Influence of graph neural network architecture on explainability of protein-protein interaction networks

Hubert Janczak

Supervisor(s): Dr. Megha Khosla, Dr. Jana Weber

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Hubert Janczak
Final project course: CSE3000 Research Project
Thesis committee: Dr. Megha Khosla, Dr. Jana Weber, Dr. Thomas Abeel

Abstract

AI explainers are tools capable of approximating how a neural network arrived at a given prediction by providing parts of the input data most relevant for the model's choice. These tools have become a major point of research due to a need for human-verifiable predictions in multiple different fields, such as biomedical engineering. Graph Neural Networks (GNNs) are often used for such tasks, which led to the development of GNNSubnet, a tool capable of finding disease subnetworks on models trained with protein-protein interaction (PPI) data. This tool has been tested with only a single GNN architecture, which left a knowledge gap about the performance of the tool under different models, which can differ significantly in the way they operate.

Here the question "How does the explainer performance vary with change in architectures of training models?" is answered.

This paper explores this knowledge gap by training and evaluating two other models (GCN and GraphSAGE) to see if the explanation performance of GNNSubnet changes. The performance is evaluated with BAGEL metrics, a tool developed for XAI analysis. These metrics allow for comparison of explanations on multiple benchmarks. Three of these - Fidelity, Validity- and Validity+ - measure how accurate an explanation was in terms of identifiying important nodes. The last one - Sparsity - assesses the nontriviality of an explanation by measuring how few nodes have been identified as important.

The experimental process shows low performance changes with different GNN architectures for accuracy-related metrics - RDT-Fidelity, Validity- and Validity+, which means that GNNSubnet is highly generalizable and not tied to a specific GNN model. However, the Sparsity score differs across models, with GIN being able to provide the most concise - and therefore useful - explanations.

1 Introduction

Due to the recent rise of artificial intelligence in correctness-critical fields, such as biomedical and chemical research, explainable AI tools have become a major area of research within the field of AI[1]. In these areas the predictions of AI can have significant impact on human lives, and therefore need to be trustworthy[2]. AI explainability techniques have gained much traction due to their ability to provide parts of input data relevant for the underlying model's prediction, which then can be crosschecked with human expertise.

Graph Neural Networks have shown impressive performance in analyzing complex structures. Graphs are sets of objects some of which have a relation to each other, such as friend networks or road systems. The objects are represented as vertices (nodes) which are connected to related vertices with edges. GNNs, when analysing this kind of data, are able to draw conclusions based on proximity and relation of different items. They have found applications in multiple areas, such as combinatorial optimization, traffic networks or recommendation systems[3], but also in fields where explainability is mandated by law[4], such as protein-protein interaction (PPI) modeling.

In organisms, proteins create dependent chains which influence one another, allowing for signal transportation and complex metabolic processes. This creates a graph-like structure that can be processed with GNNs to find networks with, among other things, higher cancer risk. However, similarly to regular neural networks, GNN models are inherently black-box and provide little to no information on how they draw their conclusions, making it difficult to assess the plausibility and generalizability of the model's approach[5].

This rise in popularity of trust-critical GNNs has led to development of multiple explainability tools for GNNs[2]. One such tool is GNNSubnet, proposed by Pfeifer et al.[6], built on top of GNNExplainer [7]. GNNSubnet can be run on GNN models trained with PPI cancer data. Then, it finds disease subnetworks in the datasets which were most relevant for the underlying model's cancer classification.

The GNNSubnet paper[6] tests the explainer with a graph isomorphism network (GIN)[8]. However, no information is provided on GNNSubnet's performance with other popular models. The underlying aggregation techniques can make it easier or more difficult for GNNSubnet to extract causal relationships, which would significantly impact its performance. Like GNNExplainer, GNNSubnet can be used with any GNN architecture, and other GNN models could lend themselves better for the task of subnetwork detection.

This paper answers the following research question: "How does the explainer performance vary with change in architectures of training models?". To that goal, multiple GNN models from different families will be trained and explained with GNNSubnet, and these explanations will be evaluated on multiple metrics. This will provide previously unexplored insight into robustness of GNNSubnet.

The following article is organised as follows: section 2 provides information about different GNN architectures as well as descriptions of explainer evaluation metrics. In section 3, the experimental setup is discussed. Finally, the results are presented in section 4 and discussed in section 6.

