DGL-25 Kaggle Competition: Team Super, 29^{th} place

Asia Belfiore, 02129867, ab6124

Ginevra Cepparulo, 06028727, gc1424

Humaid Ibrahim, 06002972, hai24

Aslihan Gulseren, 06008251, ag724

I. Methodology & Novelty (40 points)

A. Problem description & motivation (5 points)

Graph super-resolution is a problem that involves reconstructing a high-resolution (HR) graph from a low-resolution (LR) one using Graph Neural Networks (GNNs). In the Brain Graph Super-Resolution Challenge, the goal is to predict fine-grained neural connectivity maps from coarser ones. Graph super-resolution for Brain Graphs offers significant advantages compared to traditional methodologies. Firstly, high-resolution fMRI scans require extensive MRI processing, denoising and segmentation, which this method bypasses by generating HR graphs directly. This leads to significant cost and time reductions. Furthermore, HR graphs are computationally expensive to process and super-resolution allows more efficient analysis by first handling LR graphs [1]. Furthermore, higher-resolution connectivity graphs provide better insights into neurological disorders and brain function.
Graph super-resolution has several potential applications. In neuroscience, it can help identify patterns in brain connectivity. This gives insights into disorders like Alzheimer's [2]. For drug discovery, it can predict molecular interaction networks at a higher resolution.
A significant challenge in graph super-resolution is ambiguity, as multiple HR graphs can correspond to the same LR graph. Additionally, high-quality training data is limited and thus requires powerful generative models for accurate results. Finally, HR graphs are larger and computationally expensive to handle. This makes real-time inference resource-intensive.

B. State-of-the art methods (3 points)

Current state-of-the-Art (SOTA) methods are shown in TABLE I.

TABLE I
Overview of GNN-based SOTA Models

Model Name	Brief Description
GSR-Net [3]	Refines graph-based super-resolution by employing a U-Net-inspired architecture with graph convolution, pooling and unpooling operations, alongside a specialized graph super-resolution layer. GSR-Net learns feature embeddings for brain regions, predicts high-resolution connectomes while preserving structural integrity. It integrates spectral theory to enhance resolution accuracy.
STP-GSR [2]	Refines GNN-based brain graph super-resolution by leveraging a primal-dual graph formulation to transform edge regression into node regression. This enables efficient topology-preserving learning in higher-order spaces while reducing computational complexity.
Dif-GSR [4]	Introduces a diffusion-based approach for graph super-resolution, leveraging a noising process to perturb low-resolution brain graphs, a conditional denoiser to reconstruct high-resolution graphs and a sampling module for HR graph generation. Dif-GSR effectively models complex data distributions. It overcomes the scalability and probability modeling challenges of graph-based GANs. This makes it a framework for brain connectome super-resolution across varying graph sizes, distributions and modalities.

C. Main figure (4 points)

A graphical representation of the model architecture and components is shown in Figure 1. The model's Training and Evaluation pipeline is described in Figure 2. Please refer to the Appendix Figures 5 and 6 for an expanded version of the aforementioned graphics.

D. Brief overview of the proposed GNN (5 points)

We propose Graph-Dif-GSR, a Diffusion-based generative GCN model based on the SOTA Dif-GSR model [4]. It builds upon a Denoising Diffusion Probabilistic Model (DDPM) which learns how to generate HR graphs by gradually adding noise to the target HR Graphs and learning how to remove noise conditionally based on the corresponding LR graph [4]. By accurately learning how to reverse the noising process, the DDPM is able to learn

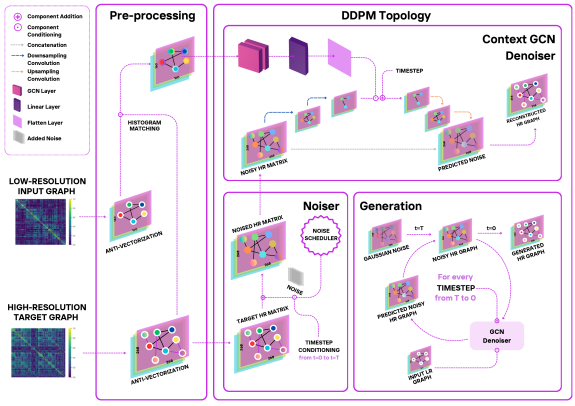

Fig. 1. Graph-Dif-GSR Architecture.

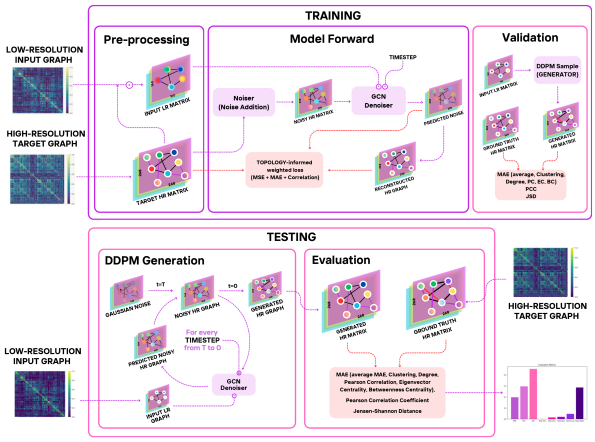

Fig. 2. Graph-Dif-GSR Training and Testing Pipeline.

how to generate realistic HR graphs from LR ones. The model comprises two main components:

- Noiser/Noise Scheduler: determines the noise added at each timestep during the model forward call.
- Conditional GCN-based Denoiser: predicts the noise that should be removed at each time step, conditioned on the source LR graph's topological features and reconstructs the original denoised HR graph. The Denoiser follows Dif-GSR's U-Net architecture [4], with a novel GCN-based embedder. It first downsamples the noised graph into a latent lower-dimensional representation, then incorporates the contextual time step and embedding of the source LR graph extrapolated through a custom GCN-based layer. The conditioning process allows the model to learn a mapping between source LR and target HR graphs.

