DGL-24 Kaggle Competition: Adjacent Aces

Reinis Tony Jenny Olivia Iqbal

I. Introduction

The problem domain put forth is to produce a novel Generative Graph Neural Network that converts low-resolution brain graphs to high-resolution brain graphs. The nature of the problem arises since it is relatively much easier to measure low-resolution brain graphs compared to high-resolution brain graphs. However, producing a way that extrapolates the corresponding high-resolution brain graphs from their low-resolution counterparts greatly saves resources given that it is within an expected, error range.

The problem definition can be roughly formulated in the following expression:

$$f(\mathbf{A}^{LR}) = \hat{\mathbf{A}}^{HR} \approx \mathbf{A}^{HR}$$

Subject to further explanation, we have utilized the previously defined AGSR-Net [2] and incrementally added our own changes to it in order to increase its performance and further solidify the structure and overall generation of the high-resolution brain graphs.

II. Datasets

The dataset has been leveraged from the SLIM functional MRI dataset for the competition comprising of low-resolution brain connectivity graphs and high-resolution brain connectivity graphs. For any given graph, the matrix denotes the connectivity between any two given parts of the brain. For the sake of simplicity, the brain connectivity matrices will be denoted as $\mathbf{A}$ henceforth. Each element $\mathbf{A}_{ij}$ denotes the connectivity between a section $\boldsymbol{i}$ and $\boldsymbol{j}$ in any given

$$\mathbf{A}^{LR} \in \mathbb{R}^{160\times 160} \qquad \mathbf{A}^{HR} \in \mathbb{R}^{268\times 268}$$

The dataset comprises 167 pairs of low-resolution and high-resolution graphs for training and 112 samples of low-resolution to be used for testing and submitting the predicted graphs to the Kaggle portal.

While not intuitive, it is still under debate whether the weights in these graphs can be negative or not. For the sake of simplicity, as part of pre-processing, all negative values are removed from the dataset.

These matrices can further be vectorized or anti-vectorized for a linear representation depending on the need, into a single vector. The given cases produce:

$$\mathbf{A}^{LR} \in \mathbb{R}^{1\times 12720} \qquad \mathbf{A}^{HR} \in \mathbb{R}^{1\times 35778}$$

III. Methods

A. Original Method

Our novel brain network super-resolution framework is predicated on the foundational AGSR-Net architecture [1], encompassing a U-Net-based super-resolution generator juxtaposed with a discriminator.

The AGSR-Net is rooted in a U-Net structure serving as an auto-encoder. It breaks the symmetry of the U-Net architecture by super-resolving the low-resolution brain graph node content. With a GSR layer and two graph convolutional network layers, the network could further learn the adjacency matrices and node embeddings in the HR graph. At last, AGSR-Net uses a GAN structure to incorporate adversarial regularization to align their respective distributions.

In our produced model named **AbGSR-Net(Attention-based Graph Super-Resolution Net)** we attempted numerous additional rectifications to excel performance. The detailed structure of the whole model can be seen through figure 2.

B. Attention based U-net

The archetypal AGSR-Net [2] harnesses a U-Net infused with standard GCN layers, subsequently downscaling and then super-resolving the low-resolution brain graph node content. To refine the efficiency and performance of the U-Net, owe supplant the GCN layers with **GAT layers**, thereby integrating an attention mechanism into the architecture. The utilization of GAT layers confers two salient benefits: 1) helping the U-Net focus on the more important features/nodes and 2) helping the network converge faster.

The GAT's efficacy was measured across various configurations within the network, with optimal results manifesting when implemented solely within the U-Net. This optimal configuration is attributed to the U-Net's augmented capacity for accentuating pivotal features and nodes, whilst concurrently attenuating the influence of extraneous elements, albeit with a propensity for overfitting. The embedding learning from the U-Net is then fed to the GSR block for high-resolution mapping.

C. GAN Stabilization and Disuse

Another novelty for AGSR-Net is its discriminator, which engenders a GAN-like pedagogical paradigm whereby it alternates between generative and discriminative functions. However, even though this method initially seemed promising, our practical tests showed that it didn't work as expected. Here we'll introduce the problems we found and the changes we finally made.

