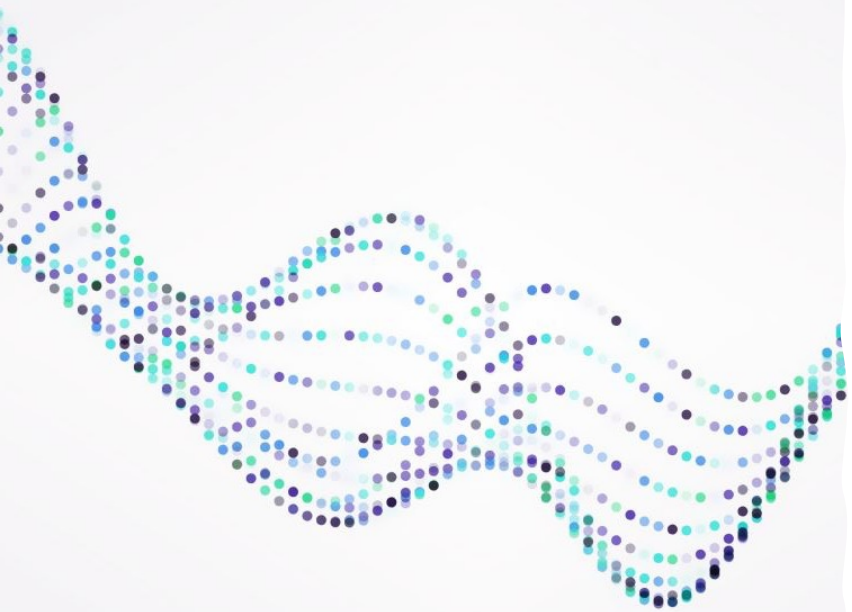


GNN Explainability Methods

Harshit Kumar

Fall Co-op 2023

The Jackson Laboratory

Introduction

Model Explainability Methods

Help you understand and interpret predictions made by machine learning models.

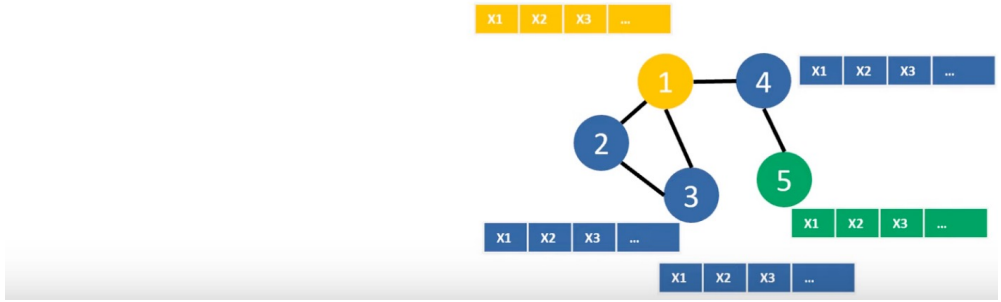
- What is our model learning?
- Which part of input is most important for our predictions?

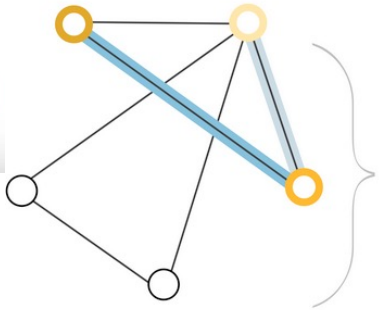
Example use case #1: in image classification, it helps determine which section of the image model is looking at while making predictions.

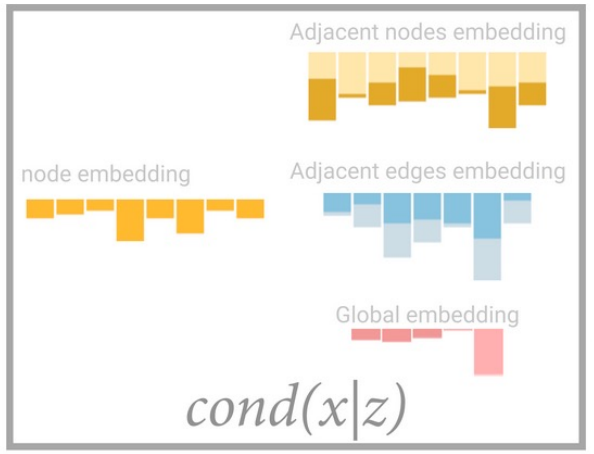
Example use case #2: identifying which genes are relevant for specific biological processes or diseases, thus streamlining lab experiments for novel drug discovery and understanding disease mechanisms.

Graph Neural Networks

$$h_u^{(k+1)} = UPDATE^{(k)}\left(h_u^{(k)}, AGGREGATE^{(k)}(\{h_v^{(k)}, \forall\, v \in \mathcal{N}(u)\})\right)$$







Message passing

update function $f =$, ...

conditioning function =
Concatenation,
Linear Layer and Add,
FiLM Layer..

https://youtu.be/ABCGCf8cJOE?si=Di_Iw5pbOZYv8Jeb
https://distill.pub/2021/gnn-intro/

Schematic for conditioning the information of one node based on three other embeddings (adjacent nodes, adjacent edges, global). This step corresponds to the node operations in the Graph Nets Layer.

Gradient based methods

- **Objective:** the primary goal of gradient-based methods is to quantify how sensitive the model's prediction is to changes in input features.

- **How?** Compute gradients (derivatives) of the model's prediction with respect to the input features

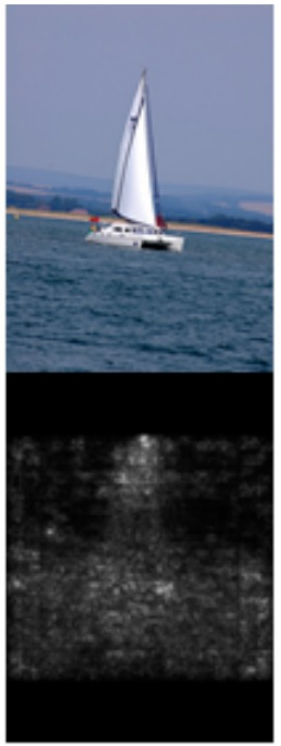
Saliency Maps

Saliency score of feature =

gradient of output wrt input * value(input feature)

$$x_i \cdot \frac{\partial S_c(x)}{\partial x_i}$$

where Sc(x) is class score of class c for input x



SmoothGrad: smoothing noisy gradients by taking local gradient average

$$\hat{M}_c(x) = \frac{1}{n} \sum_{1}^{n} M_c(x + \mathcal{N}(0, \sigma^2))$$

where Mc(x) is gradient at x

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps https://arxiv.org/pdf/1312.6034.pdf

Integrated Gradients

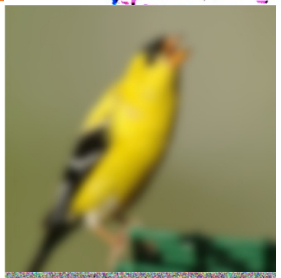
Integrated gradient of i[th] feature:

$$\phi_i^{IG}(f, x, x') = \overbrace{(x_i - x'_i)}^{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^{1}}_{\text{From baseline to input...}} \overbrace{\frac{\delta f(x' + \alpha(x - x'))}{\delta x_i} d\alpha}^{\text{...accumulate local gradients}}$$

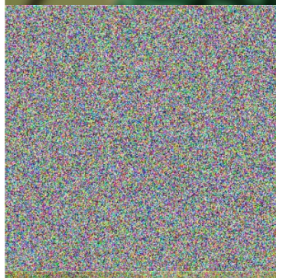
Baselines

max L1 distance

blur



uniform



gaussian

Axiomatic Attribution for Deep Networks https://arxiv.org/pdf/1703.01365.pdf

Shapely Additive exPlanations (SHAP)

SHAP assigns each feature an importance value for a particular prediction

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right].$$

A Unified Approach to Interpreting Model Predictions https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

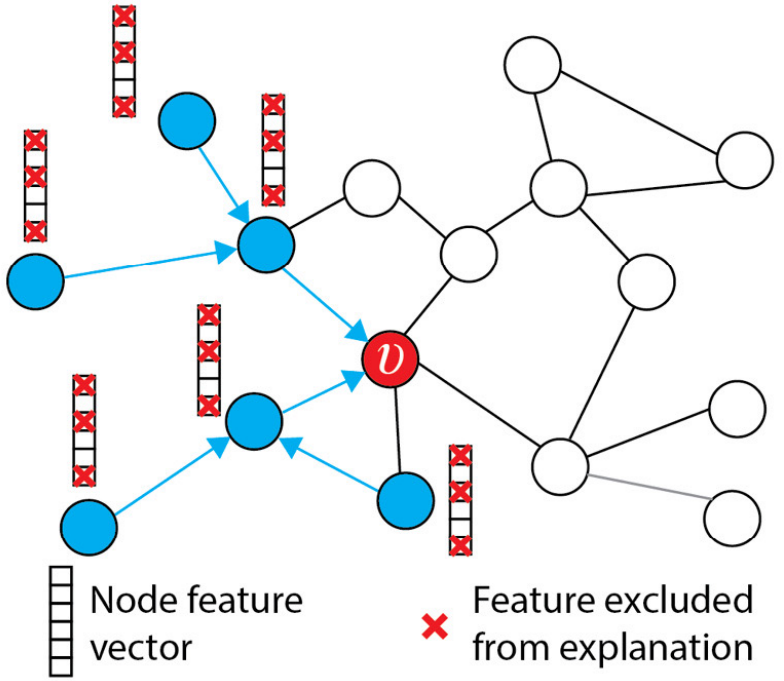
GNNExplainer

GNNExplainer takes a trained GNN and its prediction, and it returns an explanation in the form of a small subgraph Gs of the input graph together with a small subset of node features Xs that are most influential for the prediction.

$$\max_{Gs} MI\left(Y, (Gs, Xs)\right) = H(Y) - H(Y|G = Gs, X = Xs)$$

Relaxation: overall optimization problem is reduced to constructing an *edge* M_E *and feature* M_F *masks* that finds a subgraph that maximizes the mutual information.

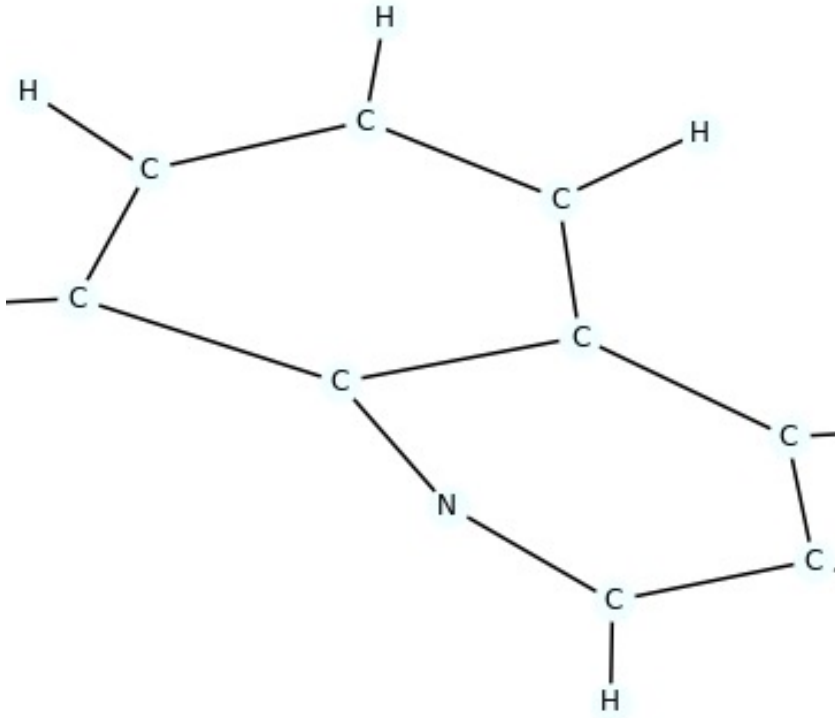
A' = A . sigma(M_E)
X' = X . M_F



GNNExplainer: Generating Explanations for Graph Neural Networks https://arxiv.org/pdf/1903.03894.pdf

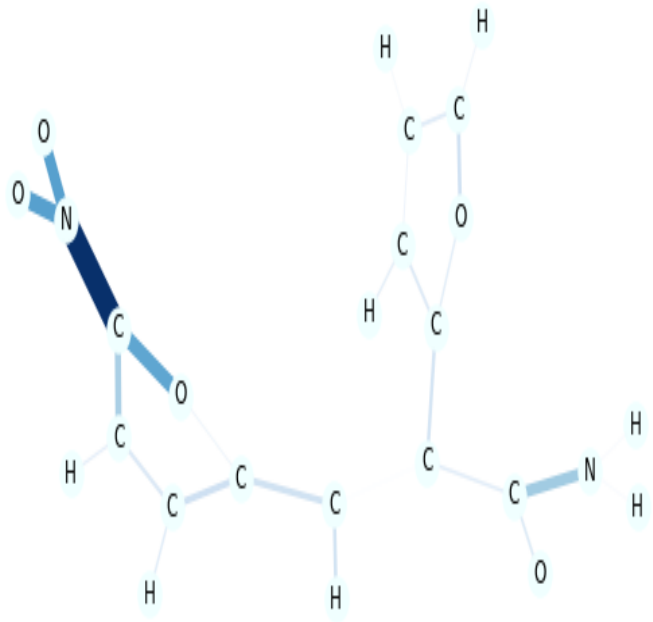
Example: Mutagenicity Dataset

- Mutagenicity dataset from TUDatasets.
- Used for the task of predicting whether chemical compounds or substances have the potential to cause mutations in DNA.
- Data categorized into two classes: mutagen and non-mutagen.
- Consists of 4337 molecule graphs where the task is to predict the molecule mutagenicity.

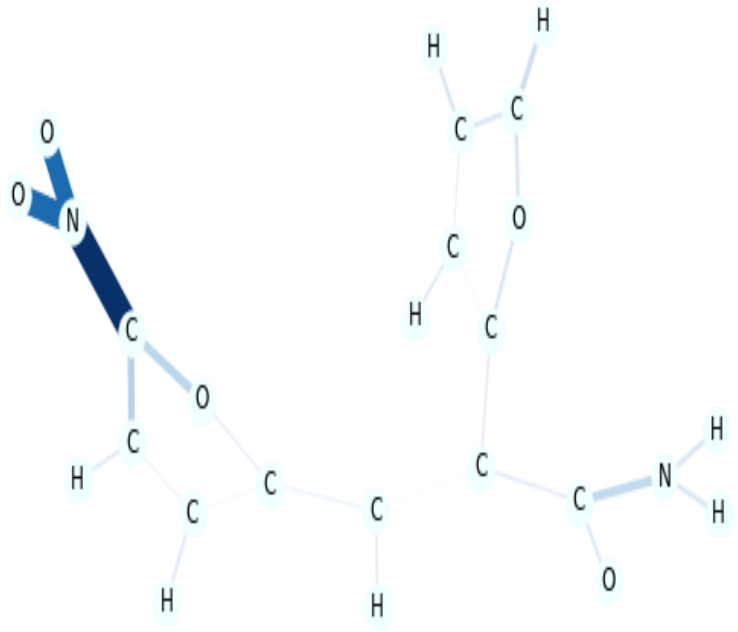


Example: Mutagenicity Dataset Results

Saliency



Integrated Gradients



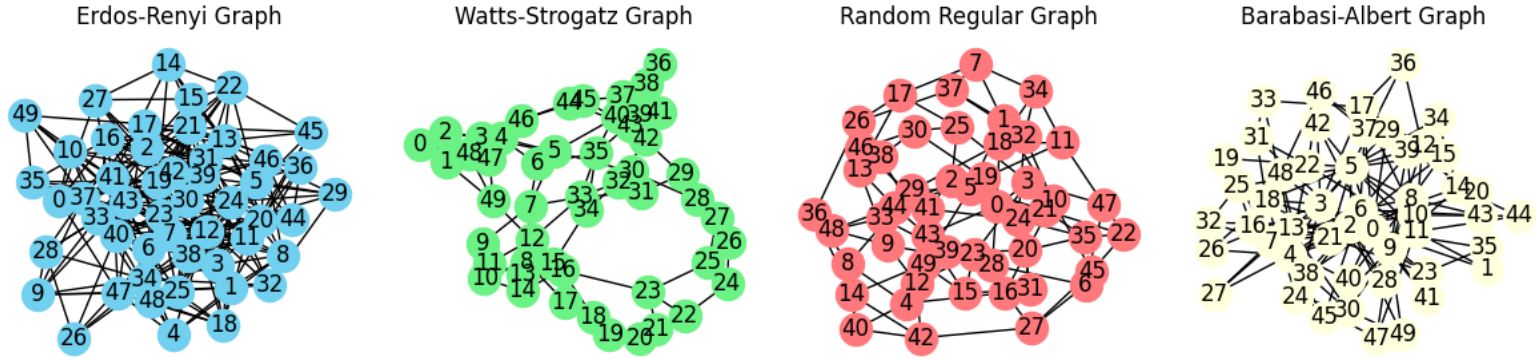
Synthetic Graph Dataset Generation

Goal: To generate graph dataset with motifs

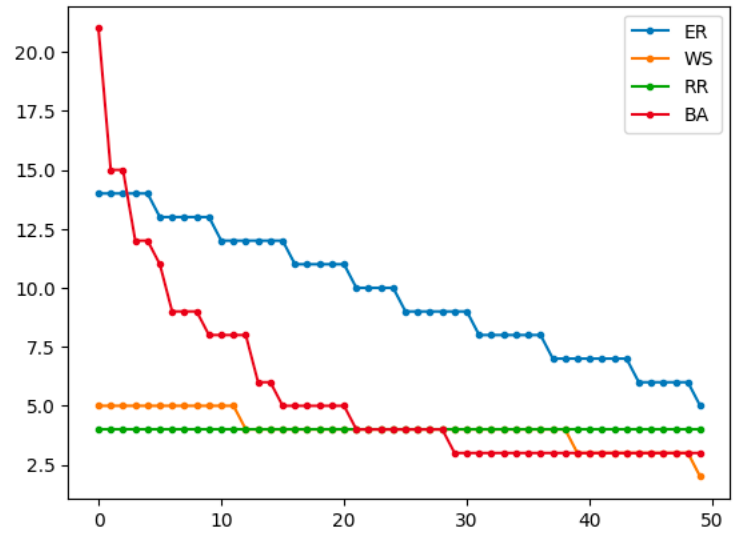
Random Graph generation models:

- ***Erdos-Renyi:*** a graph is constructed by connecting nodes randomly with given probability.
  - Drawbacks: 1. low clustering coefficient 2. not scale-free network – power law (no formation of hubs)
- ***Watts-Strogatz:*** *solves drawback #1*; start with a ring-lattice then randomly pick some links to rewire; produces small-world network; small-world network
- ***Barabasi-Albert:*** *solves drawback #2*; start with a small network and add a new node by drawing a fixed given number of links to existing nodes with preferential attachment i.e. the probability of linking to existing node is proportional to that node's degree.

Properties of Graphs generated by Random models

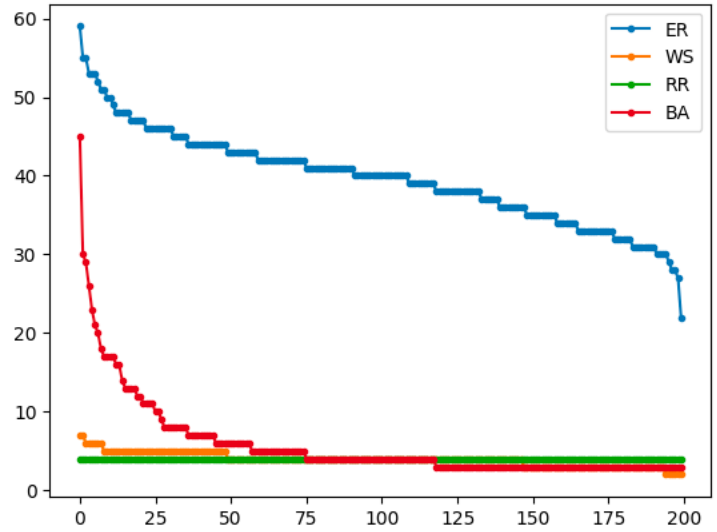


Nodes: 20

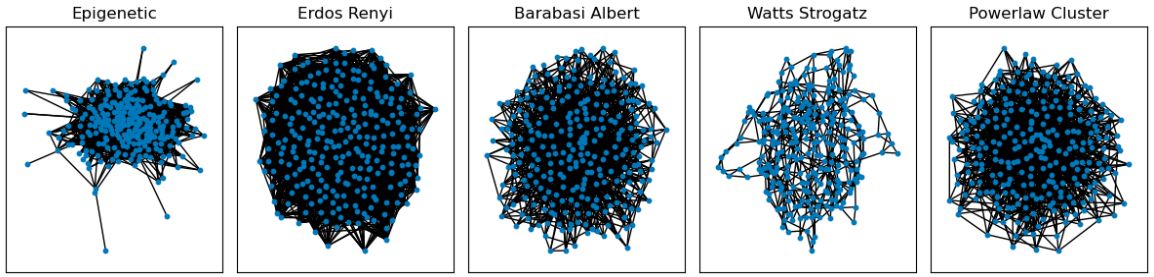


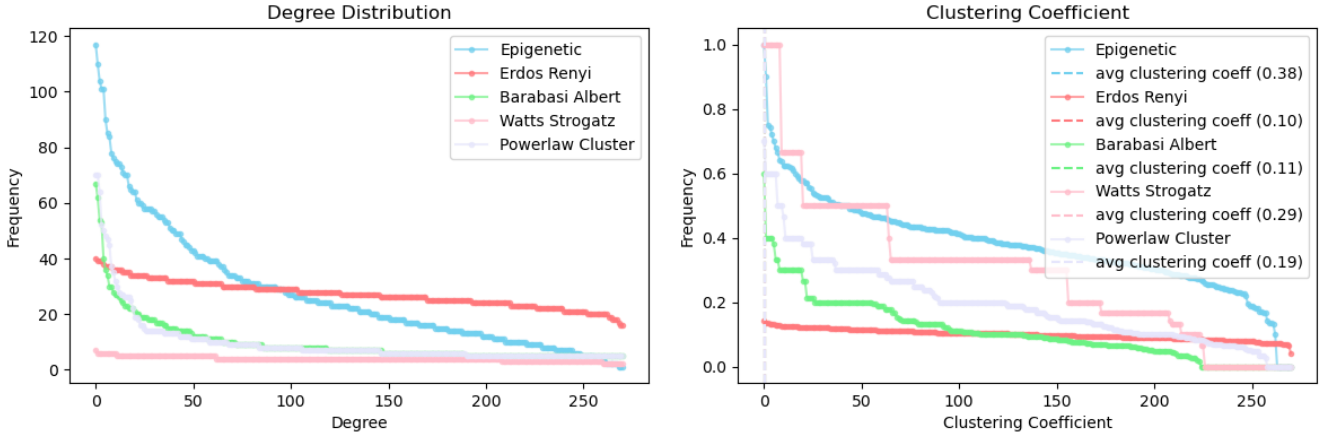
Degree distribution

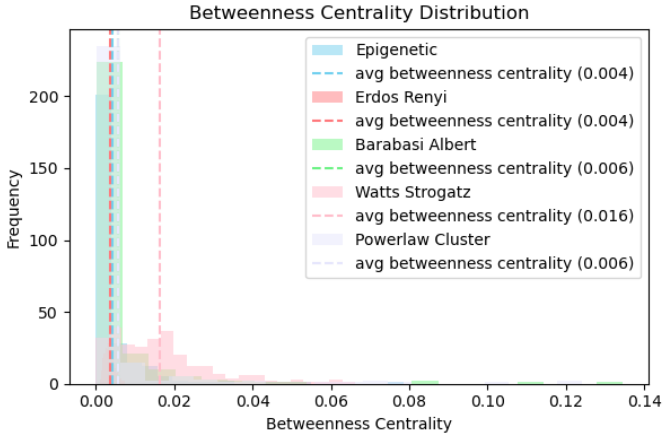
Nodes: 200

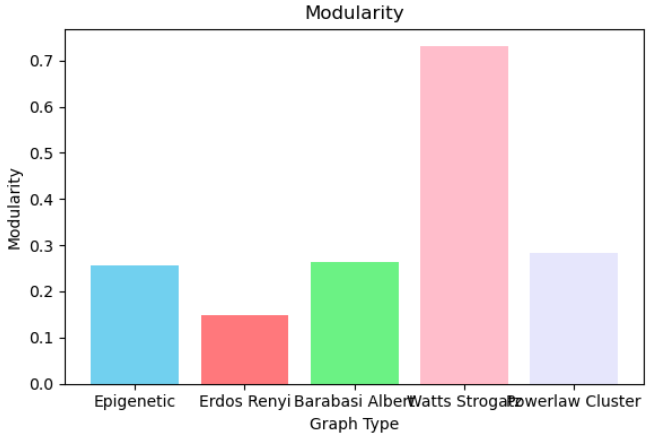


AD Biodomain Graph Networks Analysis

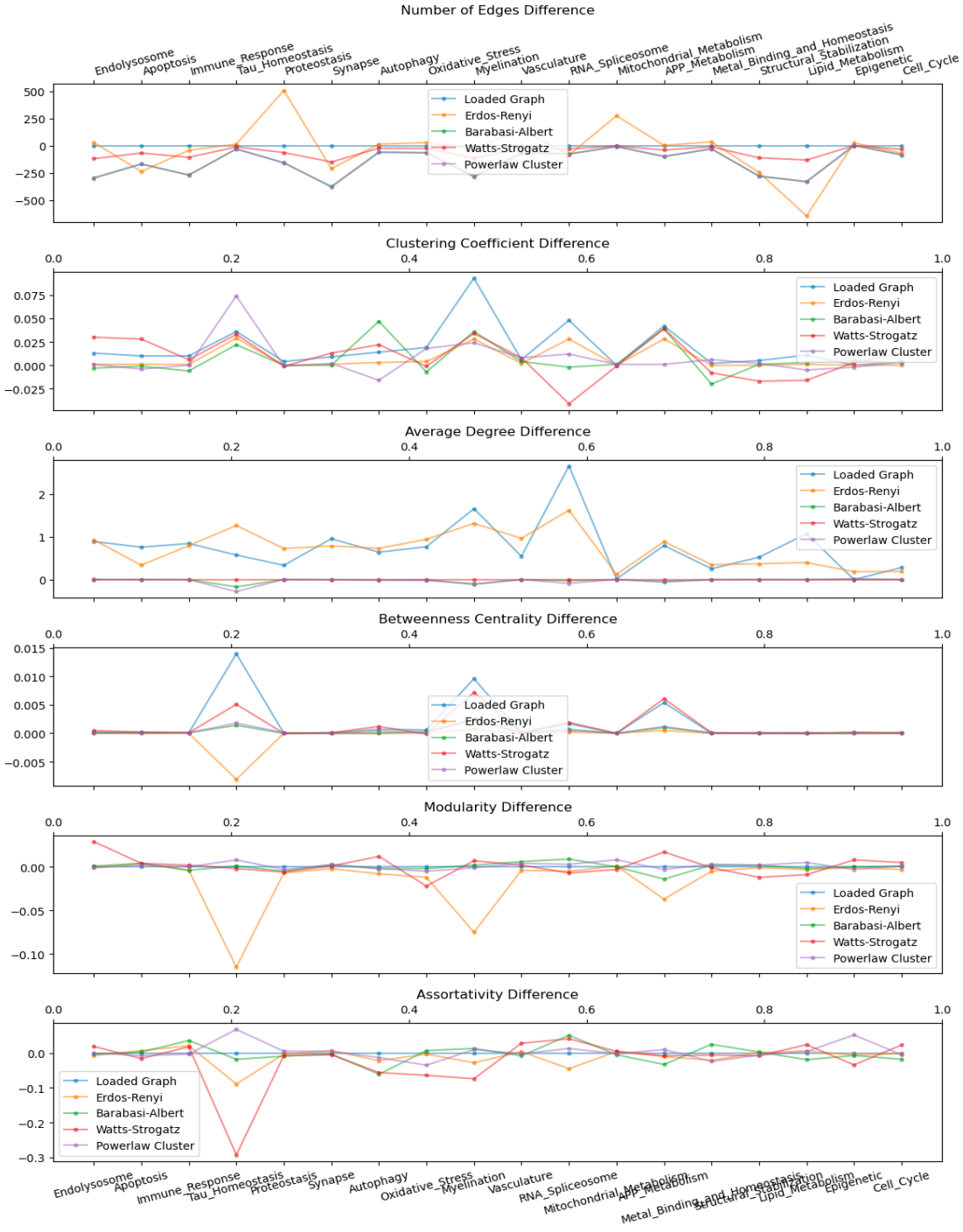








Properties of AD Biodomain Networks and Comparison with Random Models

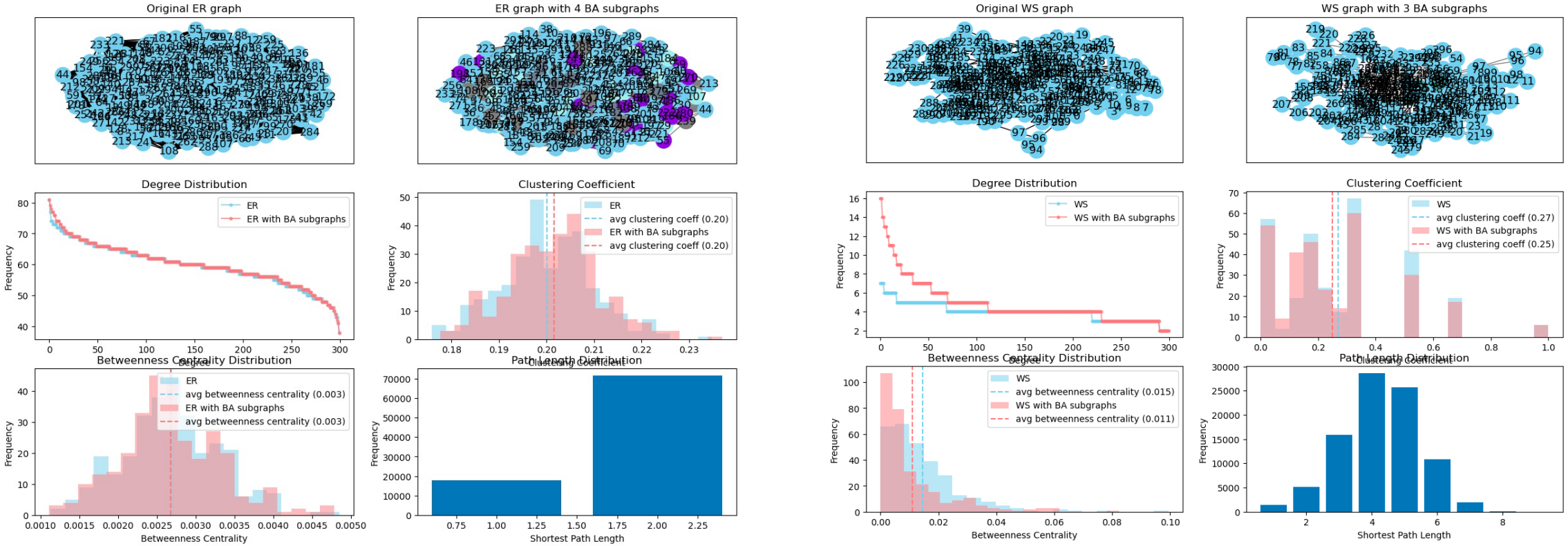


Assessing performance of interpretability methods

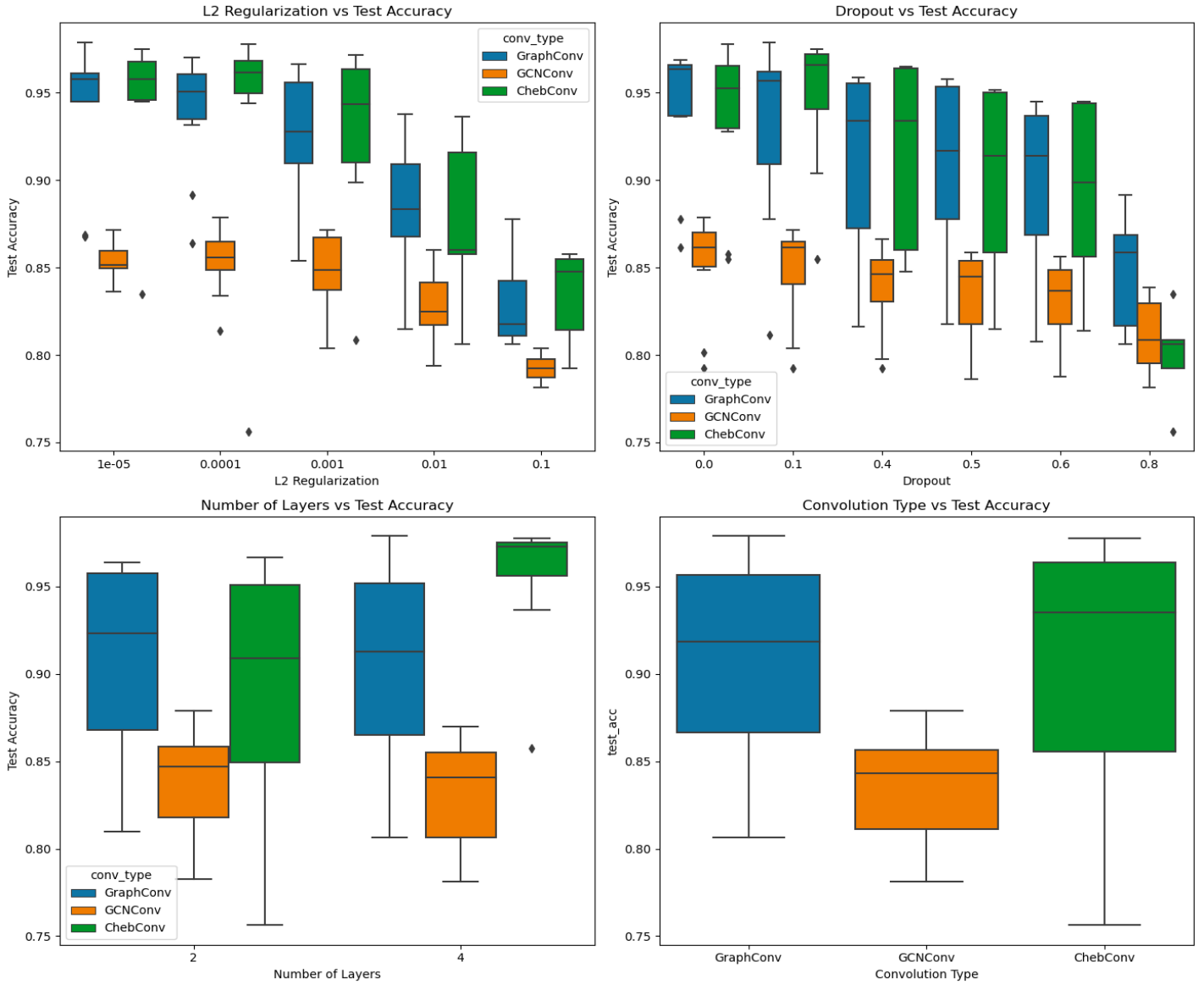
- Generate graph dataset for graph classification with ground truth explanations
  - random model type, graph sample size
- Test (grid search) different GNN model configurations
  - type of graph convolution , l2, dropout regularization
- Test different interpretability methods
  - Integrated Gradients, SHAP, GNNExplainer
- Metrics for prediction and interpretability
  - predictive accuracy
  - interpretability: cross-entropy, AUC, correlation

Graph Generation with Motifs

Create a large sparse graph using a specified base graph model and insert motifs by replacing a specified number of nodes with a specified graph model.



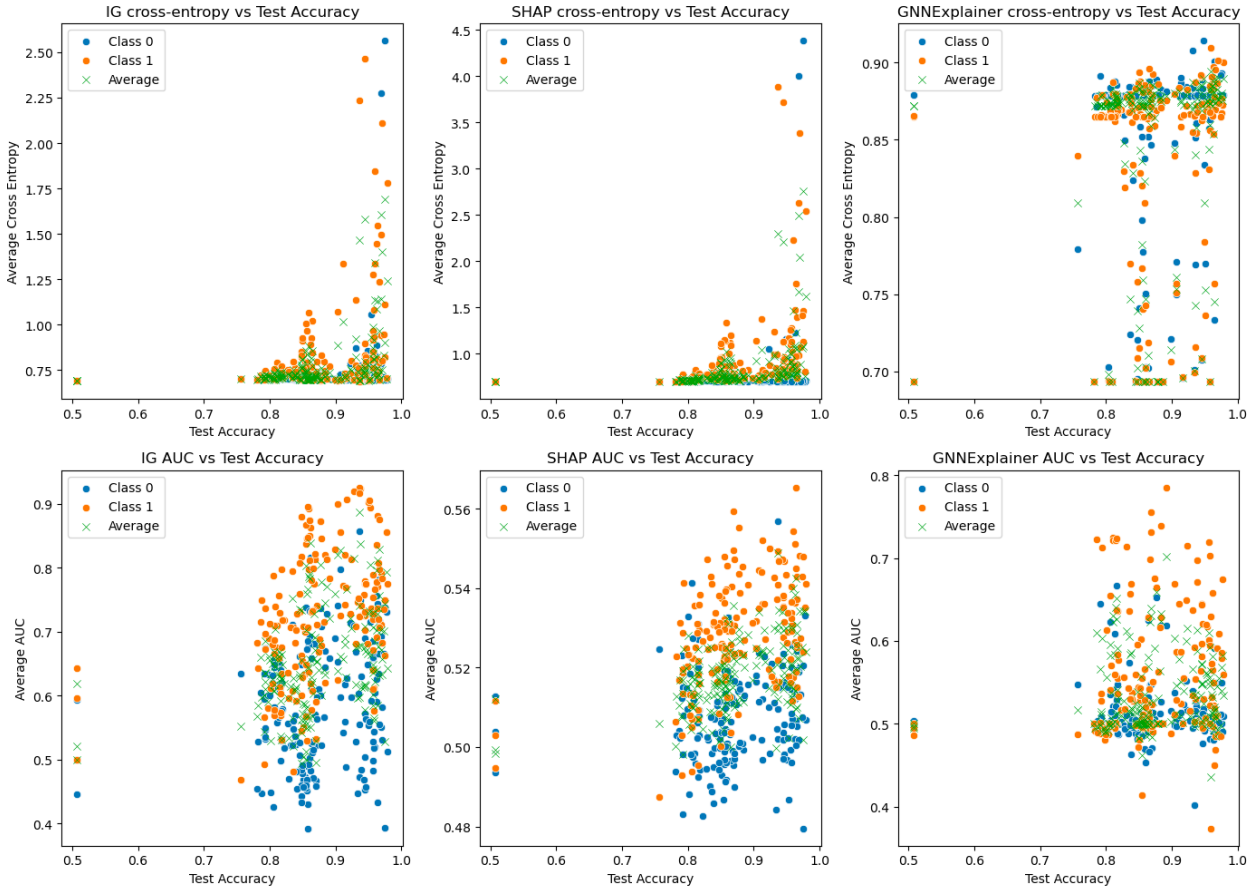
Regularization degrades GNN predictive performance



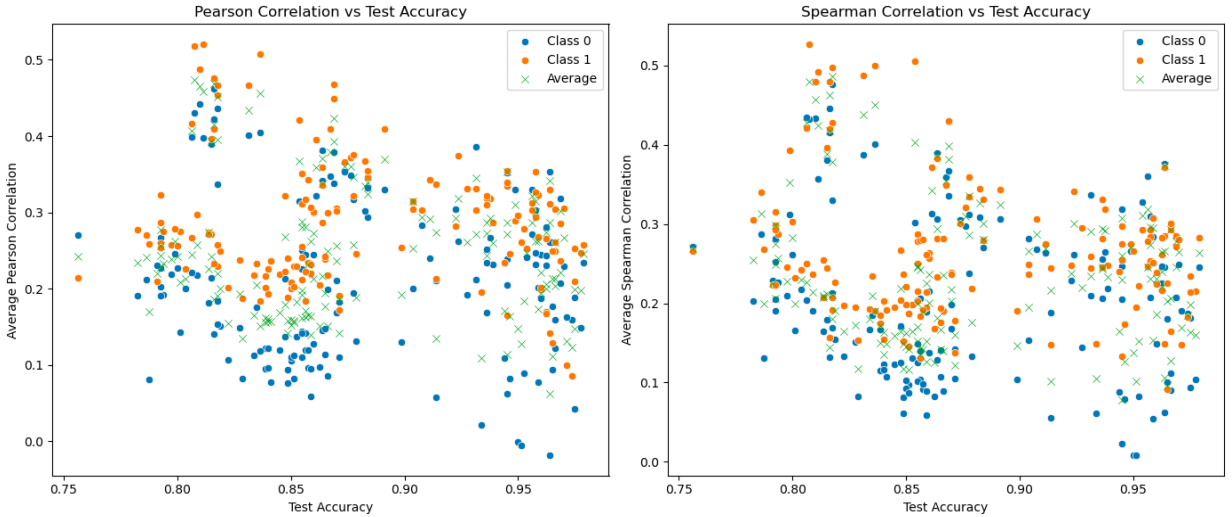
Dataset: 8000 samples
Train: 5600 {0: 2840, 1: 2760}
Valid: 1600 {0: 774, 1: 826}
Test: 800 {0: 416, 1: 384}

IG gives better explainability performance, and No correlation b/w accuracy and explainability

Explainability performance Vs Accuracy

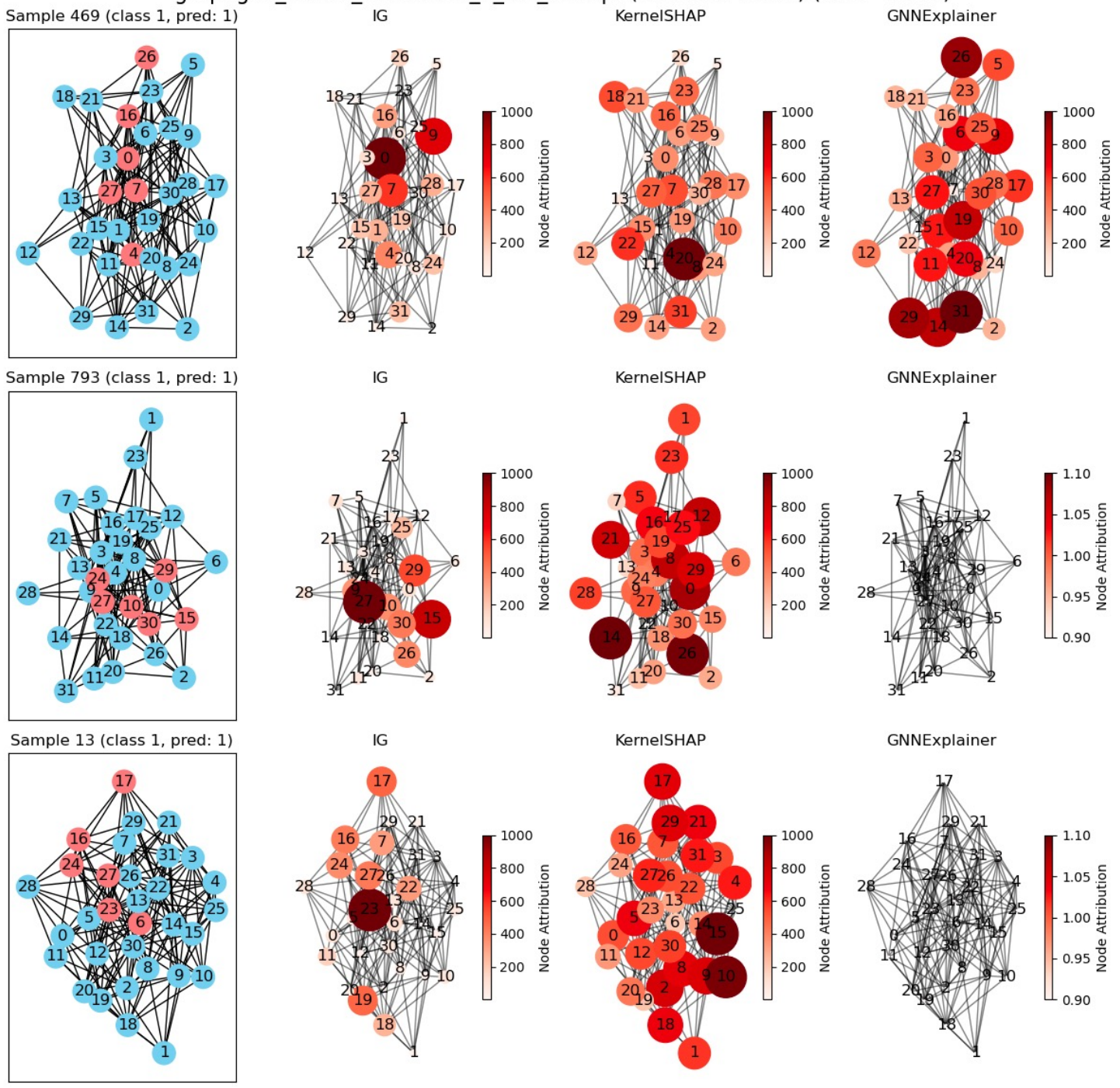


Correlation b/w node importance and node degree Vs Accuracy



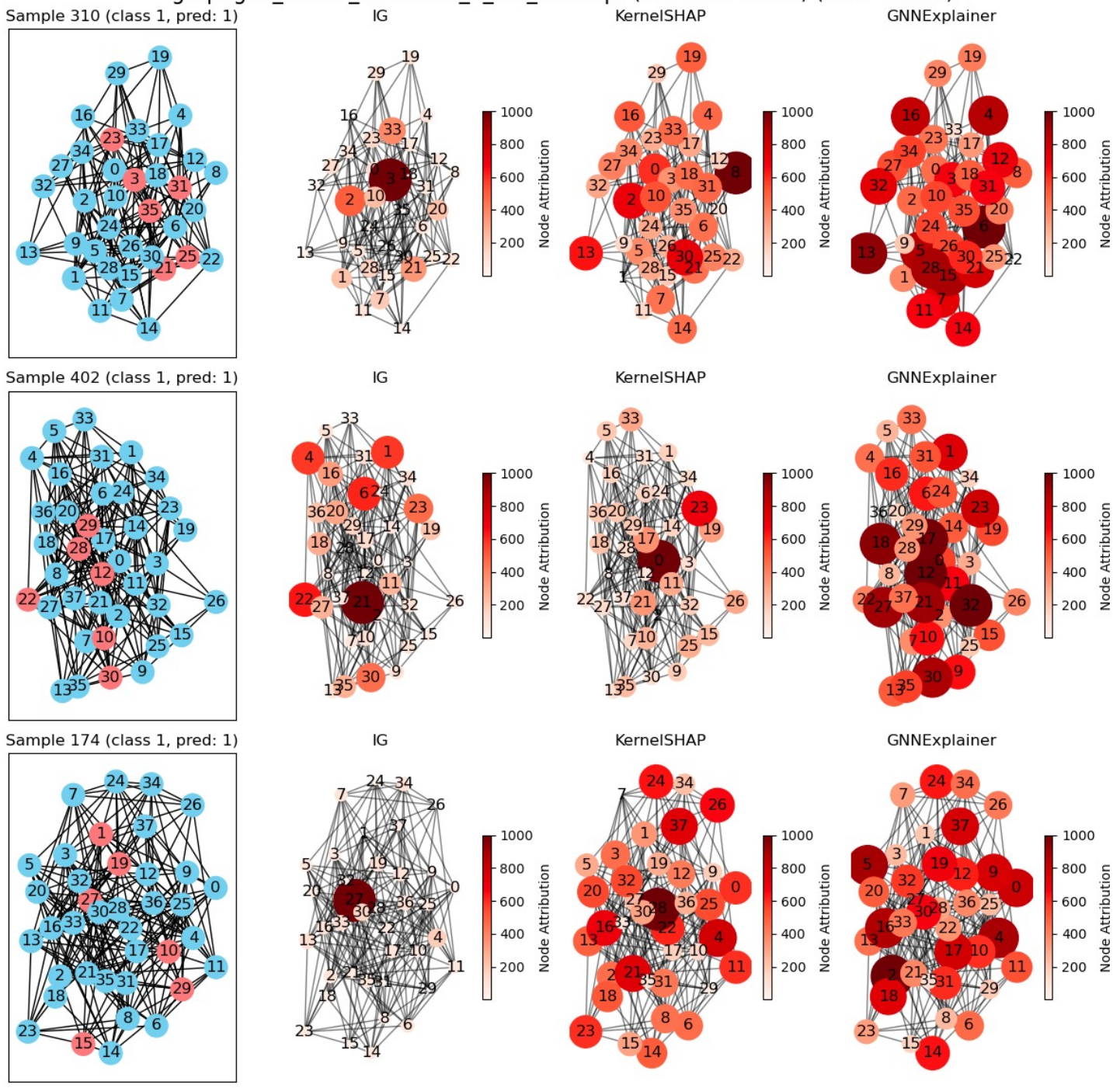
Integrated Gradients outperforms other methods (HIGH predictive performance model)

Model: ChebConv
No of layers: 4
Dropout: 0.0
L2: 0.01

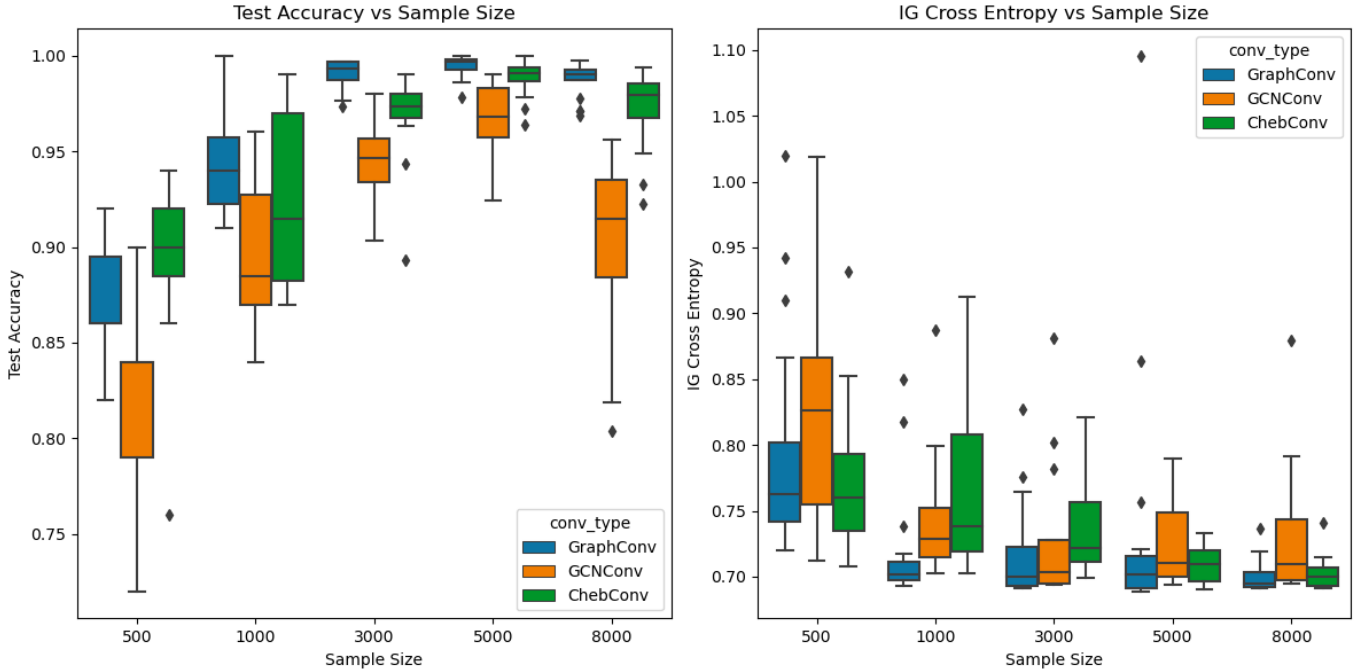


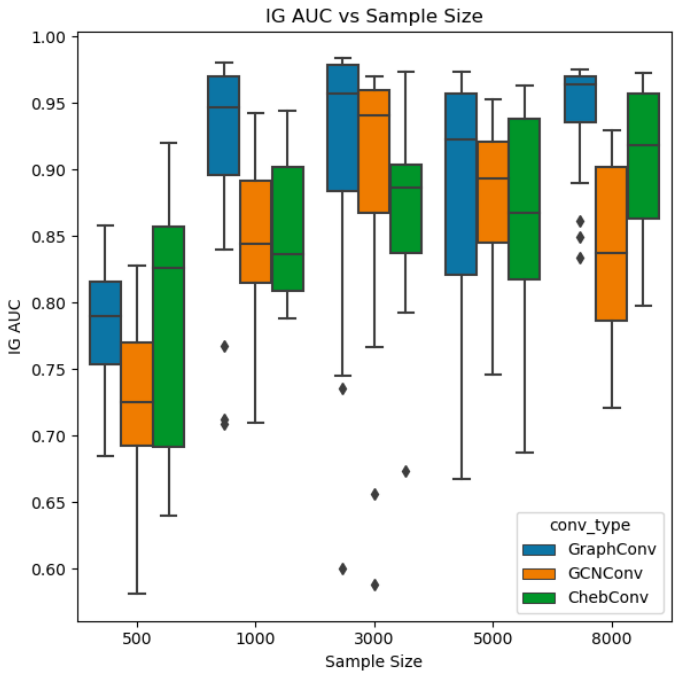
None of the Explainability methods work (LOW predictive performance model)

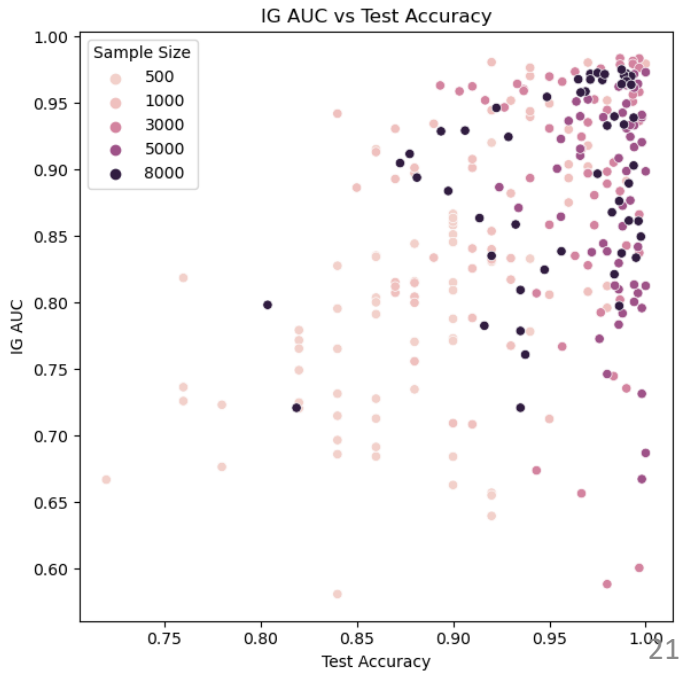
Model: GCNConv
No of layers: 4
Dropout: 0.0
L2: 0.001



