Disagreement in Graph Neural Network Explanation Methods

Anubhav Goel[1] **Devyani Maladkar**[1] **Shagun Gupta**[1] **Shray Mathur**[1]

Abstract

Graph Neural Networks are increasingly being used for complex tasks. The black-box nature of these models requires post hoc explanation methods to understand the decision-making process of the model. Many state of the art methods that exist to explain the model predictions do not always provide the same explanation. In practical applications, this disagreement needs to be handled carefully.

We propose metrics and perform an empirical study to quantify the disagreement in graph-based tasks among various explainers using multiple prediction models and datasets. We find disagreement among most explainers and find the degree of disagreement changing with both prediction models and datasets.

1. Introduction

Many real-world complex scenarios can be modeled as graphs, such as criminal justice (Agarwal et al., 2021), molecular chemistry (Sanchez-Lengeling et al., 2020), and biological networks (Zitnik et al., 2018). Thus, Graph neural networks (GNNs) are increasingly gaining popularity in the areas of representation learning. The complex black-box nature makes is difficult to attain an understanding of the decision-making process of these models. To identify systematic errors, potential biases and determine the reliability of the models, several post-hoc graph explainability techniques have been developed in recent literature. Most of the popular post hoc explanation methods focus on instance-level explanations of any given model (e.g., GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), Grad-CAM (Pope et al., 2019)). Certain generation-based model-level explanation methods have also been proposed (XGNN (Yuan et al., 2020)).

The increase in use of post hoc explanations to understand the behavior of GNNs makes it crucial to assess their quality and reliability. However, evaluating the quality of GNN explanations is challenging. In (Agarwal et al., 2022), the authors evaluate GNN explanation methods using synthetic datasets and find different explanation methods performed the best in different datasets and performance measures. Furthermore, it is also important to investigate whether the explanations provided by these methods for the same task disagree with each other. For instance, it is common practice for ML practitioners and data scientists to employ multiple such methods simultaneously, instead of using just one (Kaur et al., 2020). A coherent understanding of model behavior can be obtained if multiple methods generate consistent explanations. But this may not always be the case.

When the disagreement problem occurs, practitioners need to tackle it carefully as they may end up relying on misleading explanations. This could lead to catastrophic consequences – e.g., trusting and deploying racially biased models, trusting incorrect model predictions and recommending sub-optimal treatments to patients, etc. (Slack et al., 2020). Thus, it is critical to understand and quantify how often explanations output by state-of-the-art graph explanation methods disagree with each other. The authors in (Krishna et al., 2022) suggested metrics to quantify and performed an empirical study to measure disagreement in explanation methods in ML models for tabular, image and text datasets.

In this work, we study the disagreement problem in the context of GNNs and their corresponding explanation methods. We study two graph-based tasks, node classification and graph classification. We study a variety of graph models across a number of datasets to do a comprehensive study. Finally, we introduce graph-specific metrics to quantify the disagreement between methods and present the results.

1.1. Related Work

Our work builds on the vast literature on explainable graph machine learning and the limited literature on the disagreement problem.

Explainable Graph Machine Learning: In recent times, several approaches have been proposed to explain the predictions of deep graph models. These methods focus on different aspects of the graph models and provide different views to understand these models such as which input edges are more important, which input nodes are more im-

[1]University of Texas at Austin. Correspondence to: Shray Mathur <shray@utexas.edu>.

portant, which node features are more important and what graph patterns will maximize the prediction of a certain class. The techniques can be categorized into two main classes: instance-level methods (CAM (Pope et al., 2019), GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), GNN-LRP) and model-level methods (XGNN (Yuan et al., 2020)). We focus on instance-level explainers in this work. They are further categorized in Section 3.3.

The Disagreement Problem. (Krishna et al., 2022) introduced and studied the disagreement problem with a focus on tabular, image, and text datasets. More specifically, they formalized the notion of disagreement between explanations, and quantified the disagreement by proposing metrics that focused on top-k features output by explanation methods. Graphs explanations pose challenges that were not addressed in this study as graph explanations take into account the complex inherent structure of the data are typically not in the form of top-k features but are in terms of node or edge importance values. We look to extend the work done in this paper to graph based scenarios. We focus our metrics on disagreement in explanations that use graph structures.