1) Fixing the GAN training loop: During the training of the Discriminator and Generator in the GSR model, we encountered a problem where the fake and real data were mixed up in the training loop, and the gradients were detached for generator training, leading to no learning. Specifically, the generator's loss (except for its MSE loss part) kept increasing without contributing to the adversarial training. This was fixed by correctly using the ground truth high-resolution graphs as the real data and the generated ones as the fake data. Moreover, the gradients of the generated fake data were only detached for the discriminator's training and kept for the generator's training for learning to occur.

Fixing the GAN training loop revealed the need for stabilizing the training - a common challenge with GANs. We tried several methods to make the GAN architecture more stable. Here we'll go through our journey in more detail.

2) Label Smoothing: To smooth the learning process, we used label smoothing to add uniform noise to the real and fake labels in the discriminator.

3) Architectural Adjustment: Architectural changes were also made to ensure the generator and discriminator were well-matched in complexity so that one would not overpower the other. This was achieved by adjusting the number of layers, and features, adding layer normalization and dropout, while observing the resulting changes in the training logs (especially loss dynamics, and the predictions of the discriminator on fake data before and after training, and real data). The goal was to achieve balance where the discriminator and generator both keep improving, while not strongly over-performing one another.

4) Weight Adjustment: We carefully adjusted the weights of their mean squared error (MSE) losses to maintain a balance between the generator and discriminator losses. Reducing the weight of the reconstruction loss encouraged the generator to produce more varied outputs while reducing overfitting.

5) Replacing Dense Layers: A significant improvement was made by replacing dense layers in the discriminator with Graph Convolutional Network (GCN) or Graph Attention Network (GAT) layers, which enhanced its performance by leveraging the graph data more effectively.

D. Other Improvements

1) Normalization and Dropout: To further improve the model's training process, we improved the model's stability and learning dynamics by introducing layer normalization and dropout layers, which helped prevent overfitting by making the data more varied and the model more regularized.

2) Adding Noise: We also added Gaussian noise to both the low-resolution source and high-resolution target graph matrices to further diversify the data as a form of augmentation and regularisation.

3) Optimiser Adjustment: We implemented learning rate schedulers to gradually reduce the learning rate, leading to a more stable convergence. Moreover, setting the Adam optimizer betas to (0.5, 0.999) resulted in faster convergence, likely due to the lower value of beta1.

All these improvements stabilized the Discriminator-Generator training and performance while making intuitive sense. **However, the best improvement was seen when removing the adversarial training from the model while keeping all other improvements of regularisation, augmentation, scheduling, etc.** This allowed us to surpass the performance of the original AGSR-Net and justified our intuition with the original model.

IV. Results and Discussions

Figure 1 depicts the metrics for the average of the 3-fold cross-validation on the dataset of 167 samples. We could see there is a significant improvement for our AbGSR-Net compared with the baseline AGSR-Net.

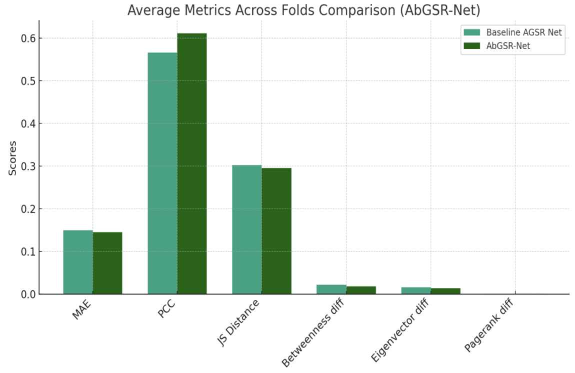

Fig. 1: Metrics Comparison between baseline and our model

The following table shows the time taken per fold, memory usage per fold, and the total time taken for the 3-fold cross-validation:

TABLE I: Performance Metrics

	Time Taken (s)	Memory Usage (MB)
Fold 1	4705.92	589.22
Fold 2	4813.18	878.11
Fold 3	4854.71	796.82
Total	14376.81	–

The presented results show that the model is reliable and the results are reproducible. The Jenson Shannon distance is relatively low depicting that the produced high-resolution brain maps depict a very similar distribution to that of their component low-resolution counterparts. The high Pearson Correlation Coefficient also shows that the learned node embeddings preserve the relationship within the graphs and the results can be trusted to be accurate and valid. The low Mean Absolute Error shows the overall model being fairly accurate to the expectation and performing well.

After the training, the complete training dataset was taken and a singular model was trained on it to produce a concrete model trained on everything. Our rank in the Kaggle competition was 10 with a score of 0.150747.

