Reconstructing Social Network Structure with Limited Information

數據所碩一
RE6124035 黃亮臻
2024-01-04

影片連結:
https://drive.google.com/file/d/1cM_YUsf2Dxy8EZbxHK7GBSp6fB0g8B3w/view?usp=sharing

Introduction

- **Background & Motivation:**

  This study explores the possibility of using limited user information to reconstruct social network structures, thereby assessing the risk of privacy leakage.

- **Problem Statement:**

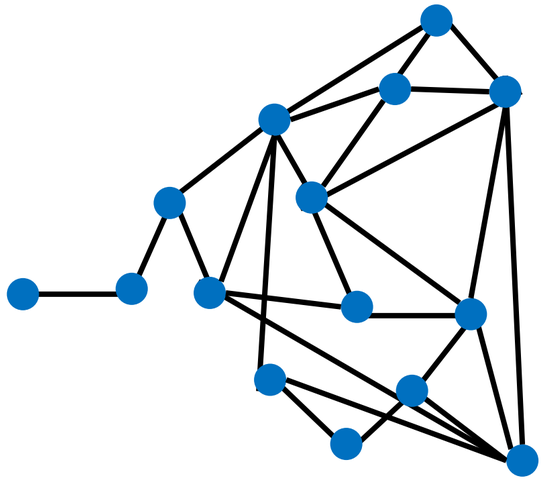
  Here, subgraphs generated by GNNExplainer are used as limited yet representative user information. The aim is to see to what extent reconstruction can be achieved with a small amount of explanatory subgraph data.

- **Challenges:**

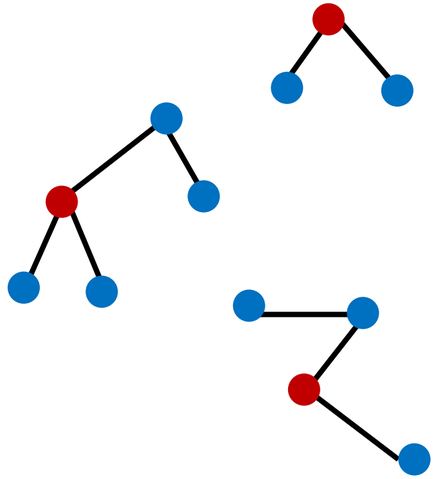
  The main challenge lies in predicting numerous unknown relationships from limited information, which often amounts to several times the known connections.

Main Objective

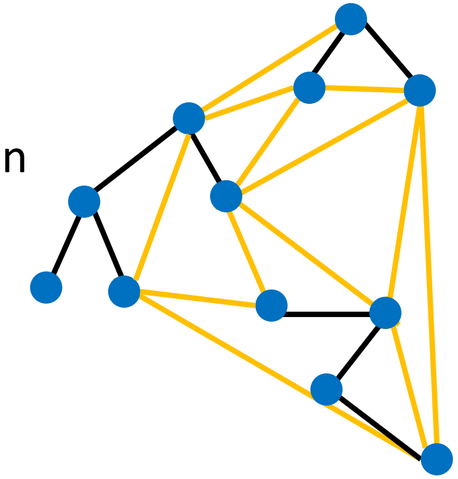
Original Graph



K Subgraphs

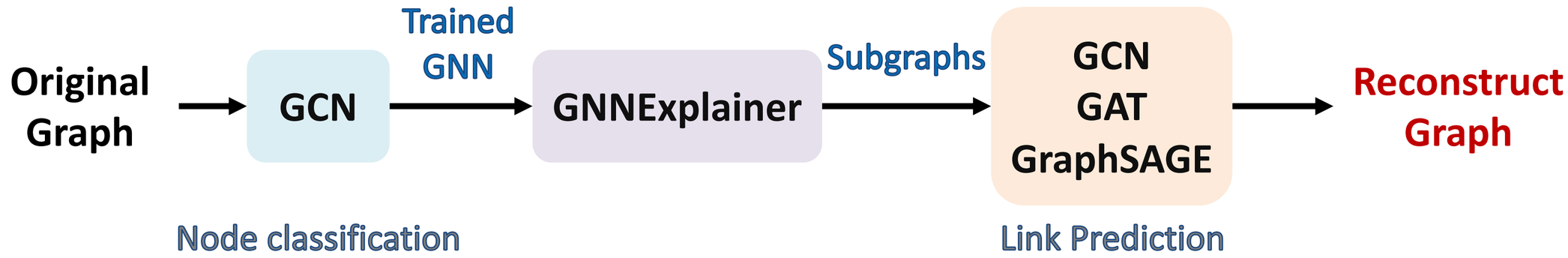


Link Prediction



Overview

- Stage1: Train the base GNN model.
- Stage2: Use GNNExplainer to obtain subgraph explanations.
- Stage3: Attempt to reconstruct the original graph using the subgraphs from GNNExplainer.