2 Theoretical background

2.1 GNNSubnet

GNNSubnet is capable of uncovering disease subnetworks - communities in a graph whose features are most responsible for a model's decision - based on analyzing the predictions of a GNN model. Firstly, it optimizes a node mask by sampling graphs from the input space and assigning importance values to each node. Then, it performs community detection to find communities with high importance scores, which are the potential disease subnetwork. The process is explained in detail in [6].

GNNSubnet was benchmarked with two additional GNN models: a Graph Convolutional Network (GCN)[9], and

GraphSAGE[10].

Table 1: Descriptions of mathematical notation used in section 2

Notation	Description
$\mathbf{h}$	Updated features of a node
ψ	Transformation function
$\bigoplus$	Aggregation function
ϕ	Combination (update) function
$\mathbf{x}$	Node feature vector
$\mathcal{N}(v)$	Neighbours of node v
ReLU	Rectified Linear Unit function
MEAN	Mean aggregation function

2.2 Description of GNNs

The input data for a GNN is a graph G in the form of $G = (V, E)$, where V is a set of vertices and E is a set of edges between those vertices. Moreover, as is in the case of data analysed by GNNSubnet, nodes can have additional (multi-omic) information, such as DNA methylation. Specifically in case of PPI networks analysed here, the nodes represent proteins and edges represent the interactions between them. An additional feature of PPI networks is that nodes and edges remain the same across all graphs, representing the interactions in the human body. The differences in the networks come from the node features.

Most graph neural networks (including the ones analysed here) work in a similar framework[3], described underneath.

The graph goes through multiple message passing layers. In each layer, every node performs local pooling - it transforms the information about each of its neighbours through some message passing function ψ, and then runs the data through an aggregation function $\bigoplus$ (e.g. mean, sum, max). Then, the data of the node and aggregated neighbour information is combined with another function, ϕ. This process is repeated multiple times until each node's information is a representation of its data and its spatial relationship with its surroundings. Overall, the equation for each step, as presented by Bronstein et al. [11] is:

$$\mathbf{h}_u^k = \phi\left(\mathbf{h}_u^{k-1}, \bigoplus_{v\in\mathcal{N}_u} \psi(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1})\right) \tag{1}$$

With $\mathbf{h}_u$ the information of the node being iterated over, and $\mathbf{h}_v$ a neighbour node. The step of getting the neighbour information and running it through the function $\bigoplus$ is usually referred to as the aggregation step. Then, combining current node and aggregated neighbour data is referred to as the combine step.

Graph Isomorphism Network

GIN is the model chosen by the creators of GNNSubnet for the underlying architecture that the explainer was tested on. This model is remarkable due to it being proven by its authors to be as powerful as the Weisfeiler-Lehman isomorphism test[8]. In the paper, Xu et al. also prove that no GNN can be more powerful than the WL test, making GINs the most powerful GNNs - in terms of number of graphs they can distinguish - that can be constructed. The aggregation and combination formula, as presented by the authors, is:

$$a_v^{(k)} = \sum\nolimits_{u\in\mathcal{N}(v)} h_u^{(k-1)} \tag{2}$$

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + a_v^{(k)}\right) \tag{3}$$

Here, the aggregation step is a sum of all the neigbouring node values. Then, in the combination step, the aggregated sum is added to the node value multiplied by some (potentially learnable) scalar, and finally run through a multi-layer perceptron to obtain the final value.

Graph Convolutional Network

A GCN[9] is one of the simplest types of GNNs, with both GraphSAGE and GIN having been built on top of the theoretical framework of Graph Convolutional Networks. The equation for a GCN layer is as follows:[8]

$$h_v^{(k)} = \mathrm{ReLU}\left(W\cdot\mathrm{MEAN}\left\{h_u^{(k-1)},\ \forall u\in\mathcal{N}(v)\cup\{v\}\right\}\right). \tag{4}$$

The neighbours of the node and the node itself are aggregated by calculating element-wise mean pooling, which is then multiplied by a learnable matrix, W. Then the result is put into the rectified linear unit (ReLU) activation function.

GraphSAGE

GraphSAGE stands for SAmple and aggreGatE. Unlike the other models, it is an inductive learning model - it learns from a sample of the input data and then generalizes, which allows it to perform well on evolving graphs with previously unseen data. The authors propose a few aggregation functions[10]. Here, the variant using the MEAN function is used:[8]

$$a_v^{(k)} = \mathrm{MEAN}\left(\left\{\mathrm{ReLU}\left(W\cdot h_u^{(k-1)}\right),\ \forall u\in\mathcal{N}(v)\right\}\right) \tag{5}$$

$$h_v^{(k)} = W\cdot\left[h_v^{(k-1)}, a_v^{(k)}\right] \tag{6}$$

In the combine step, the aggregation results and the value of the current node are multiplied by a learnable weight matrix W to obtain the final result.

In this research the dataset allows for global sampling, so no neighbourhood sampling was used. Neighbourhood sampling is useful to lower the runtime of the model, which here was not a major bottleneck. The authors of the GraphSAGE paper[10] show that global sampling is the most optimal method for model accuracy.

2.3 Chosen evaluation metrics

GNNSubnet's explanation performance on different models is compared using BAGEL benchmarks[5], a tool developed to analyse the effectiveness of explainers with different metrics. Detailed explanation and equations for each of the metrics can be found in [12] and [5]. The metrics used are Faithfulness, Validity+, Validity- and Sparsity.

Faithfulness and Validity allow for assessing the "accuracy" of the explainer - whether it correctly chooses the important subnetworks. Sparsity shows how nontrivial the explanation is - that is, how few nodes were marked as important. Those metrics together give a good overview of the usefulness of the result.

Some other metrics defined in BAGEL, such as sufficiency, necessitate removing parts of the graph, which makes the PPI network incorrect, so they are not useful for this dataset.

Faithfulness (RDT-Fidelity)

The Faithfulness metric answers whether the explanation correctly reflects the model's decision process. This is achieved by randomly perturbing all proteins whose importance value is below a certain threshold. If the values were correctly identified as unimportant, the model's prediction should not change. The specific formula for faithfulness used here is named RDT-Fidelity, as explained in [5]:

"The RDT-Fidelity of explanation $\mathcal{S}$ corresponding to explanation mask $M(\mathcal{S})$ with respect to the GNN f, input X and the noise distribution $\mathcal{N}$ is given by

$$\mathcal{F}(\mathcal{S}) = E_{Y_{\mathcal{S}}|Z\sim\mathcal{N}}\left[\mathbb{1}_{f(X)=f(Y_{\mathcal{S}})}\right]. \tag{7}$$

where the perturbed input is given by

$$Y_{\mathcal{S}} = X \odot M(\mathcal{S}) + Z \odot (\mathbb{1} - M(\mathcal{S})), Z \sim \mathcal{N}, \tag{8}$$

where $\odot$ denotes an element-wise multiplication, and $\mathbb{1}$ a matrix of ones with the corresponding size and $\mathcal{N}$ is a noise distribution."

In short, Equation 8 applies a noise mask to all features below the importance threshold. Then, Equation 7 finds the average number of times that the explanation of the perturbed input agrees with the explanation of the old input. The higher the result is, the better the accuracy of GNNSubnet.

Sparsity

This metric evaluates the nontriviality of an explanation. If an explainer lists all nodes (proteins) as relevant to the model's decision, that is a trivial explanation - no information was learnt. The more concise the explanation, the better (and therfore higher) the Sparsity value.

Formally, it is defined as the entropy over the normalized distribution of masks:

$$H(p) = -\sum_{\phi\in M} p(\phi)\log p(\phi).[5] \tag{9}$$

Validity

Validity aims to answer a similar question to faithfulness - how well the explanation reflects the models workings. However, instead of using randomised values, baseline values (like average of the node values) are used. Here we use two variations of Validity: Validity+, which perturbs the important nodes and expects high change, and Validity-, which perturbs the unimportant nodes and expects low change.

$$Validity+ = \frac{1}{N}\sum_{i=1}^{N}(f(\mathcal{G}_i)_{y_i} - f(\mathcal{G}_i^{1-m_i})_{y_i}) \tag{10}$$

$$Validity- = \frac{1}{N}\sum_{i=1}^{N}(f(\mathcal{G}_i)_{y_i} - f(\mathcal{G}_i^{m_i})_{y_i}), \tag{11}$$

Here, $f(\mathcal{G})_{y_i}$ is the model prediction for some given graph i. $\mathcal{G}^{1-m}$ is a mask of the graph with unimportant nodes perturbed, while $\mathcal{G}^m$ is a graph with important node perturbed. This yields the average of changes over the important and unimportant predictions for Validity+ and Validity-, respectively.

Finally, the obtained Validity+ scores were put into the following normalization function:

$$v' = 1 - 2(|0.5 - v|) \tag{12}$$

With v being the result of the Validity+ function.