REFERENCES

[1] M. Isallari and I. Rekik. Gsr-net: Graph super-resolution network for predicting high-resolution from low-resolution functional brain connectomes. In Machine Learning in Medical Imaging: 11th International Workshop, MLMI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings 11, pages 139–149. Springer, 2020.

[2] 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

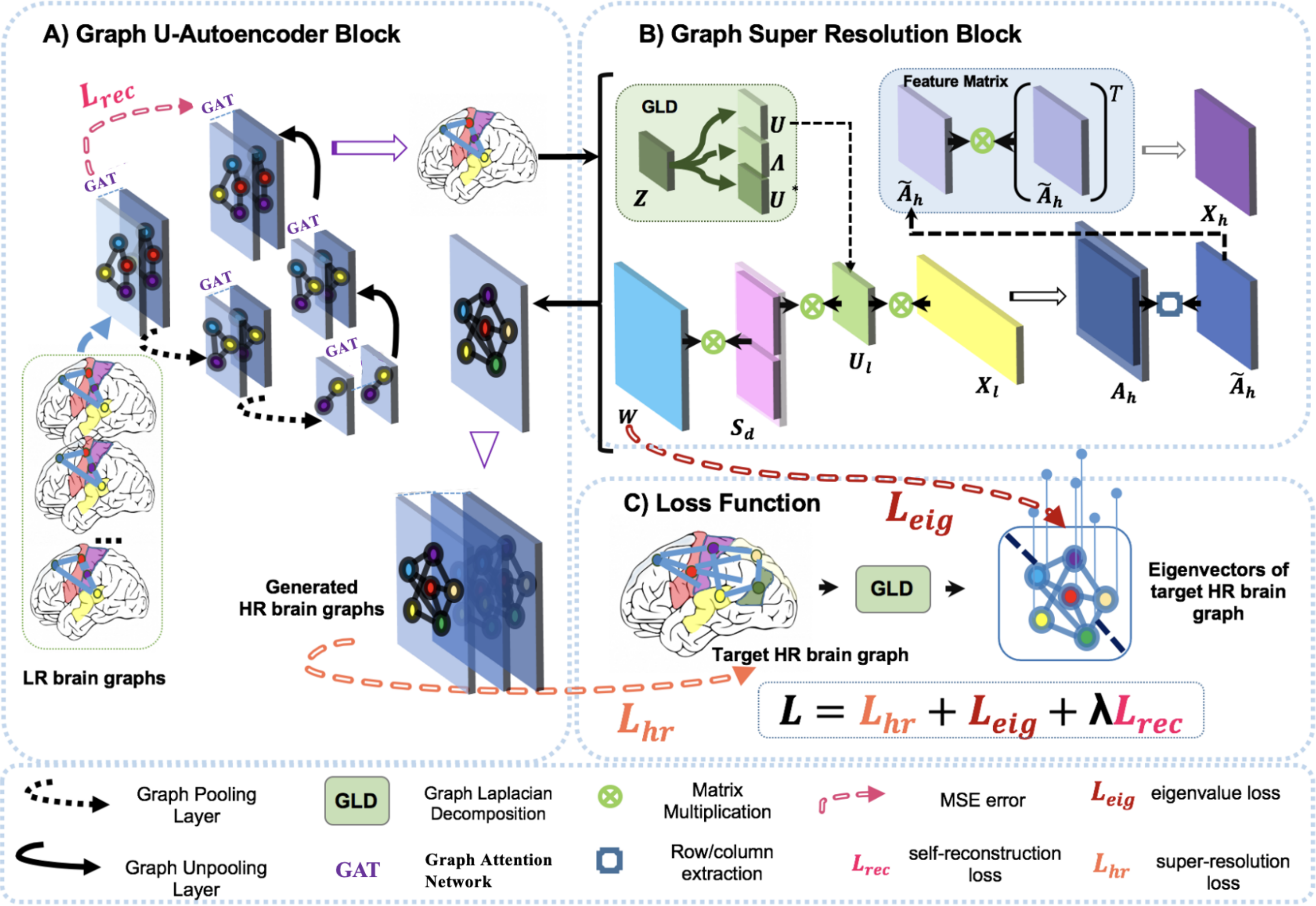

Fig. 2: AbGSR-Net