1.2. Contributions

This paper contributes to the existing literature as follows:

1. We investigate the existence of disagreement in graph-explainable machine learning. We study the problem across various graph-based tasks, datasets, GNN-based models, and explanation methods.
2. We then formalize the notion of explanation disagreement in graphs using evaluation metrics to measure the disagreement between two explanation methods. The metrics focus on the disagreement among explainers in terms of the graph structure being used.

2. Measuring Disagreement in GNNs

GNNs are used for a number of graph-related tasks. We focus our attention on node classification and graph classification in a supervised setting. Node classification assigns a label to a node in a graph. The input is a training set of graphs with a subset of nodes in each graph annotated with their associated labels. Similarly, graph classification assigns a label to a given graph. The input is a training set of graphs annotated with their associated labels.

Explainers are used to analyze what features are being utilized by the GNNs for making these predictions. We look at the output of various explainers and come up with a formal notion of disagreement between them. For our experiments, the canonical output of an explainer is a list of important nodes associated with the prediction made by the GNN model. We use several ways to parse the output of an explainer to this list which have been described as follows:

1. If an explainer outputs a list of binary importances associated with each node for a particular prediction, we create a list of important nodes by storing the nodes corresponding to indices with bool value true.
2. If an explainer outputs a list of soft importances associated with each node (a floating point number between 0 and 1), we binarize this vector using a threshold of 0.5 and follow the methodology outlined for the first case.
3. If an explainer outputs a set of edge importances, we track the subgraph formed by the important edges and pick the nodes that are a part of this subgraph.

Having obtained the output as a uniform structure across various explainers, we now explain the metrics associated with measuring the disagreement between these outputs.

2.1. Jaccard Index

Jaccard Index measures the similarity between two sets using the idea of intersection-over-union. More formally, the Jaccard Index between two sets A and B, $J(A, B)$ is given as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

where $|S|$ denotes the number of elements in set S. As stated above, we obtain a list of important nodes associated with each prediction as the output of all explainers. For the node classification task, we obtain a list corresponding to each node in the graph. For each of our node classification datasets, we have a single graph in the dataset. Correspondingly, we average the Jaccard Index over the nodes in the graph between important nodes of two explainers. Formally, we calculate the Jaccard Index between two explainers $\mathcal{E}_1$ and $\mathcal{E}_2$ with output list of important nodes S_{1i} and S_{2i} for node i for a graph $\mathbb{G}$ with node set $\mathbb{N}$ as

$$J(\mathcal{E}_1, \mathcal{E}_2) = \frac{1}{|\mathbb{N}|} \sum_{i \in \mathbb{N}} \frac{|S_{1i} \cap S_{2i}|}{|S_{1i} \cup S_{2i}|}$$

We extend the above calculation for multiple graphs in the dataset by taking the average over all the graphs (as in graph classification) or maintaining tuples with members corresponding to each graph. A lower Jaccard Index indicates a higher degree of disagreement.

2.2. Centrality-based Measures

Centrality-based measures are scalar values assigned to nodes in a graph that quantify the importance of the node in

the graph. Based on varying notions of importance, there exist various centrality-based measures such as Degree Centrality, Closeness Centrality, Betweenness Centrality, Eigenvector Centrality, Hub Scores, and Authority Scores. The procedure that we follow to calculate disagreement based on these scores remains the same irrespective of which metric is chosen. We focus on two metrics particularly, these are described below:

1. Degree Centrality: Degree centrality for a node in a graph is the ratio of the degree of the node (number of edges connected to a node) to the maximum possible degree in the graph. It is the simplest centrality-based measure but quite effective. A node with a high degree will be more central in the graph, for example, while modeling social networks using graphs, a node with a high degree centrality will imply a highly-connected person in the social network.