Related Work

- **GNNExplainer** [1] follows the idea in perturbation/casual-based methods and learns soft masks that cover the key nodes and edges while maintaining the original prediction score. This method tends to provide a sufficient subgraph explaination.
- **CF2** [2] formulating an optimization problem based on the causal inference theory's insights of Factual and Counterfactual extracts explanations that are both sufficient and necessary.
- **MixupExplainer** [3] identifies a distribution shifting issue in graph neural network explanations and uses augmentation to align the distribution closely with the original graph's space.
- **EGNN** [4] learns to predict edge-labels, facilitating explicit clustering by iteratively refining edge labels based on intra-cluster similarity and inter-cluster dissimilarity.
- **EMPIRE** [5] proposes a novel edge splitting method for specific usage of each edge and a negative sampling strategy targeting 'hard' negatives, significantly enhancing performance.
- **GRAPE** [6] the missing data problem using a graph representation, where the observations and features are viewed as two types of nodes in a bipartite graph, and the observed feature values as edges.

Notations

- Given a graph: $\mathrm{G} = \{\mathrm{V}, \mathrm{E}\}$
- Let $\mathrm{S} = \{\mathrm{S}_1, \mathrm{S}_1, \ldots \mathrm{S}_\mathrm{k}\}$ be the set of subgraphs generated by Explainer, where each S_i is a smaller graph representing portion of G. In this research, k was chosen to be 3.
- Each S_i is defined as $\mathrm{S}_\mathrm{i} = (\mathrm{V}_\mathrm{i}, \mathrm{E}_\mathrm{i})$, where $\mathrm{V}_\mathrm{i} \subseteq \mathrm{V}$ and $\mathrm{E}_\mathrm{i} \subseteq \mathrm{E}$.
- Define $G' = (V', E')$ as the collection of all nodes in the subgraphs and the edges that should exist between these nodes in the original graph G. Where:
  - $V' = \bigcup_{i=1}^{n} \mathrm{V}_\mathrm{i}$
  - $E' \subseteq \mathrm{E}$, including edges shown in the subgraphs and edges between V' in G that are not displayed in the subgraphs.

Problem Formulation

- **Problem:**

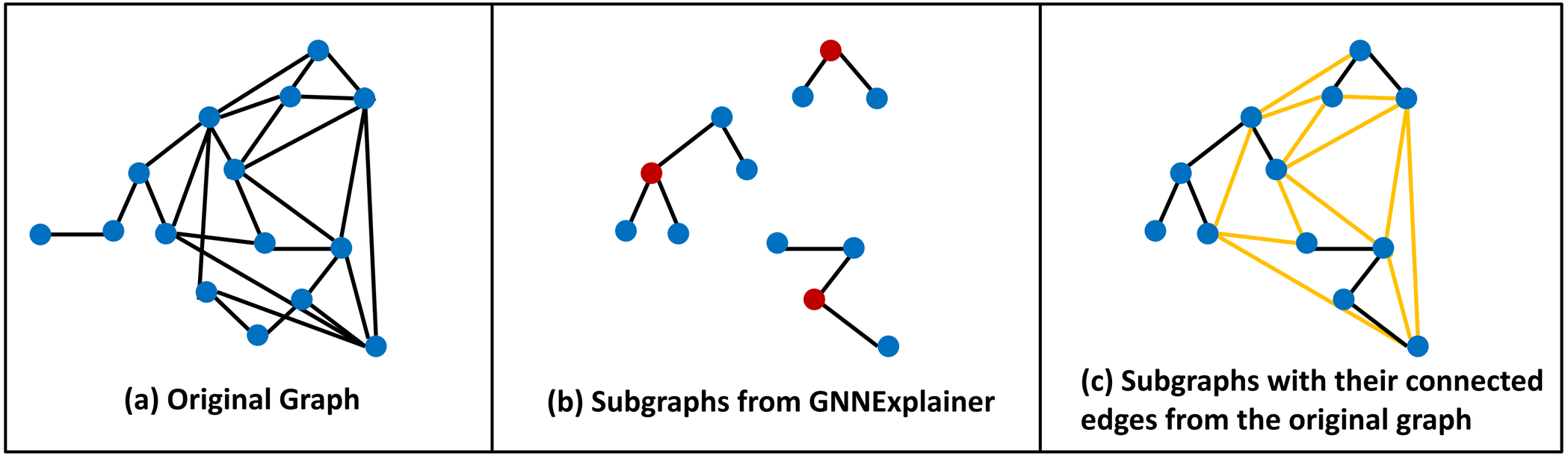
  Predicting $E' - \bigcup_{i=1}^{n} \mathrm{E_i}$, which are the edges that are not shown in the subgraphs but should exist in the original graph.

