DGL_24 Kaggle Competition: Edge Engineers

Aran Ubhi *01730438* | Daniel Kang *02299574* | Euan Goodbrand *02546834* | Lewis Kerley *01943126* | Majed Althonayan *01949378*

I. Introduction

Traditionally, producing high resolution brain graphs from low resolution brain graphs has been a tedious and difficult task. Therefore, in this following report we aim to solve the following research question: "How can generative Graph Neural Networks be effectively utilized in the context of brain graph super-resolution to predict high-resolution brain connectivity graphs from their low-resolution counterparts?"

The objectives of our methods are to design and implement a generative GNN model that can, given a low-resolution brain graph, accurately and precisely infer a high-resolution counterpart. Our model should also display expressiveness, generalisability and scalability in order to capture the complex graph data of unseen graphs for the purpose of creating a robust and efficient model capable of addressing the aforementioned research question.

As for the motivation behind our model, we conducted a thorough literature review which included the analysis of recognised, existing solutions as well as current topics of research. Given this range of solutions, we analysed their effectiveness in our specific context, and from which modelled an architecture containing a medley of techniques each of which serving a purpose to assist us in achieving our established goals and objectives.

II. Datasets

The competition dataset contains both the low and high resolution encodings of brain connectivity represented in symmetric weighted matrices, where each element of the matrix represents the neural correlation between two brain regions. The dataset consisting of 279 samples, is split up into training, public test and private test sets, with each set containing 167, 56 and 56 samples respectively. The training and public test sets were used to develop and train our model with the ultimate task of obtaining the maximum performance on the private test set.

The feature initialisation utilised in our model is based on AGSR-NET (Isallari and Rekik, 2021) where the features are initialised with an identity vector and where node feature embeddings are consequently learnt by "locally averaging the features of its neighboring nodes based on their connectivity weights." When tested against other feature initialisations, such as random initialisation or topological initialisations we found that the AGSR-NET initialisation yields the greatest results. As for the data post processing, we found that combining predictions from multiple GNNs, each of which trained uniquely, led to the greatest results by increasing generalisation and decreasing overfitting as can be seen in the works of (Wei, Qiao and Jadav, 2023).

III. Methods

Building upon the foundational principles of AGSR-NET, EnGIN is designed to be a robust and expressive generative model capable of transforming low-resolution brain graphs into their high-resolution counterparts with remarkable accuracy.

A. EnGIN Architecture Components

EnGIN's architecture is composed of four pivotal blocks (illustrated in Figure 1): a graph U-NET block, a graph super-resolution block, a discriminator block, and an ensemble block. Each of these components plays a critical role in the overall functionality of the framework:

Graph U-NET Block: Inspired by the graph U-NET architecture proposed by Gao and Ji (2019), this block transitions from traditional node-centric models to a graph-oriented approach. It accepts a low-resolution connectivity matrix as input, where each node is initialized by an identity matrix, and outputs a feature-embedded low-resolution matrix.

Graph Super-Resolution Block: This segment is crucial for the elevation of resolution. The low-resolution matrix undergoes transformation via a Graph Super-Resolution (GSR) layer, followed by two layers of Graph Isomorphism Networks (GIN). The GSR layer aims to break the symmetry inherent in the U-NET architecture, using the propagation rule to upscale the connectivity matrix. The subsequent GIN layers are pivotal for accurately capturing the high-resolution matrix's feature embeddings. The choice of GIN, in preference to other GNN variants like GCN or GraphSAGE, was motivated by its permutation invariance and injective aggregation capabilities, which are paramount for identifying and distinguishing intricate brain regions' features.

Discriminator Block: This component is responsible for comparing the distribution of the model-generated data against the ground truth, facilitating the regularization of the super-resolution block to enhance the model's output.

Ensemble Block: To boost generalizability, we adopt ensemble methods (Dietterich, 2000), creating multiple models trained with varied parameters. The ensemble's output is an average, producing a refined high-resolution connectivity matrix. The final prediction is calculated such that, given the high resolution prediction by the i-th model is A_i. The ensemble prediction, $\bar{A}$ is the average of these predictions: $\bar{A} = \frac{1}{N}\sum_{i=1}^{N} A_i$

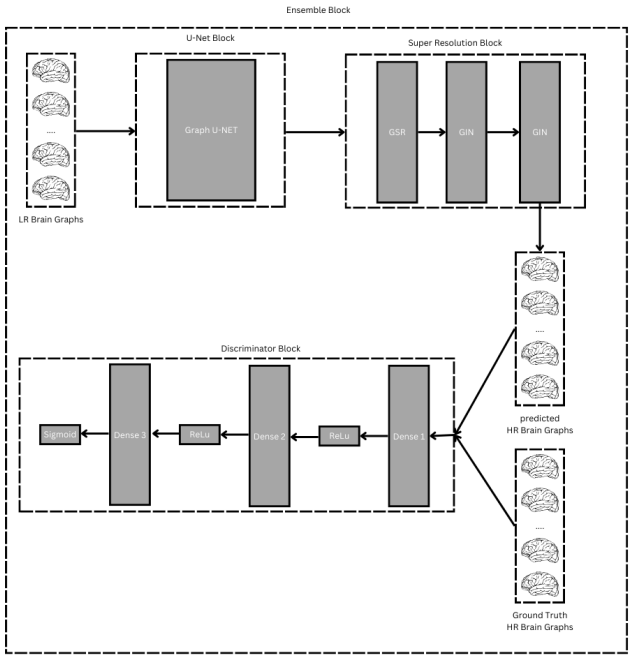

Fig. 1. EnGIN Architecture

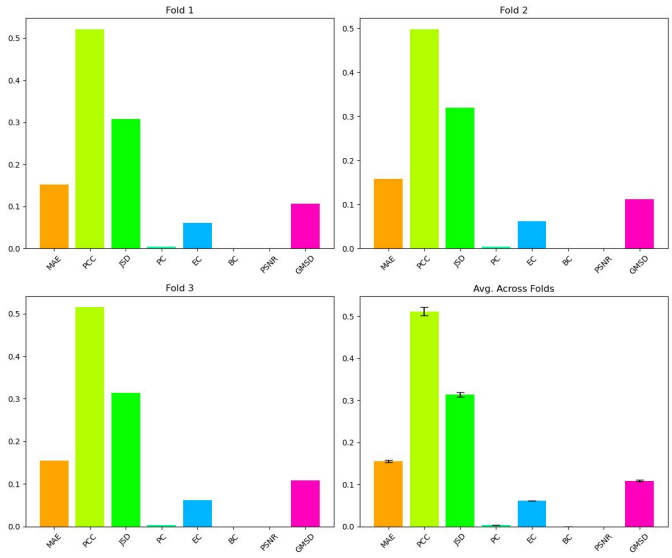

Fig. 2. Losses Across Folds

B. Model Selection Rationale

Our journey to EnGIN involved an evaluation of GNN architectures, assessing their permutation invariance and injectivity of aggregation functions. GCN, for instance, relies on spectral convolutions that, while powerful for learning graph features, lack permutation invariance due to their dependence on the graph's Laplacian eigen basis, which can vary with node ordering. GraphSAGE extends GCN by sampling and aggregating features from a node's neighbours; however, its aggregation functions (mean, max, LSTM) are not strictly injective, leading to information loss when distinguishing between different graph structures. Conversely, GIN stands out by employing a more sophisticated aggregation function that combines node features with a learnable, injective function, ensuring both permutation invariance and the retention of distinctive graph structures through its injective aggregation. This combination allows GIN to effectively capture the nuances of complex graph topologies, making it particularly suitable for our task. Our decision to select GIN, augmented with MLPs for additional feature processing, was significantly informed by these considerations and the insights provided in the "18-Limitations of Graph Neural Networks" report by the Syllogisms team on Weights & Biases. GIN is central to EnGIN's design for its inherent node-permutation invariance and equivariance. This ensures the model's robustness and adaptability, preserving the topological integrity of the brain graphs irrespective of initial node ordering.

C. Training and Testing

The model underwent rigorous training using a 3-fold cross-validation method with shuffling, extending over 200 epochs for each fold. Subsequent retraining over the entire dataset aimed at maximizing accuracy. For testing, mini-batch Stochastic Gradient Descent (SGD) with Mean Squared Error (MSE) loss were employed.

D. Model Foundation

The foundation for EnGIN was derived from a comprehensive analysis of existing solutions, including GSR-NET [Isallari and Rekik, 2020], IMANGraphNet [Mhiri et al., 2021], AGSR-NET and TIS-NET (Pala, Islem Mhiri and Islem Rekik, 2021). AGSR-NET in particular, demonstrated superior results in our context, leading to its adoption as our base model. Our exploration extended to various GNN models and features, like Dropout, batch normalization, early stopping, one-sided positive label smoothing, and sampling, culminating in the identification of the most effective GNN layers and optimal hyperparameters.

IV. Results and Discussion

The findings of our 3-fold cross-validation in the initial 167 samples showed consistent results across all 3 folds as can be seen in figure 2 where after 200 epochs, the losses began to level out indicating the presence of a local minimum. Moreover, the total training time of our model was around 50 minutes across 3 folds of 200 epochs, while our RAM usage averaged between 150-200MB per epoch. These results speak to the scalability and efficiency of the GNN model. Our process of retraining the model includes running the pre-trained model on the entire 167 training samples before testing it on the public test set. Finally, our Kaggle score was 0.128226 resulting in our team obtaining a public ranking of 8th place.

REFERENCES

[1] Isallari, M. and Rekik, I. (2021). Brain graph super-resolution using adversarial graph neural network with application to functional brain connectivity. Medical Image Analysis, 71, p.102084. doi:https://doi.org/10.1016/j.media.2021.102084.

[2] Wei, W., Qiao, M. and Jadav, D. (2023). GNN-Ensemble: Towards Random Decision Graph Neural Networks. [online] Available at: https://arxiv.org/pdf/2303.11376.pdf.

[3] Gao, H. and Ji, S. (2019). Graph U-Nets. arXiv:1905.05178 [cs, stat]. [online] Available at: https://arxiv.org/abs/1905.05178.

[4] Dietterich, T.G. (2000). Ensemble Methods in Machine Learning. Multiple Classifier Systems, [online] 1857, pp.1–15.

[5] A. and Mahjoub, M. (n.d.). Non-isomorphic Inter-modality Graph Alignment and Synthesis for Holistic Brain Mapping. [online] Available at: https://arxiv.org/pdf/2107.06281.pdf [Accessed 11 Mar. 2024].

[6] Isallari, M. and Rekik, I. (n.d.). GSR-Net: Graph Super-Resolution Network for Predicting High-Resolution from Low-Resolution Functional Brain Connectomes.

[7] Pala, F., Islem Mhiri and Islem Rekik (2021). Template-Based Inter-modality Super-Resolution of Brain Connectivity. Lecture Notes in Computer Science, pp.70–82.

[8] "18-Limitations of Graph Neural Networks." Weights & Biases, Syllogismos team. Accessed March 11, 2024.

[9] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen. Improved Techniques for Training GANs (2016).