2. Authority Scores: Hubs and authorities come from the idea of ranking nodes in a network of web pages. Hubs are nodes that do not contain a large amount of information (or have low authority) but lead users to pages with a high amount of information. Alternatively, hubs can be seen as nodes pointing to a large number of other nodes. Conversely, nodes with high authority scores are pointed to by a large number of pages. Authority scores can be computed using the Hyperlink-Induced Topic Search algorithm (Kleinberg, 1999) and are based on the eigenvalues of the adjacency matrix. Since hub and authority scores can be viewed as duals of each other, we focus only on authority scores.

To utilize these metrics for measuring disagreement in node classification, we calculate the average score for the set of important nodes corresponding to each node. This gives a vector of size $|\mathbb{N}| \times 1$ of average centrality scores corresponding to each explainer where $\mathbb{N}$ is the set of nodes in the graph. We calculate cosine distance $D_C(\mathbf{A}, \mathbf{B})$ between two vectors $\mathbf{A}, \mathbf{B}$ corresponding to two explainers which is given by

$$D_C(\mathbf{A}, \mathbf{B}) = 1 - \frac{\mathbf{A}^T\mathbf{B}}{\|\mathbf{A}\|_2\|\mathbf{B}\|_2}$$

to quantify the disagreement in this case. The value for this metric ranges from 0-2 and a higher value of this metric indicates a higher degree of disagreement.
In Graph Classification task, for each graph that is classified, we can calculate the average value of the centrality-based scores associated with the list of important nodes and obtain a vector of size $|\mathbb{G}| \times 1$ where $\mathbb{G}$ is the set of graphs that we perform the classification task on. We can apply the cosine distance metric to this vector to quantify the disagreement in this case.

3. Experiments

We leverage the metrics outlined in Section 2 and carry out analysis with six explanation methods. We consider GNN models for two tasks Node Classification and Graph Classification. We train GNN models for each of the tasks on widely-used datasets. In this section, we describe the datasets that we use (Section 3.1), GNN models (Section 3.2), explanation models (Section 3.3) and findings (Section 3.4).

3.1. Datasets

For the empirical analysis, we use widely used datasets. For node classification, we use Cora (Mccallum et al., 2000) and CiteSeer (Giles et al., 1998) datasets available in PyTorch Geometric. The datasets consist of academic publications as the nodes and the citations between them as the links: if publication A cites publication B, then the graph has an edge from A to B. The nodes are classified into one of the subjects.

For graph classification, we use TUDataset (Morris et al., 2020) introduced by for graph classification, implemented in PyTorch Geometric. The MUTAG (Debnath et al., 1991) and PROTEINS (Dobson & Doig, 2003) dataset in particular are selected for training. The MUTAG dataset consists of of small molecules as graph (node as atoms and edge as bonds) and class label representing the mutagenicity of the molecule. The PROTEINS dataset arises from the field of bio-informatics and consists of macro-molecules with each amino-acid represented as node. The class labels indicate whether the macro-molecule is an enzyme or not.

3.2. GNN Models

The GNN models trained for the node classification and graph classification tasks are briefly described below. The models were trained to obtain sufficient accuracy (Table 1 and Table 2). More advanced training regimes were not required due to simplicity of models. Complex models were avoided, but could be used for further analysis to study effect of complexity on explainers.

GCN: This simple neural network model consists of a graph convolution layer (Kipf & Welling). This layer is similar to a conventional dense layer, with additional ability to use structure of the graph, in particular information of the node neighbours. The model was trained for node classification.

GAT: Graph Attention Network (Veličković et al., 2017) uses additional attention coefficients alongside the graph convolution layer implementation. We used two attention layers and trained the model for node classification task.

GCN_3L: The model uses 3 Graph Convolution layers and is trained for graph classification.

Table 1. Node Classification Models Summary

DATASET	MODEL	TRAIN ACC.	TEST ACC.
CORA	GCN	81.24	80.7
	GAT	80.83	78.7
CITESEER	GCN	70.57	71.6
	GAT	69.7	70.5

Table 2. Graph Classification Models Summary

DATASET	MODEL	TRAIN ACC.	TEST ACC.
MUTAG	GCN_3L	76.79	75
	GRAPH CONV	84.82	90.79
PROTEINS	GCN_3L	71.66	68.39
	GRAPH CONV	81.26	70.18

GraphConv: This graph neural network model uses the GraphConv layer introduced in (Morris et al., 2018). It improves the generalisation ability by modifying the normalisation term accounting for the neighbouring node outputs. This performs better in recognising higher-order structures in graphs. These higher-order structures are important in the characterization of social networks and molecule graphs and hence makes this model suitable for the graph classification task.