Explainability performance doesn't consistently increase with sample size

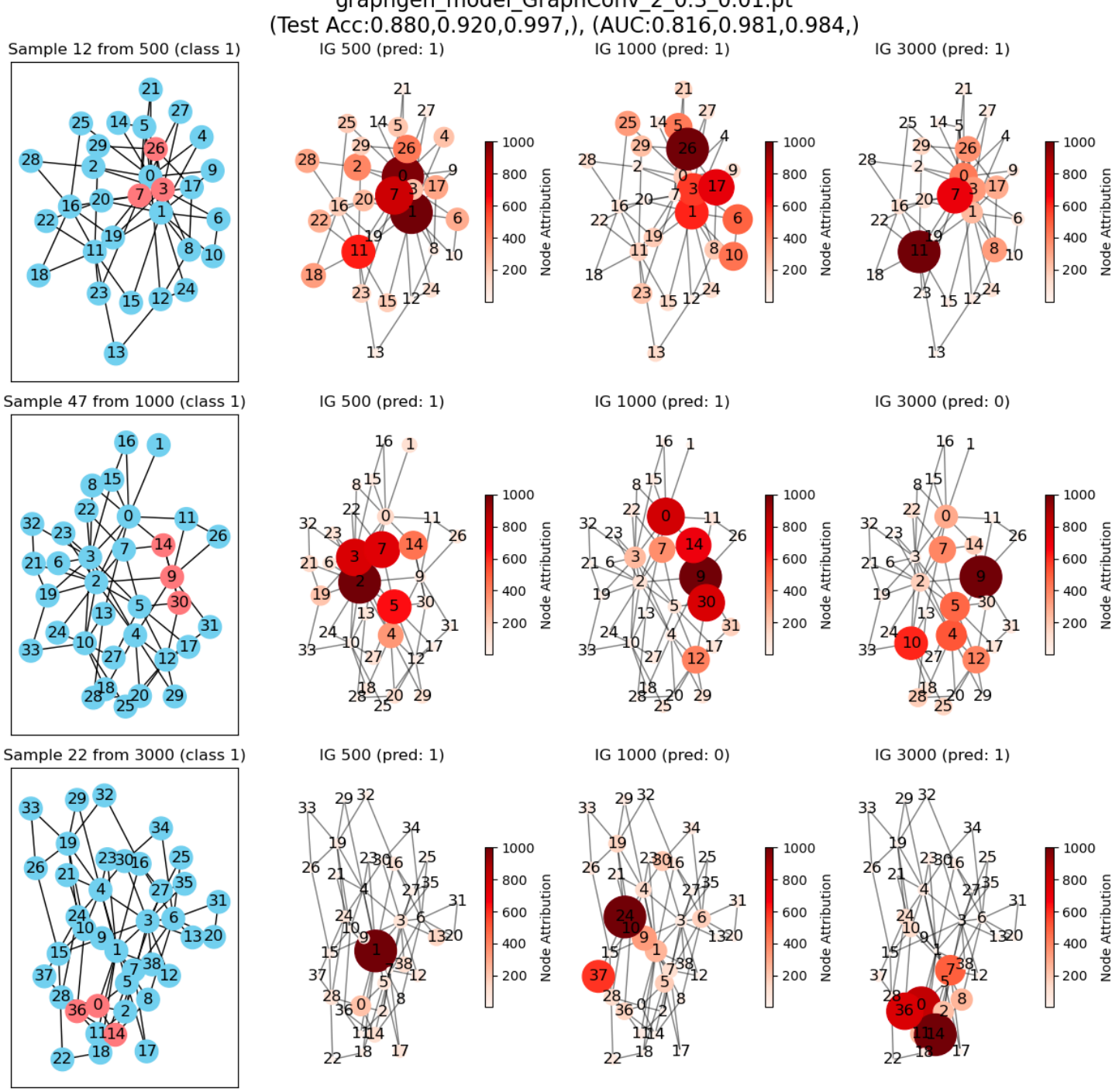




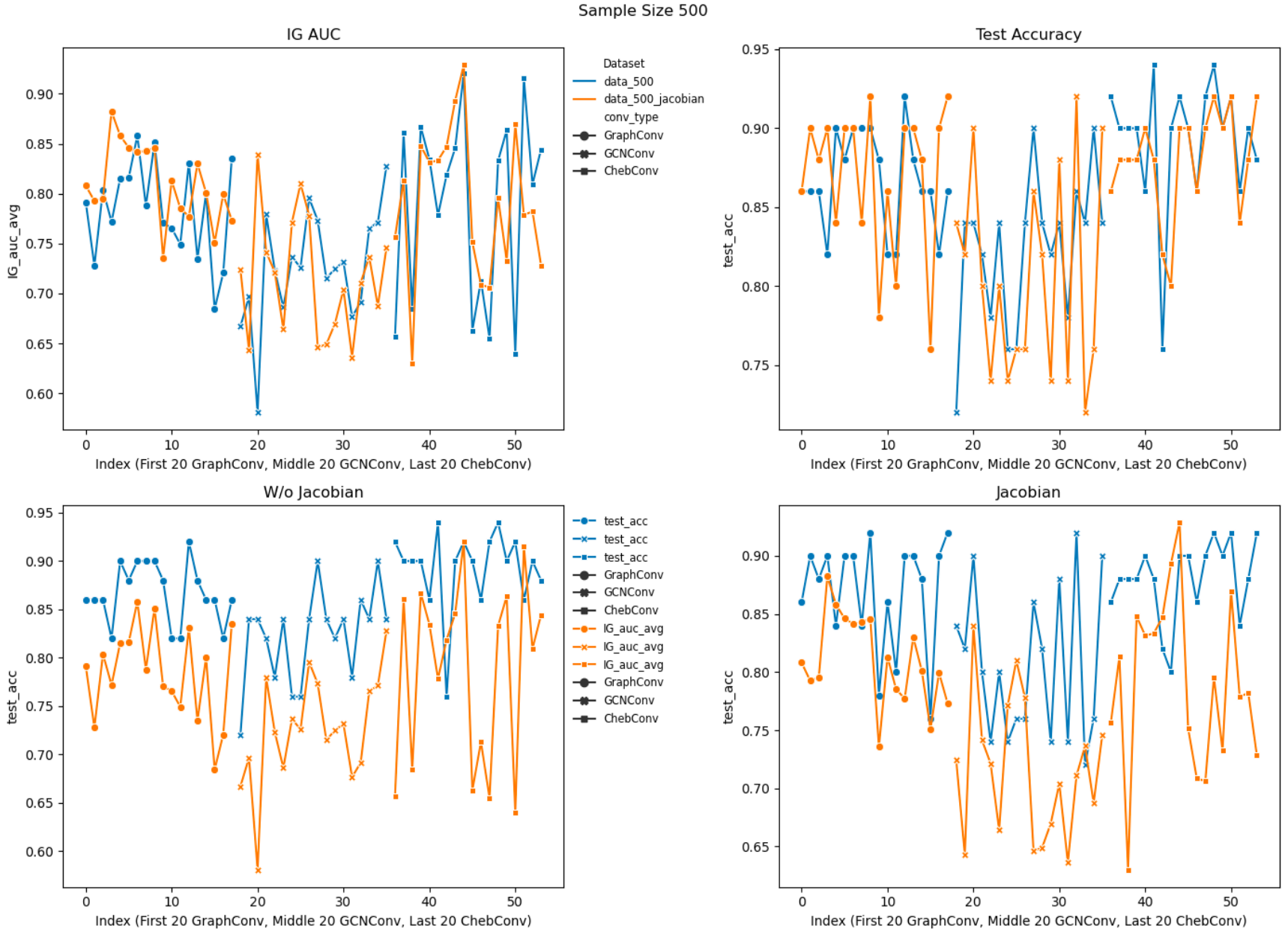


Explainability methods (Different sample sizes)

Model: GraphConv
No of layers: 2
Dropout: 0.3
L2: 0.01



Jacobian regularization doesn't improve explainability performance



Summary & Future Work

- Need of model explainability methods
- Discussed attribution approaches (gradient based: Saliency maps, Integrated gradients), KernelSHAP, GNNExplainer.
- Generated synthetic dataset with motifs.
- Assessed performance of interpretability methods on synthetic datasets.
- Test Expected Gradient attribution (averaging over multiple baselines).
- Experiment with different forms of attribution priors, such as graph topology-based or node feature-based priors, to guide the attribution process more effectively.