- **Objective:**

  Reconstruct the structure of the graph G'as accurately as possible using the limited subgraph information S.

Dataset Synthesis Using GNNExplainer

1. Select 3 nodes for explanation, then GNNExplainer will generate 3 explanatory subgraphs.
2. Verification of path existence between all nodes in subgraphs within the original graph.
3. Combine subgraphs with their connected edges from the original graph into a large graph.
4. Create 140 synthesized graphs, 100 for training, 20 for validation and 20 for testing.

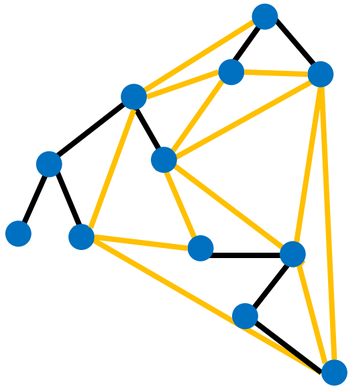
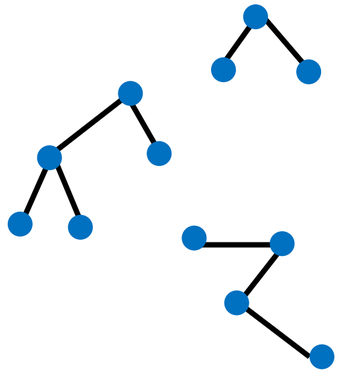


(a) Original Graph

(b) Subgraphs from GNNExplainer

(c) Subgraphs with their connected edges from the original graph

Dataset Synthesis Using GNNExplainer

	Input	Calculate Loss / ACC, AUC…
Training (100 graphs)		Positive edges — + Negative edges (Randomly generated in each batch)
Valid/ Testing (20 graphs, each)		Positive edges — + Negative edges (Fixed)

Original Dataset

Facebook

Dataset	FacebookPagePage
# Graph	1
# Nodes	22470
# Edges	342004
# Features	128
# Classes	4
Avg node degree	15.22
Is undirected	True

Github

Dataset	Github
# Graph	1
# Nodes	37700
# Edges	578006
# Features	128
# Classes	2
Avg node degree	15.33
Is undirected	True

Synthesized Dataset

	Facebook			Github		
	Train	Valid	Test	Train	Valid	Test
Number of graphs	100	20	20	100	20	20
Average number of nodes	6015	5807	5690	11662	12074	8805
Average number of given edges	141732	38906	38655	245029	19889	15513
Average number of edges to predict	197783	190120	197000	453645	468803	345869

Facebook: Predicted edges ≈ 5 * known edges

GitHub: Predicted edges ≈ 22 * known edges

Experiment Setup

Stage 1: Training the base GNN model

	Facebook	**Github**
Model	GCNConv(128, 64) GCNConv(64, 32) GCNConv(32, 4)	GCNConv(128, 64) GCNConv(64, 2)
Learning rate	0.001	0.03
Optimizer	Adam	Adam
Epoch	1000	1000
Accuracy	91.10%	86.55%

Stage 2: GNNExplainer

	Facebook	**Github**
num_hops	3	2

Stage 3: Link prediction

	Facebook	**Github**
Model	GCN, GAT, GraphSAGE	
Layer Sizes	(128, 128, 64)	(128, 128, 64)
Learning rate	GCN, GAT: 0.01 GraphSAGE: 0.03	GCN, GAT: 0.01 GraphSAGE: 0.05
Optimizer	Adam	Adam
Epoch	150	120
Batchsize	8	8