3.3. Explainers

In order to study disagreement, we select six explainers. These explainers were used to obtain node-importance lists used for measuring disagreement as mentioned in Section 2. The three main types are Gradient Based, Perturbation based and Surrogate Based, summarised in Figure 1.

Gradient Based

The gradient based methods are very popular neural network explanation methods. These use the gradients or hidden feature map values as an approximation for input importance. Using backpropagation, the gradients of the target prediction are computed w.r.t. the inputs. These are extending in a straightforward manner to GNNs. The explanation methods - integrated gradients (ig), CAM (cam) (Pope et al., 2019) and GradCAM (gcam) (Pope et al., 2019) belonging to this type were used for studying disagreement.

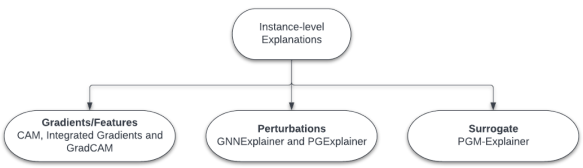

Figure 1. Flowchart categorizing the GNN explainers

Perturbation Based

Perturbation based methods are widely used for explaining deep image models. The idea is to use different input perturbations to study how the output of the model varies. Explainers for graph neural networks generate masks of node, edge, node features and combine them with the input to create a new graph that is fed to the GNN. The masks are updated using the prediction from the GNN for the new graph. This mask update and creation implements the notion of perturbation in the input. Depending on the information retained in the graph and the effect on the prediction the importance of the various graph attributes is obtained. The GNNExplainer (gnn) (Ying et al., 2019) and PGExplainer (pge) (Luo et al., 2020) are two such perturbation methods used for studying disagreement. The methods mainly differ in the mask generation and mask update procedures.

Surrogate Based

Surrogate based explainers use an interpretable surrogate model to approximate the predictions of the deep model for the neighbouring areas of the input model. For graph neural networks, for a given input graph they first obtain a local dataset and fit an interpretable model. PGM Explainer (pgm) (Vu & Thai, 2020) uses a probabilistic graphical model to provide instance-level explanations and the local dataset is obtained by random node feature perturbations.

3.4. Results

In this section we present our metrics over the models and explainers previously described. Due to implementation and time constraints, we present results for only Integrated Gradients, GNN Explainer and PGM Explainer for GAT node classification models. But these are still explainers of different categories as discussed in the previous section. We are unable to present results for GraphConv models for graph classification. While calculating disagreement in node classification, we only consider the nodes correctly classified by the model. Similarly, we only consider correctly classified graphs in graph classification models.

Node Classification

Figure 2 and Figure 3 summarize the Jaccard Index among explainers for GCN models for Cora and Citeseer datasets respectively. The low Jaccard Index among explainers showcases the disagreement in the set of nodes. We see agreement among PGE Explainer, CAM, and Grad-CAM but we observe these explainers have been returning a trivial set of nodes for the explanation, i.e., these explainers have been returning all the neighbors accessible to the convolution layers of the GCN model.

Figure 4 and Figure 5 summarize the authority score based centrality score cosine distance among different explainers

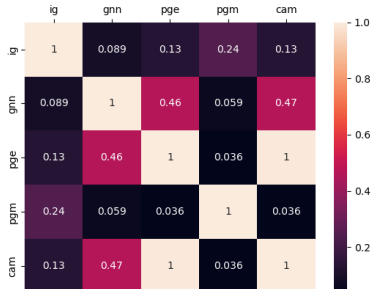

Figure 2. Heatmap of Jaccard Indices for GCN model with Cora dataset

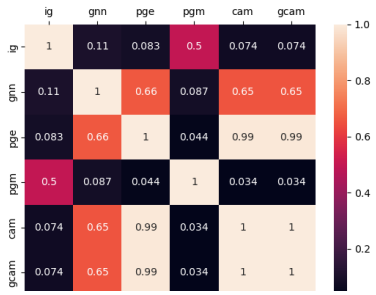

Figure 3. Heatmap of Jaccard Indices for GCN model with Citeseer dataset

for the GCN model for Cora and Citeseer datasets respectively. Other than the trivial explanation methods (PGE Explainer, CAM and Grad-CAM), all other explainers show high distances. Thus, while the nodes being selected by the explainers are different shown by the Jaccard Index, the nodes being selected also have a different connectivity within the graph. Thus the explainers disagree on the type of nodes being selected as well.

In Figure 6 and Figure 7, we take a random sample of nodes classified from test data and plot the average degree centrality of important nodes selected by the explainers. You can see by the spread across explainers that by degree centrality as well, the nodes being selected have different connectivity. A heatmap for degree centrality cosine distances and all results for GAT models showing similar trends can be found in the appendix.

Graph Classification

Figure 8 and Figure 9 summarize the Jaccard Index among explainers for GCN_3L model for MUTAG and PROTEINS dataset respectively. Again, the low values of Jaccard Index for both datasets show disagreement in the nodes being selected by the explainers to understand the classification. When we look at the authority score-based cosine distances, we observe very low values among certain explainers in MUTAG and almost among all explainers in the PROTEINS dataset.

While there exists disagreement in the type of nodes being

Figure 4. Heatmap of Authority score cosine distances for GCN model with Cora dataset

Figure 5. Heatmap of Authority score cosine distances for GCN model with Citeseer dataset

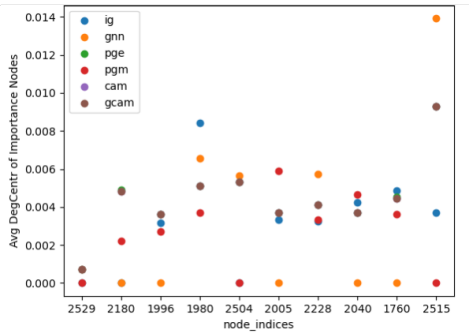

Figure 6. Average degree centrality score of important nodes selected by explainers for classification of a random sample of nodes in Cora dataset by GCN model

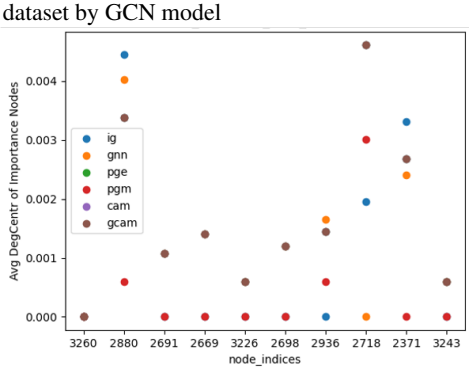

Figure 7. Average degree centrality score of important nodes selected by explainers for classification of a random sample of nodes in Citeseer dataset by GCN model

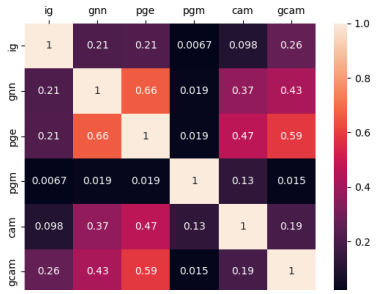

Figure 8. Heatmap of Jaccard Indices for GCN_3L model with MUTAG dataset

Figure 9. Heatmap of Jaccard Indices for GCN_3L model with PROTEINS dataset

selected by explainers in the MUTAG dataset, the nodes being selected by the explainers in the PROTEINS dataset are the same based on authority score. This shows that the explanations being provided in the PROTEINS dataset while selecting different nodes, do agree on the type of nodes being used for the explanation. Thus, the explanations are alternatives to one another. In Figure 12 and Figure 13 we observe that in the MUTAG dataset, we still have the spread across explainers, while the average degree centrality of important nodes of explainers bunch together and overlap in PROTEINS datasets. PGM Explainer is found to be an odd one out in all graph classification models which we believe is the artifact of it being the only surrogate-based explainer. The heatmap for degree centrality cosine distances is available in the appendix.

Figure 10. Heatmap of Authority score cosine distances for GCN_3L model with MUTAG dataset

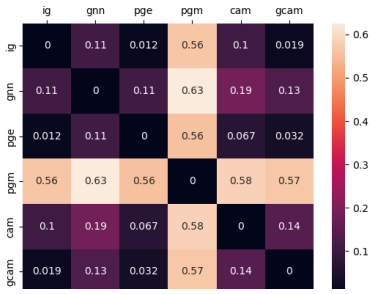

Figure 11. Heatmap of Authority score cosine distances for GCN_3L model with PROTEINS dataset

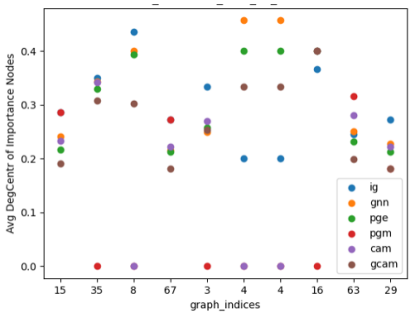

Figure 12. Average degree centrality score of important nodes selected by explainers for classification of a random sample of graphs in MUTAG dataset by GCN_3L model

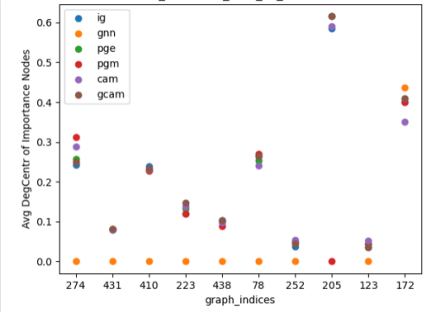

Figure 13. Average degree centrality score of important nodes selected by explainers for classification of a random sample of graphs in PROTEINS dataset by GCN_3L model

4. Final Remarks

In this project, we study the disagreement problem in graph neural networks for node classification and graph classification tasks. We proposed 2 metrics to measure disagreement among explainers. We conduct this empirical study across various datasets and classification models. Using Jaccard Index, we found disagreement in the nodes being selected by explainers across all the cases we examined. Using centrality based scores we found that these explanations are sometimes different in the type of nodes being selected. We also found examples where the explanations select different graph nodes that have similar connectivity within the network. We believe these are alternate explanations being provided by different explainers.

Based on our findings, we firstly always recommend using multiple explainers and not trust one explainer when analysing GNNs as well. As we see, the agreement among explainers can change with both the model and the dataset, such a study before model deployment is really useful to find the best explainers for use. Also, the study should always include explainers from various categories. Within the limited number of explainers we study, we find GNN explainer to be the most coherent with other explainers for both node and graph classification tasks.

We focused our metrics to elements of the explanation that exploit the graph structure but a study of disagreement with respect to other features like node features is also necessary. The study also can be extended to link prediction tasks. As we found, there are sometimes fundamental differences in the connectivity of nodes being selected. Thus an study that focuses on properties of explainers that makes them select different nodes is an important avenue to explore.

Software and Data

The codes are available on GitHub https://github.com/YANI-ALT/FML-GNN_DisagreementProblem. The datasets were obtained from PyTorch Geometric. The GNN models were implemented in PyTorch and were trained locally on a CPU, the weights and models can be found in the GitHub repo. The implementations for the explainers were obtained from the GraphXAI library (Agarwal et al., 2022).

Acknowledgements

We would like to thank the authors of GraphXAI library (Agarwal et al., 2022) and DIG library (Yuan et al., 2022) for open-source implementations of explainers for the study.

References

Agarwal, C., Lakkaraju, H., and Zitnik, M. Towards a unified framework for fair and stable graph representation learning. In *Uncertainty in Artificial Intelligence*, pp. 2114–2124. PMLR, 2021.

Agarwal, C., Queen, O., Lakkaraju, H., and Zitnik, M. Evaluating explainability for graph neural networks. *arXiv preprint arXiv:2208.09339*, 2022.

Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., and Hansch, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34 (2):786–797, 1991. doi: 10.1021/jm00106a046. URL https://doi.org/10.1021/jm00106a046.

Dobson, P. D. and Doig, A. J. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771—783, July 2003. ISSN 0022-2836. doi: 10.1016/s0022-2836(03)00628-4. URL https://doi.org/10.1016/s0022-2836(03)00628-4.

Giles, C. L., Bollacker, K. D., and Lawrence, S. Citeseer: An automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries*, DL '98, pp. 89–98, New York, NY, USA, 1998. Association for Computing Machinery. ISBN 0897919653. doi: 10.1145/276675.276685. URL https://doi.org/10.1145/276675.276685.

Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., and Wortman Vaughan, J. Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–14, 2020.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. URL https://arxiv.org/abs/1609.02907.

Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

Krishna, S., Han, T., Gu, A., Pombra, J., Jabbari, S., Wu, S., and Lakkaraju, H. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv preprint arXiv:2202.01602*, 2022.

Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., and Zhang, X. Parameterized explainer for graph neural network. *CoRR*, abs/2011.04573, 2020. URL https://arxiv.org/abs/2011.04573.

Mccallum, A., Nigam, K., and Rennie, J. Automating the construction of internet portals. 03 2000.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks, 2018. URL https://arxiv.org/abs/1810.02244.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., and Hoffmann, H. Explainability methods for graph convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10764–10773, 2019. doi: 10.1109/CVPR.2019.01103.

Sanchez-Lengeling, B., Wei, J., Lee, B., Reif, E., Wang, P., Qian, W., McCloskey, K., Colwell, L., and Wiltschko, A. Evaluating attribution for graph neural networks. *Advances in neural information processing systems*, 33: 5898–5910, 2020.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186, 2020.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks, 2017. URL https://arxiv.org/abs/1710.10903.

Vu, M. N. and Thai, M. T. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *CoRR*, abs/2010.05788, 2020. URL https://arxiv.org/abs/2010.05788.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

Yuan, H., Tang, J., Hu, X., and Ji, S. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 430–438, 2020.

Yuan, H., Yu, H., Gui, S., and Ji, S. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

A. Appendix

The results for GAT models for graph classification and a few supplementary results for the GCN models for node classification and GCN_3L for graph classification are presented in this section.

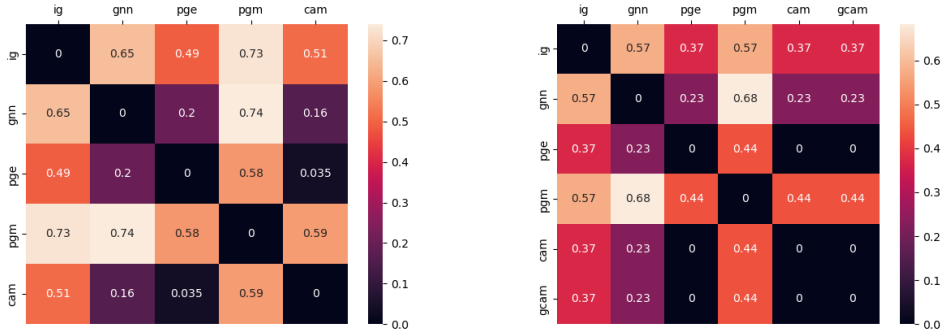

Figure 14. Heatmap of degree centrality cosine distances for GCN model with Cora dataset on the left and Citeseer dataset on the right