The model's learning and generative pipeline consist of:

- Noising Process: during the forward call, the model gradually adds noise to the target graph based on the noise scheduler.
- Denoising Process: at each timestep, the model predicts the noise to be removed through the Denoiser and uses it to reconstruct the original graph. The aim of the model is to minimize the error between the real and predicted noise removed. We employ a custom hybrid topologically-informed loss in order to enhance the model's predictive ability specifically for connectome data.
- Generation/Sampling: the model starts from a fully-noised graph and gradually denoises it by conditioning on the given input LR graph, generating its corresponding HR graph.

The full processes are shown in depth in Figure 2.

E. Innovative components (10 points)

Inspired by SOTA models, we implemented the components and techniques described in TABLE II. We employ a histogram patching mechanism to align the statistical distribution of LR and HR graphs on a patch level, which enhances the model's ability to generate HR graphs that preserve the underlying topology of the graph. This technique is inspired by those used in image generation like [5], where they aligned figure statistics across local patches to align color distributions of low and high resolution images. Instead of color. we adapt this idea to graphs by aligning the distribution of edge weights in each patch, ensuring that local statistical properties are similar between the LR input and the HR target.

For the model, we incorporate a conditional GCN-based denoiser following U-net architecture. The LR graph is processed by a novel GCN-based context embedder that learns the global node representations from the LR graph, providing the model with an informative signal that is fused with the time embeddings.

In addition, we proposed a novel weighted loss function that incorporates an L1 reconstruction loss and a differentiable Pearson correlation loss to ensure that the generated graph closely matches the ground truth structurally. These complemented the MSE loss of diffusion models, which minimizes the difference between the noise and output of the Conditional Denoiser. The total loss is defined as: $\mathcal{L} = \alpha\mathcal{L}_\epsilon + \lambda\mathcal{L}_{PC} + \beta\mathcal{L}_{rec}$, where the weights were fine-tuned during the modelling process.

TABLE II
Innovative Components of the Proposed GNN Framework

Novel Contribution	Rationale
Patch-Level Histogram Matching for Data Alignment	The aligned data enhances the conditioning signal used in the denoiser by aligning LR's statistical distribution with HR.
Topology-Informed Loss	The hybrid loss helps the model learn better distribution alignments and structural consistency.
Diffusion-based Conditional GCN Denoiser	Incorporates a DDPM with a denoiser uses a U-Net-like architecture built from graph convolutional layers that are conditioned on the LR graph's topology.

F. Mathematical properties of the proposed GNN (13 points)

Permutation invariance (5 points)

a) A graph function f_k is a permutation-invariant if for each GCN layer k, we satisfy the following for any permutation matrix $\mathbf{P}$:
$f : \mathbb{R}^{n\times n} \times \mathbb{R}^{d_k \times n} \to \mathbb{R}^{d_{k+1}}$
$f(A, H_k) = f(P^T AP, H_k P)$. (1 point)

b) Model ensures permutation invariance via GCN-based architecture in ContextUnet-Graph. DenseGCNLayer computes normalized adjacency matrices, making operations invariant to node ordering. Diffusion process uniformly adds, removes noise, preserving invariance throughput. Topology-aware loss functions (Pearson correlation,MAE) reinforce this property, as supported by cross-validation results showing minimal metric variation across different node orderings. (4 points)

Permutation equivariance (5 points)

a) If P is a permutation matrix then: $H_{k+1}P = f(H_k P, P^T AP, \phi_k)$. Considering all nodes in a graph, GCN computation is permutation equivariant. If we permute the node indices, the node embeddings at each stage will be permuted in the same way (1 point)

b) Permutation equivariance is maintained through DenseGCNLayer, which applies same transformation to each node based on its neighborhood.Adjaceny normalization ensures that permuted input nodes yields equivalently permutated features. While the UNet includes some position-sensitive components, dominant GCN-based operations preserve equivariance, enabling consistent predictions regardless of node indexing.(4 points)

Expressiveness (3 points)

a) The expressiveness of a GNN refers to its ability to distinguish different local neighborhoods by mapping them to different embeddings. An expressive GNN maximizes its node representation capacity in the embedding space.

b) The model is expressive. It captures complex graph structures using DenseGCNLayer for neighborhood aggregation. The multi-scale UNet enhances local and global representations, while diffusion models high-resolution graph distributions. Its expressiveness is validated by a high Pearson correlation (0.226) in fold 2, demonstrating meaningful brain connectivity learning.(2 points)

II. Experimental Setup & Evaluation (27 points)

A. Results (9 points)

a) We employed two additional topological measures: Clustering MAE and Degree MAE, detailed in Table III. The reason behind using Clustering MAE is to ensure that our model reproduces the same connectivity patterns seen in the original HR graph. Similarly, we chose Degree MAE to ensure that the overall connectivity distribution in the predicted graph matches that in the original HR graph. To keep Degree MAE on the same numerical scale as other metrics, we multiply it by (10^{-2}).

b) Graph-Dif-GSR shows consistent results with very little variability among different folds, as show in Figure 3 and in the Appendix (Fig. 8), indicating that the model has good generalization power.