Experiment Result

	Facebook					Github				
	AUC%	Acc%	Pr%	Re%	F1%	AUC%	Acc%	Pr%	Re%	F1%
GCN	85.59	73.21	67.21	90.66	77.19	81.23	61.67	56.98	95.28	71.31
GAT	87.25	74.32	68.26	90.90	77.97	75.26	53.24	51.68	99.30	67.98
GraphSAGE	68.07	53.70	51.94	98.84	68.10	52.67	52.19	51.65	68.57	58.92
GCN*	87.40	75.42	69.05	92.16	78.95	.	.	.	.	.
GAT*	85.23	72.64	66.76	90.17	76.72	.	.	.	.	.

* Use twice as many negative samples during training.

Conclusion & Future Work

- In the Facebook dataset, the AUC reached 87%, and in the GitHub dataset, it reached 81%.
- Subgraph explanations may pose privacy risks; however, when data is missing, retaining essential parts offers a chance for partial reconstruction.

- Resolving the challenge of predicting a large number of edges from a small subset [4].
- Try new methods for selecting negative samples [5].
- Consider the features selected by GNNExplainer (Solve NA problem) [6].

Questions

- Assuming that I have obtained good experimental results using explanatory subgraphs, the data from a minority of users in practical applications may still differ from these subgraphs.
- Or should my objective be to develop a generic model that can reconstruct social network structures, trained on samples obtained from explanatory subgraphs?