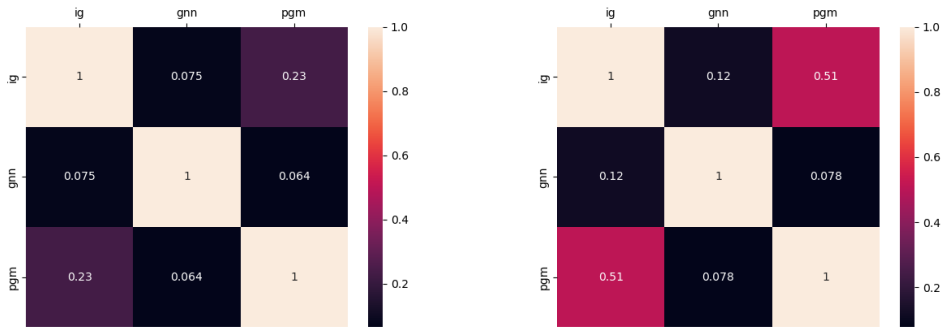

Figure 15. Heatmap of Jaccard similarity scores for GAT model with Cora dataset on the left and Citeseer dataset on the right

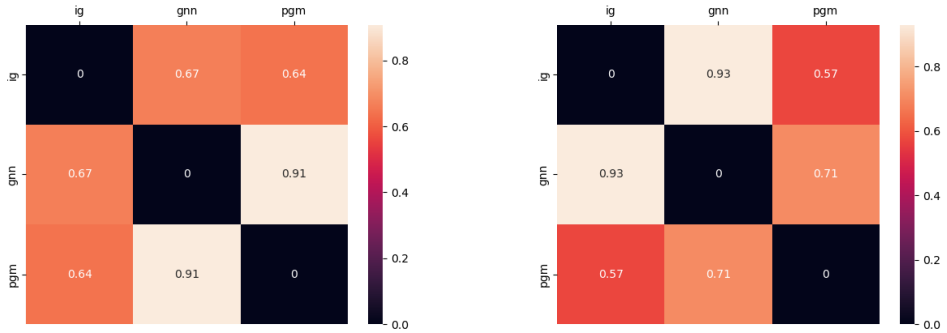

Figure 16. Heatmap of Authority score cosine distances for GAT model with Cora dataset on the left and Citeseer dataset on the right

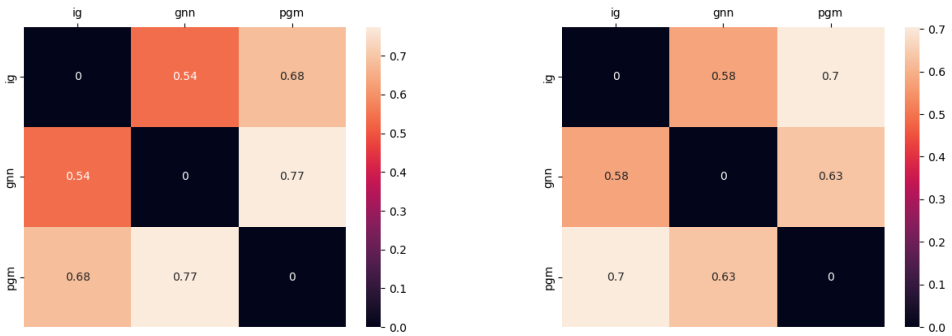

Figure 17. Heatmap of degree centrality cosine distances for GAT model with Cora dataset on the left and Citeseer dataset on the right

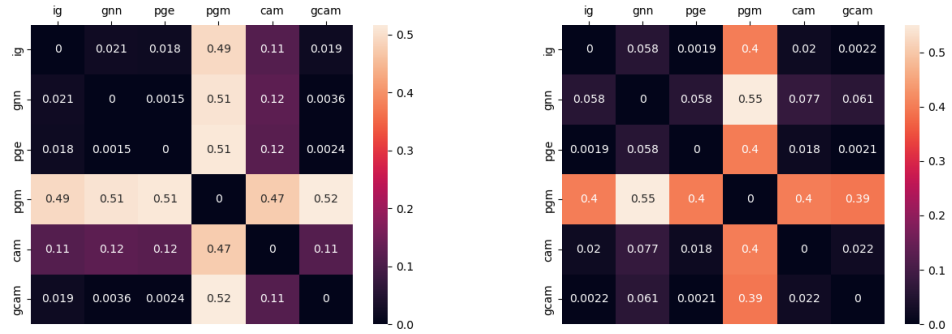

Figure 18. Heatmap of degree centrality cosine distances for GCN_3L model with MUTAG dataset on the left and PROTEINS dataset on the right