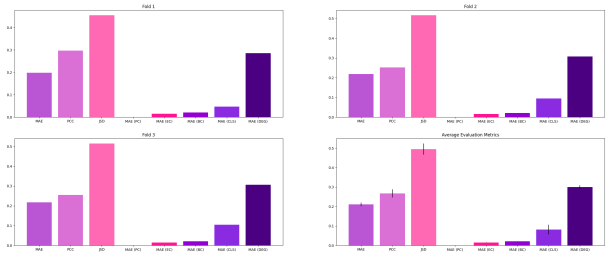

Fig. 3. Graph-Dif-GSR Performance Metrics Across All Folds

The model performs strongly on the topological measures, showing solid topology-preserving abilities. This is corroborated by the achieved low ($\leq$ 0.1) average error in all topological metrics (PC, EC, BC, Clustering), and slightly higher error in Degree MAE ($\approx$ 0.3) throughout different runs. The model achieved an average MAE of $\approx$ 0.2 across every run, which signals that the model has consistent overall good generational abilities. However, the model significantly struggles to accurately learn the underlying distribution, as shown by the average achieved JSD of $\approx$ 0.5 across every fold.

c) Diffusion models require significant computational resources. Our model peaked at around 16780 MB of VRAM and 1484 MB of RAM. The run time increases from 252 seconds to 320 seconds going from fold 1 to fold 3. The total training time was approximately 825 seconds, or 13.75 minutes. The full training time and computational usage are provided in Table VI in the appendix.

d) For the Kaggle submission, the provided train dataset was split into 80% for training and 20% for validation. The model was trained on the 80% split and its

performance was measured on the 20% split to ensure no overfitting has occurred. After training, the model was used to predict on the test set to generate the final submission file. The final Kaggle score was 0.193554, ranking us at 29th place.

TABLE III
Additional Topological/Geometric Measures

Measure Name	Brief Description & Rationale
Clustering MAE	This metric measures how closely the predicted graph's local clustering coefficients match the ground truth, capturing how well nodes cluster together.
Degree MAE	This measures the average absolute difference in node degrees between the predicted and ground truth graphs, indicating how many neighbors each node has.

B. Comparison Against Other Methods (6 points)

Figure 4 shows the comparison between metrics achieved by each model (Our Proposed Graph-Dif-GSR, a baseline 4-layered MLP and BasiraLab's baseline Dif-GSR [4]). The metrics for the 3-folds for each model are shown in the Appendix (Figures 8, 9, 10).

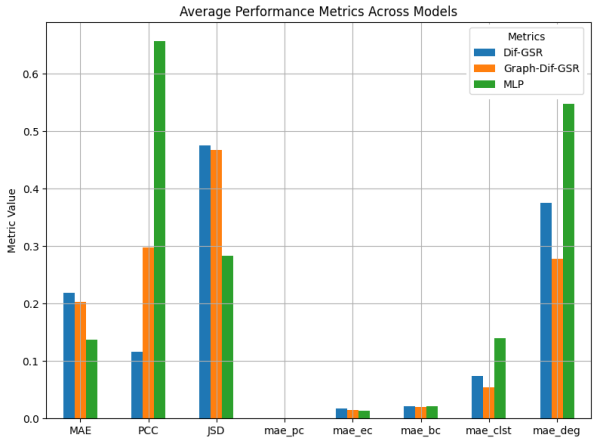

Fig. 4. Overall Comparison of Metrics Across All 3 Models. *mae_cls* denotes the MAE computed on the clustering coefficient, and *mae_deg* is the MAE on the node degree distribution.

Overall, Graph-Dif-GSR generally outperforms the SOTA Dif-GSR model across all base metrics (PC, EC,BC) and additional Topological ones (Clustering and Degree). However, it performs worse than the baseline MLP in MAE, PCC and JSD, but shows better results for betweenness centrality, clustering and degree error compared to the MLP. For the MAE_{pc}, all three models perform very similarly with minor differences. In terms of raw numerical error and distribution similarity, our model is worse than the MLP. However, it is more effective at preserving the graph's underlying structural and connectivity properties. This is an important ability for the tasks at hand, since it heavily relies on the network topology rather than just numerical matches, and for other important applications, like identifying the presence of neurodegenerative disorders [2].

C. Scalability of Your Proposed GNN Model (7 points)

The memory complexity Graph-Dif-GSR is predominantly driven by the dense representation of the input graph. In a dense graph scenario, the $N \times N$ adjacency matrix requires $O(N^2)$ memory. Additionally, we initialize node features using the identity matrix which results in node feature vectors of dimension N. However, embeddings are reduced to lower-dimensional representation (F where $F << N$) thus reducing the per-layer memory cost to $O(NF)$. The iterative diffusion process further increases the memory footprint by requiring storage of intermediate states across T timesteps, potentially leading to an overall memory requirement of $O(T(N+LNF))$, especially during training when these intermediates must be maintained for backpropagation.

Considering time complexity, each Conditional GCN Denoiser Network layer involves matrix multiplications with the dense adjacency matrix, resulting in a per-layer computational cost of $O(N^2)$. With L layers per timestep, a single diffusion iteration incurs $O(LN^2)$ operations. Since the model performs T diffusion iterations, the total time complexity becomes $O(TLN^2)$.

This quadratic dependency on the number of nodes, compounded with the iterative nature of diffusion, indicates that for larger datasets and more dense connectomes, the computational demands will increase significantly, highlighting a key scalability challenge for the model. Despite this, Table IV shows how in practice, Graph-Dif-GSR has a smaller parameter count compared to Dif-GSR, largely due to the introduction of GCN layers that reduced model complexity of the Conditional Denoiser Network, thus allowing us to achieve lower memory usage and computation time compared to SOTA Dif-GSR [4].

TABLE IV
Comparison of Model Sizes

Model	Number of parameters (10^6)
Graph-Dif-GSR	19.478
Dif-GSR	21.057
MLP	19.637

D. Reproducibility of Your Proposed GNN Model (5 points)

The model was trained and validated using 3-fold cross-validation with shuffling. The results per fold are shown in Fig 4. Across the three folds, Graph-Dif-GSR shows moderate consistency on the PCC, JSD and the topological MAEs. It is very consistent in the numerical MAE, with the differences being very minor between the folds. The clustering MAE is the most varied out of all

the metrics. This means it is more sensitive to shifts in the exact distribution of edge weights across different data splits.

The GNN's permutation invariance and equivariance help ensure that relabeling nodes does not affect how the model aggregates features or learns embeddings, thus outputting stable results for metrics like MAE or PCC across folds. However, local metrics like clustering are more sensitive to small changes in adjacency. While the GNN's expressiveness enables it to capture complex topological variations, slight distributional shifts can disproportionately affect clustering scores, explaining why it varies more across folds even though the model remains robust on more global, permutation-invariant measures.

III. Discussion & Reflections (8 points)

a) Our model excels at capturing and preserving the underlying network topology. It significantly improves the clustering and degree errors compared to the MLP and SOTA baselines, while marginally improving the betweenness centrality. This is crucial in applications involving neuroscientific analysis (as noted in [2]), where accurate representation of the connectivity is important. Additionally, the inclusion of a GCN for context embedding and conditioning on the LR graph provided substantial benefits compared to the SOTA Dif-GSR model across all evaluated metrics.
Despite the strong topological performance, Graph-Dif-GSR underperforms on raw numerical error metrics compared to the baseline MLP. The model trades-off numerical precision with preserving graph structure. Another weakness is that while the connectivity is better, the overall distribution similarity (as measured by JSD) is worse than the MLP baseline. While structural fidelity is high, the predicted connectivity values are not as good.

TABLE V
Strengths & Weaknesses of the Proposed GNN

Preservation of Graph Structure	As indicated by its lower errors in betweenness centrality, clustering and degree compared to the other models.
Improved Overall Performance vs. Dif-GSR by using GCNs	Incorporating a GCN-based context embedder with LR conditioning led to consistent superior performance compared to the Dif-GSR model.
Inferior Numerical Accuracy	Graph-Dif-GSR has higher raw numerical errors (MAE) compared to the MLP baseline.
Distribution Similarity Trade-Off	Despite its topological strengths, Graph-Dif-GSR shows worse distribution similarity (JSD and PCC) metrics relative to the MLP.

b) A key area for improvement concerns optimizing the diffusion process by reducing the number iterations and the computational overhead per iteration. One promising strategy could be to utilise consistency models or other sampling methods to enable better convergence, as suggested by the authors of Dif-GSR [4]. For example, Denoising Diffusion Implicit Models (DDIM) allow for non-Markovian sampling and have been shown to achieve faster convergence by refining the sampling distribution more effectively and avoiding to follow the gradual noise removal process [6]. Consistency models, instead, learn a direct mapping that is consistent with the diffusion process but allows for one-step or few-step generation, thereby removing the need for many iterative denoising steps. Taking another direction, once could optimize the diffusion process and reduce computational overhead by incorporating GCN sampling methods within the Conditional GCN Denoiser Network. Techniques such as node sampling, layer sampling and graph sampling can help by limiting the number of nodes or layers processed in each iteration [7][8][9]. This means, they could reduce both the memory footprint and the time complexity, making the model more scalable to larger, denser graphs.

References

[1] F. Pala, I. Mhiri, and I. Rekik, "Template-based inter-modality super-resolution of brain connectivity," in Predictive Intelligence in Medicine, I. Rekik, E. Adeli, S. H. Park, and J. Schnabel, Eds. Cham: Springer International Publishing, 2021, pp. 70–82.

[2] P. Singh and I. Rekik, Strongly Topology-Preserving GNNs for Brain Graph Super-Resolution. Springer Nature Switzerland, Oct. 2024, p. 124–136. [Online]. Available: http://dx.doi.org/10.1007/978-3-031-74561-4_11

[3] M. Isallari and I. Rekik, "Gsr-net: Graph super-resolution network for predicting high-resolution from low-resolution functional brain connectomes," 2020. [Online]. Available: https://arxiv.org/abs/2009.11080

[4] N. Rajadhyaksha and I. Rekik, "Diffusion-based graph super-resolution with application to connectomics," in Predictive Intelligence in Medicine, I. Rekik, E. Adeli, S. H. Park, C. Cintas, and G. Zamzmi, Eds. Cham: Springer Nature Switzerland, 2023, pp. 96–107.

[5] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," 2017. [Online]. Available: https://arxiv.org/abs/1703.06868

[6] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," CoRR, vol. abs/2010.02502, 2020. [Online]. Available: https://arxiv.org/abs/2010.02502

[7] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," CoRR, vol. abs/1706.02216, 2017. [Online]. Available: http://arxiv.org/abs/1706.02216

[8] D. Zou, Z. Hu, Y. Wang, S. Jiang, Y. Sun, and Q. Gu, "Layer-dependent importance sampling for training deep and large graph convolutional networks," in Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/91ba4a4478a66bee9812b0804b6f9d1b-Paper.pdf

[9] W. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," CoRR, vol. abs/1905.07953, 2019. [Online]. Available: http://arxiv.org/abs/1905.07953

TABLE VI
Performance Metrics for the 3 Models During Full Training and Inference Cycles

Model	Fold	Max RAM (MB)	Max VRAM (MB)	Time (sec)
Dif-GSR	1	1328.12109375	16756.5	166.199
	2	1419.1015625	16758.5	161.476
	3	1438.71875	16758.5	169.020
Graph-Dif-GSR	1	1373.078125	16778.5	252.753
	2	1465.24609375	16780.5	275.650
	3	1484.359375	16780.5	320.334
MLP	1	899.5390625	1956.5	182.774
	2	1021.71484375	1956.5	208.478
	3	1025.59375	1956.5	198.733

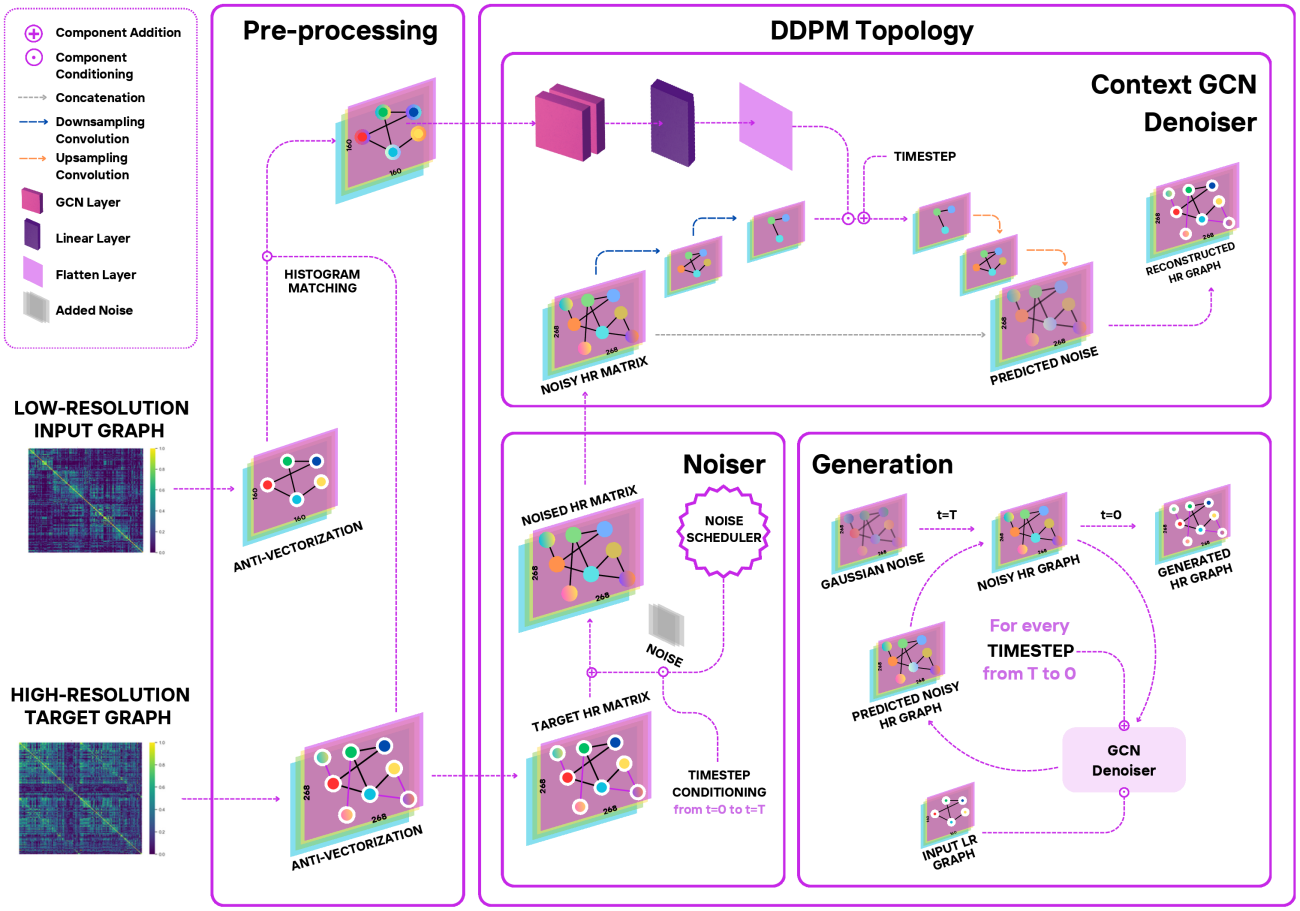

Fig. 5. Graph-Dif-GSR Architecture.