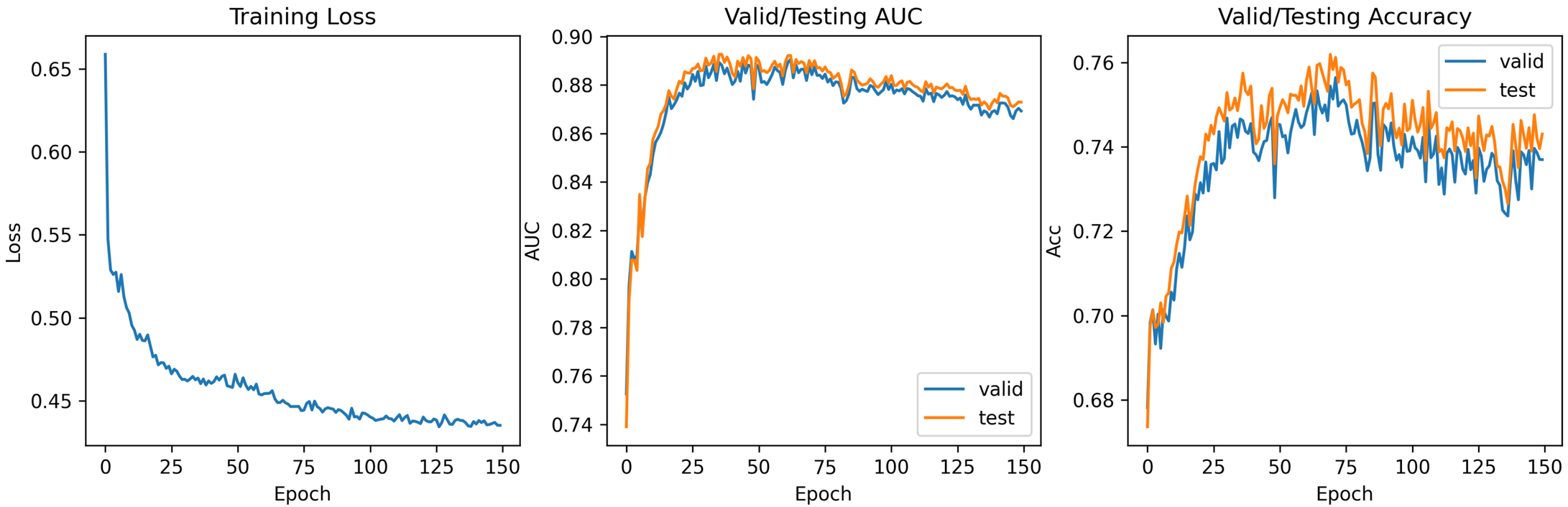
Reference

- [1] Ying et al. **“GNNExplainer: Generating Explanations for Graph Neural Networks”** NeurIPS 2019. https://arxiv.org/abs/1903.03894
- [2] Tan et al. **“Learning and Evaluating Graph Neural Network Explanations based on Counterfactual and Factual Reasoning”** WWW 2022. https://arxiv.org/abs/2202.08816
- [3] Zhang et al. **“MixupExplainer: Generalizing Explanations for Graph Neural Networks with Data Augmentation“** KDD 2023. https://arxiv.org/abs/2307.07832
- [4] Kim et al. **“Edge-Labeling Graph Neural Network for Few-shot Learning”** IEEE2019.
- [5] Jin et al. **“Refined Edge Usage of Graph Neural Networks for Edge Prediction”**
- [6] You et al. **“Handling Missing Data with Graph Representation Learning”** NeurIPS 2020.

Thank you!

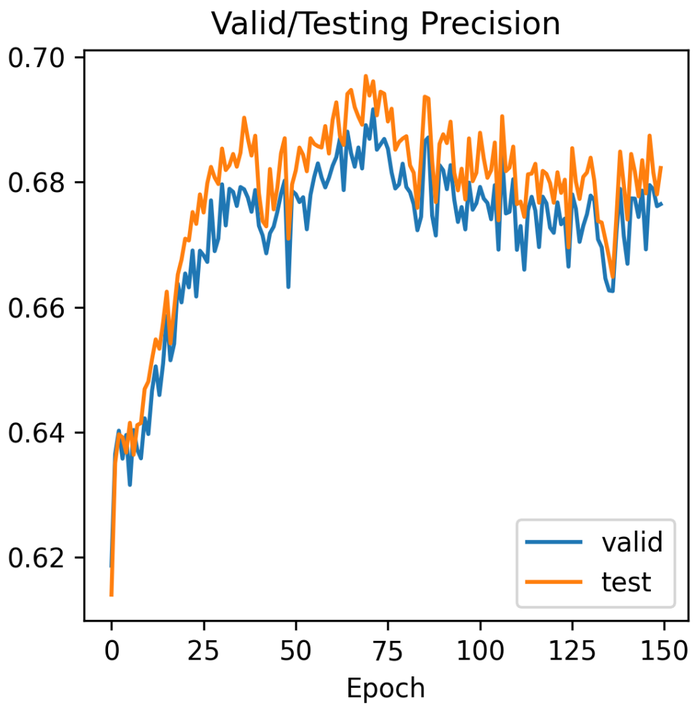
Experiment Result- Facebook

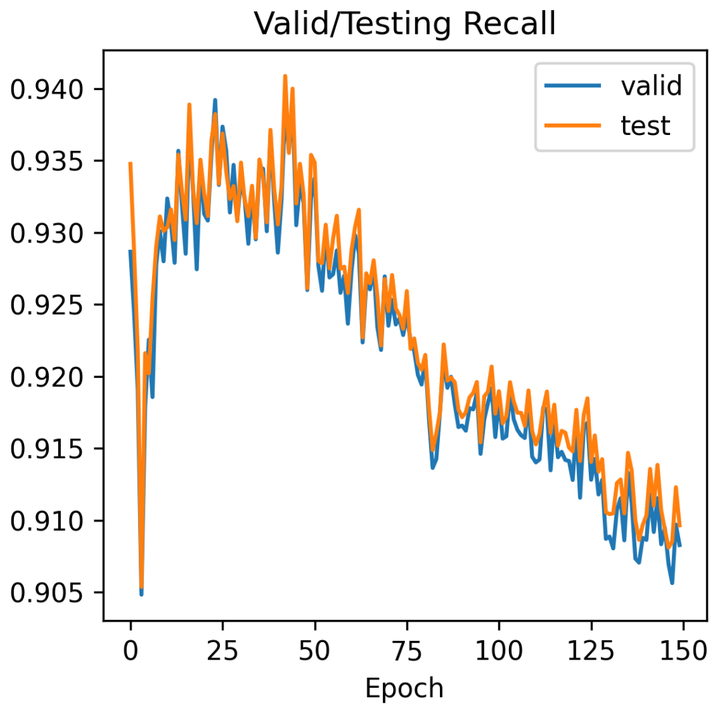
GAT Model:

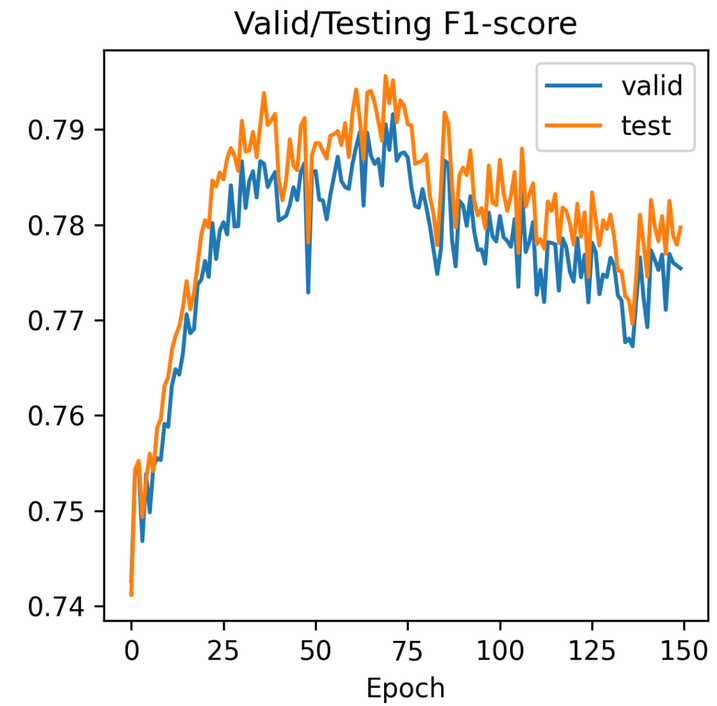


Experiment Result- Facebook

GAT Model:

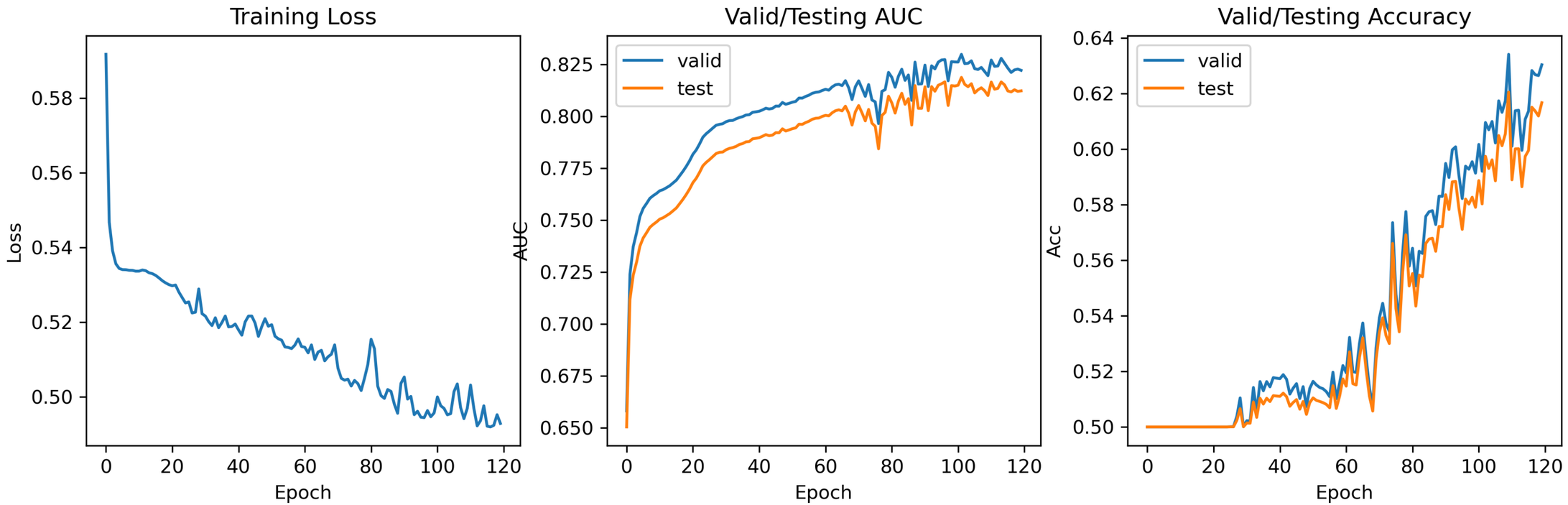






Experiment Result- Github

GCN Model:



Experiment Result- Github

GCN Model: