learned embedding matrix is then passed to the super resolution block. This block up-scales the LR graph of 160 nodes to a HR graph of 268 nodes and uses the learned embedding matrix to predict the edges between the nodes in the up-scaled graph. Because we can only up-scale by integer multiples, we specifically upscale from 160 nodes to 320 nodes, then remove a specific set of nodes to reach the requisite 268 nodes, matching our HR dataset.

Following this there are a number of stacked Probabilistic Local GCN (PLGCN) layers which refine the embeddings of the HR graph. Finally the output is post-processed to ensure symmetry in the adjacency matrix by averaging the output with its transpose and filling the principle diagonal with ones.

B. Probabilistic Local GCN (PLGCN)

The generated HR adjacency graphs are fully connected, with varying connection strengths. It doesn't make sense for a node to aggregate even from nodes with very weak connections, so we designed Probabilistic Local GCN (PLGCN) an aggregation scheme which prioritises aggregating from strongly-connected neighbours. It is inspired by GraphSAGE - we pick k neighbours of each node to aggregate from. We implemented a few different sampling strategies but found that randomly sampling neighbours with probabilities proportional to their connection strength worked best, possibly because it balances prioritising highly connected nodes while also introducing an element of randomness which helps the model generalise better. We found that k=50 neighbours sampled led to good performance.

These blocks combined define the generator which takes in the LR matrix and produces an estimated HR matrix. In addition to the generator, we define a discriminator network which is fed the output of the generator and also a noisy version of the ground truth HR. The Gaussian noise acts as a form of regularization to make the discriminator more robust. The goal of the discriminator is to identify when the input is the generator's output, and when it is the ground truth. Thus the generator and discriminator compete such that the generator learns better generated graphs to fool the discriminator, and the discriminator learns to distinguish generated graphs from true graphs.

C. Dropout

We tried adding dropout to the dense layer of the discriminator to increase generalisability but found that this did not improve results. As an alternative we configured an ensemble of discriminators with dropout (Mordido et al. [2018]), which had little to no effect on the loss during training or validation.

D. Permutation Invariance and Equivariance

Our model is permutation equivariant since all layers used are equivariant. The model is not permutation invariant. The PLGCN layers are equivariant since relabelling the nodes would not affect the nodes aggregated from if the sampled indices are also permuted. Furthermore, the weights of each neighbour will be unaffected, so the sample distribution will be the same. The ETGCN layers are equivariant since the same weights are applied to all the node embeddings. Therefore reordering the nodes would just reorder the output in the same way. The GSR layers are permutation equivariant because if we permute the graph adjacency matrix, the eigenvectors are permuted equally. The graph pooling and unpooling layers are equivariant because we choose the top k scoring nodes to remain, regardless of their index within the graph.

IV. Results and Discussion

Figure 1 shows that the model performance was very stable across the 3 folds, as can be seen by the very narrow error bars, representing the standard deviation across all 3 folds. This tells us that our model's performance is not strongly dependent on the specific training data split it was trained on.

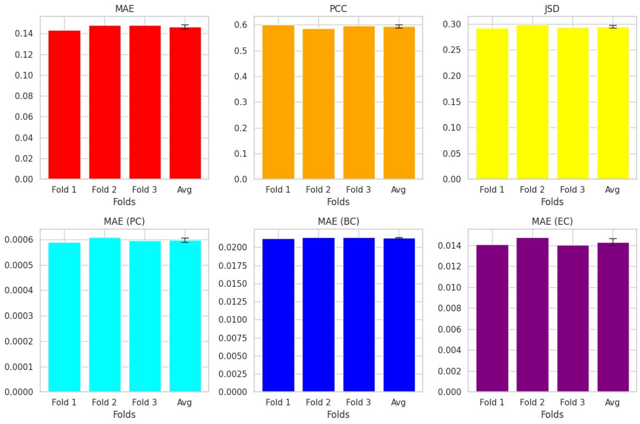

Fig. 1. Results from 3-fold Cross Validation

The Pearson Correlation Coefficient lies between -1 and 1 and quantifies the correlation between the connections in the predicted and ground-truth HR graphs. This averages to 0.595, indicating a reasonably high positive correlation. The MAE averages to 0.147. As the connection strengths vary from 0 to 1, this is quite a significant error. The Jensen-Shannon Divergence is between 0 and 1, and averages to 0.295. This implies that our distributions are reasonably similar.

We see that the topological centrality metrics are generally quite low, indicating that our super-resolved graph is topologically similar to the ground truth HR graph. We see that the average MAE of the Pagerank Centrality (PC), Betweenness Centrality (BC), and Eigenvector Centrality (EC), PC is lowest while BC is highest. BC for each node is calculated by the number of shortest paths that pass through it. This indicates that our model is able to predict topological structure well but struggles the most with making graphs whose shortest-path trees align with the ground-truth graphs' shortest-path trees.

We can conclude that our super-resolved graphs generally match the topology and general connective structure of the ground-truth HR matrices well (struggling the most with shortest paths), but the specific connection strengths have room for improvement.

Public Kaggle score: 0.135023

Private Kaggle score: 0.165149

Kaggle Ranking: 29

(note: some improvements were added subsequent to Kaggle competition closure, so the scores listed here are not reflective of the improvements and results discussed above or the provided code.)

RAM usage: 1.43GB

Total training time: 226min

Time per training fold: 43min

Time per fold evaluation: 97min

We are able to reduce RAM usage through the PLGCN layers which selectively aggregate from nodes rather than all nodes. The training time is mainly due to evaluation of the folds due to the time-consuming process of creating graphs to compare their centralities.

References

Megi Isallari and Islem Rekik. Brain graph super-resolution using adversarial graph neural network with application to functional brain connectivity. *Medical Image Analysis*, 71:102084, 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2021.102084.

Nico Dosenbach, Binyam Nardos, Alexander Cohen, Damien Fair, Jonathan Power, Jessica Church, Steven Nelson, Gagan Wig, Alecia Vogel, Christina Lessov-Schlaggar, Kelly Barnes, Joseph Dubis, Eric Feczko, Rebecca Coalson, John Pruett, Deanna Barch, Steve Petersen, and Bradley Schlaggar. Prediction of individual brain maturity using fmri. *Science (New York, N.Y.)*, 329:1358–61, 09 2010. doi: 10.1126/science.1194144.

Xilin Shen, F Tokoglu, X Papademetris, and Robert Constable. Groupwise whole-brain parcellation from resting-state fmri data for network node identification. *NeuroImage*, 82, 06 2013. doi: 10.1016/j.neuroimage.2013.05.081.

Gonçalo Mordido, Haojin Yang, and Christoph Meinel. Dropout-gan: Learning from a dynamic ensemble of discriminators. *CoRR*, abs/1807.11346, 2018. URL `http://arxiv.org/abs/1807.11346`.

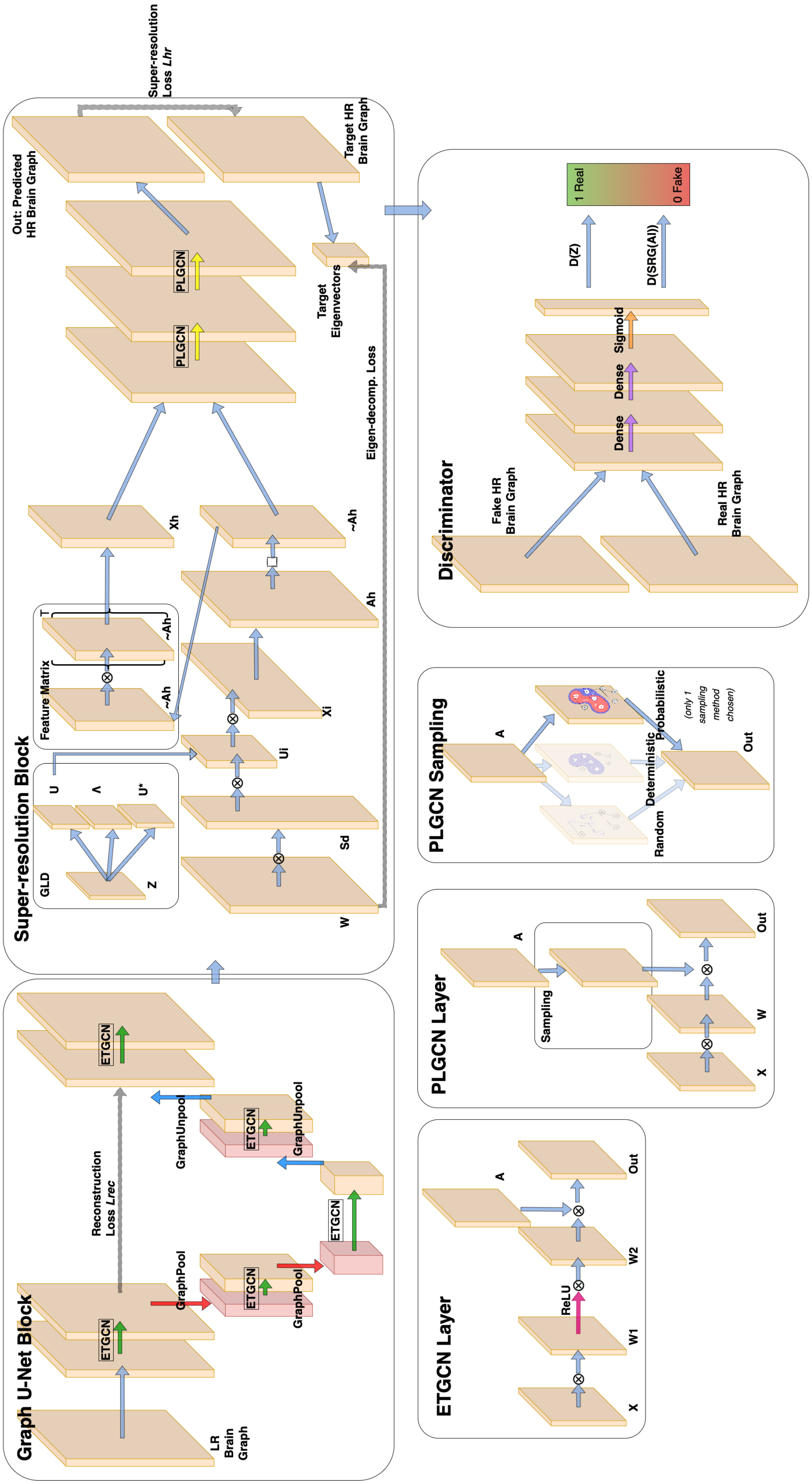

Fig. 2. Figure: AGLSR Architecture