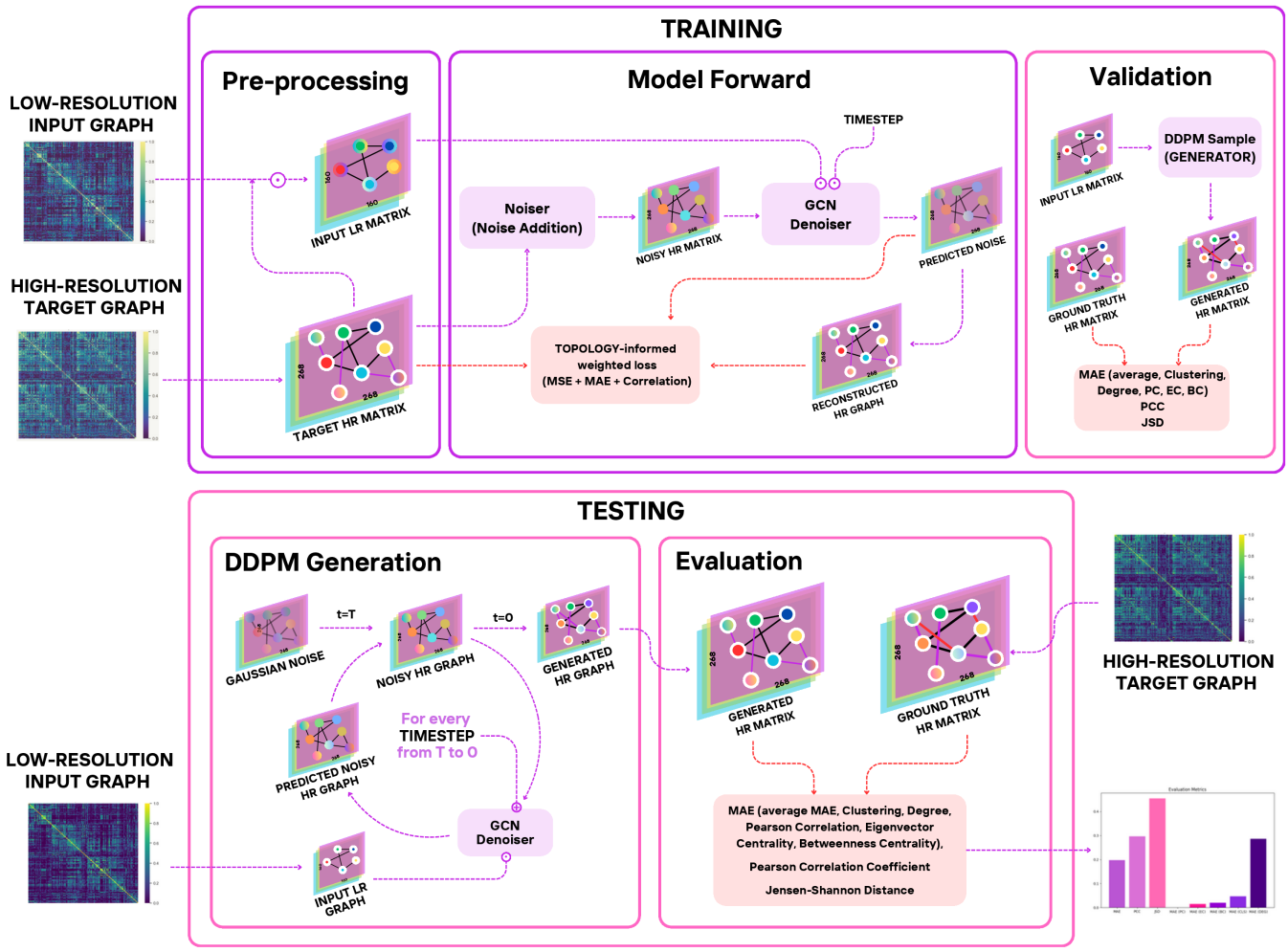

Fig. 6. Graph-Dif-GSR Training and Testing Pipeline.

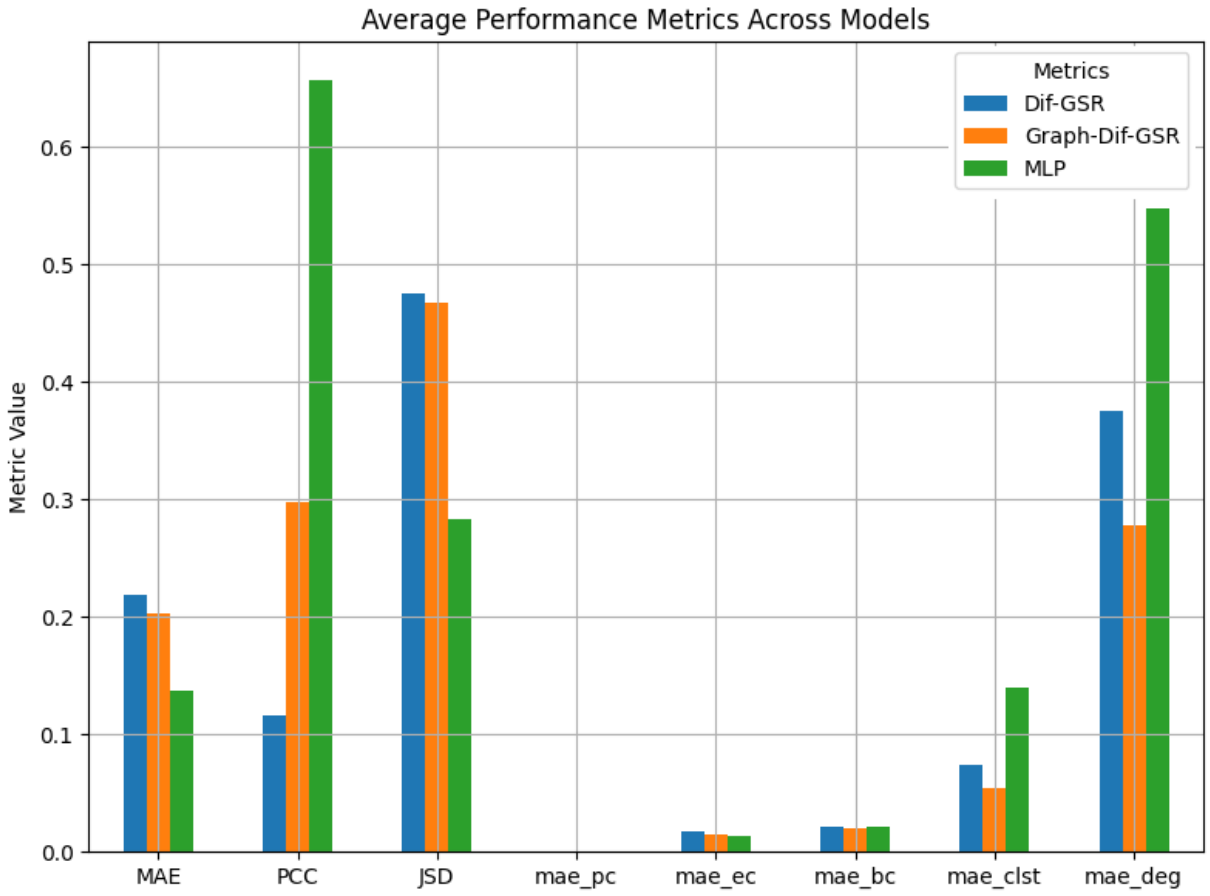

Fig. 7. Overall Comparison of Metrics Across All 3 Models

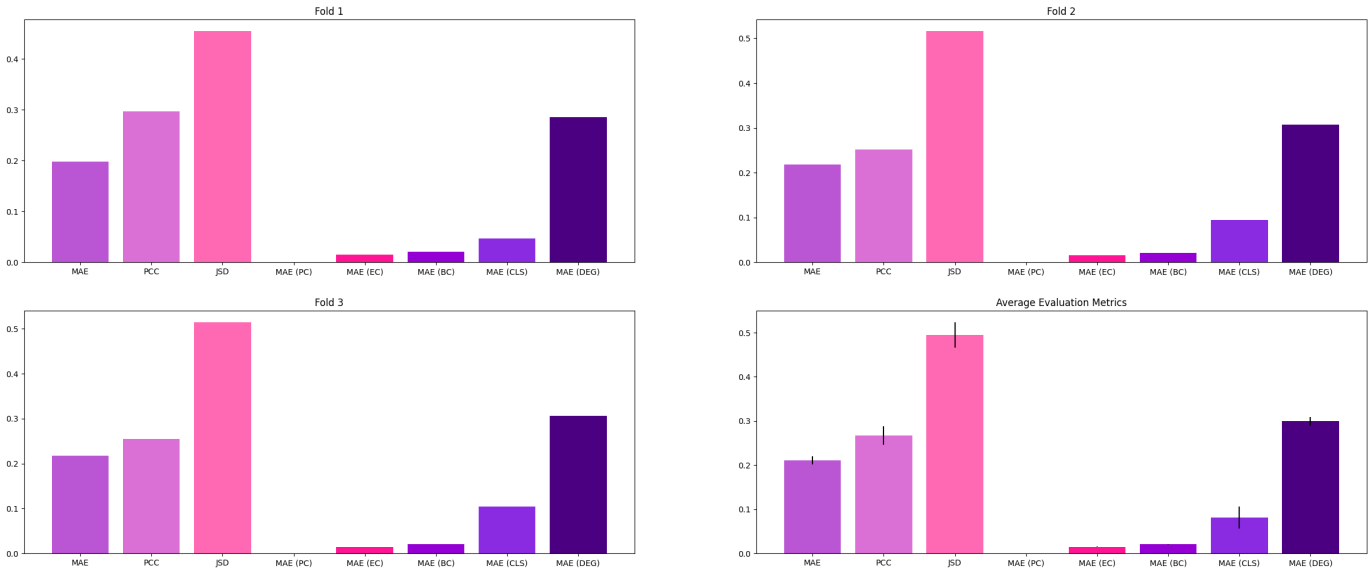

Fig. 8. Graph-Dif-GSR Performance Across All Folds

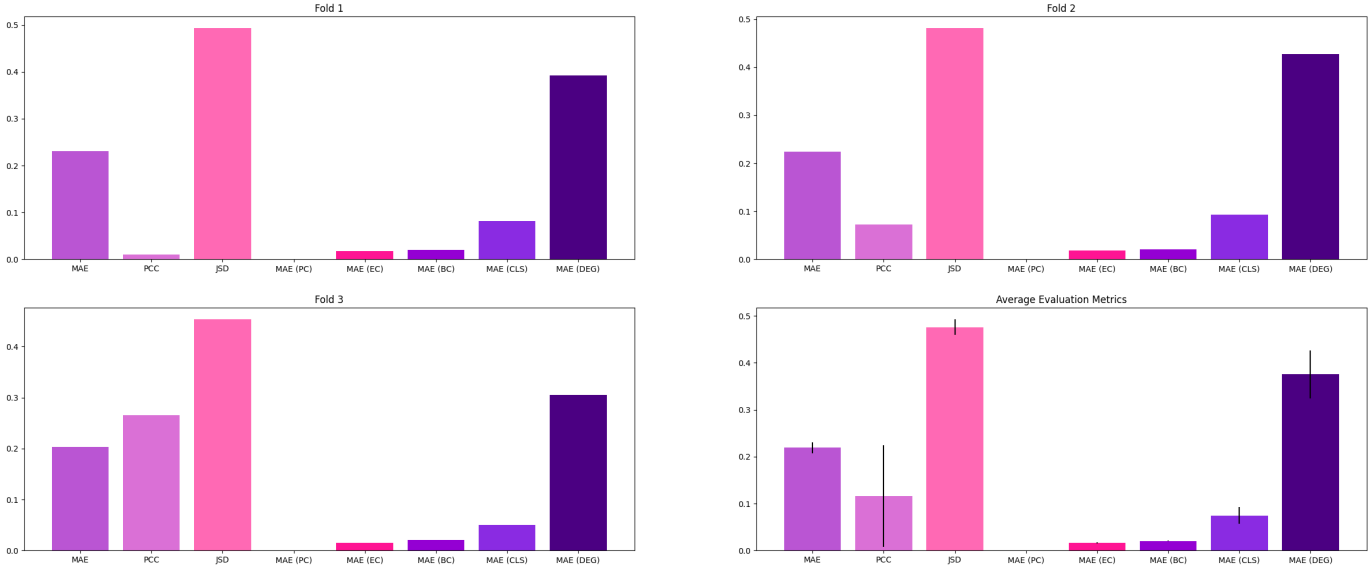

Fig. 9. Dif-GSR Performance Across All Folds

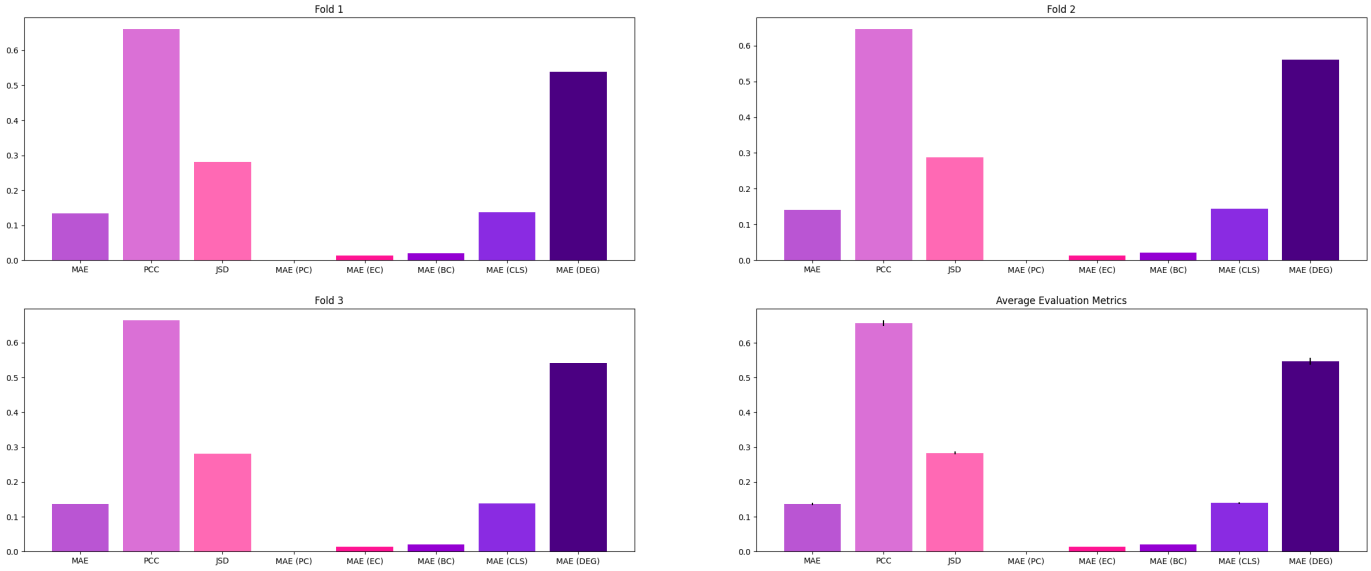

Fig. 10. MLP Performance Across All Folds