The normalization was performed since originally, the optimal Validity+ score is 0.5, with the score becoming worse in both directions (i.e. 0.4 is worse than 0.5, and as bad as 0.6). This occurs because of the way the GNN operates on the perturbed values. If the explainer was entirely correct, all important nodes will be perturbed to mean values, making them look unimportant. This leads to the GNN having to make a prediction at random, leading to an average score of 0.5. Deviation of this score in either direction means that the GNN is not guessing, so some important nodes remained in the graph. The normalization function makes it so that score of 1 is the most optimal (as v' is 1 when v is 0.5) and 0 is the least optimal.

Global and local explanations

The PPI datasets contain large number of graphs with the same node and edge structure, which only differ in their node feature information. This allows for two approaches to explaining a model: global and local explanation.

In a global explanation, the model provides a single node mask that assigns an importance score to each node. This is achieved by sampling values from the input data, then optimizing a mask using gradient descent. This way, the final node mask represents the entire dataset. GNNSubnet was written to provide global explanations of the dataset it is operating upon.

[13] modified GNNSubnet to acquire local explanations from the tool. Instead of sampling nodes' features from multiple graphs to obtain a single mask, the node masks are provided for each graph separately (effectively providing N masks for N graphs). Then, these local explanations can be aggregated (e.g. using a mean function) to obtain a final result.

The following study, while focusing on global explanations, also compares both local and global explanations on each of the three models.

3 Explainer evaluation pipeline

The overview of the experimental process is available in Figure 1.

3.1 Tools

The two new models being evaluated were programmed as an extension of the existing codebase of GNNSubnet. The models were created using the PyTorch-Geometric library, in line with the authors' GIN implementation to reduce efficiency differences. Then, all models were trained using The Cancer Genom Atlas (TCGA) dataset.

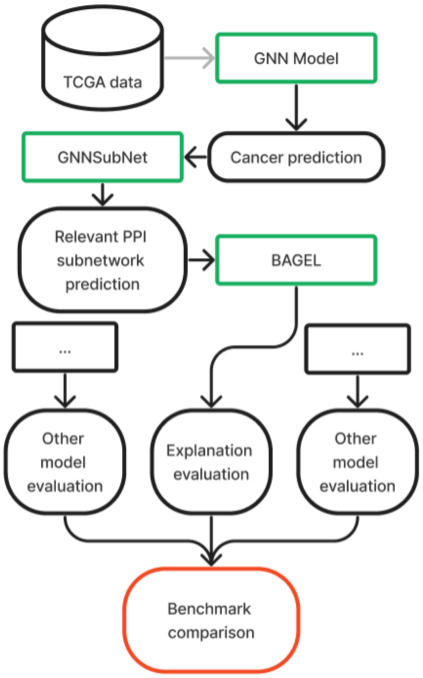

Figure 1: Demonstration of the evaluation process. Each model is trained with TCGA data, then explained with GNNSubnet. The resulting explanations were processed with BAGEL to provide a comparison of the benchmarks.

Table 2: GNN model accuracy on the KIRC dataset. Each model was trained 10 times. Both GIN and GCN occasionally learnt a model which always returned the same value, which resulted in validation accuracy of 0.5.

Model	Min	Mean	Max	δ
GIN	0.5	0.69	0.85	0.12
GCN	0.5	0.61	0.72	0.09
GraphSAGE	0.80	0.87	0.92	0.03

Once trained, the models are evaluated with GNNExplainer and processed with GNNSubnet's subnetwork detection algorithm. Finally, the explainer's performance was evaluated with BAGEL.

3.2 Model training

First, each of the three models discussed in subsection 2.2 were implemented and trained using the KIRC dataset obtained from the TCGA database. The models were trained over 20 epochs with a learning rate of 0.01. Once trained, the models achieved accuracies presented in Table 2.

3.3 Explanations and BAGEL Evaluation

The models were linked to GNNSubnet and each trained version of the model was processed with two versions of the explainer - global and local - obtaining 10 explanations for each of the models on each explainer. Finally, every explanation obtained was evaluated with the BAGEL explainer analysis tool[5] with four different metrics, explained in subsection 2.3. The results are shown and discussed in section 4.

4 Results and Discussion

Global explanations

The evaluation of the global explanations of the three models is available in Table 3. The comparison of mean values can also be seen in Figure 2.

Despite a significantly higher accuracy of GraphSAGE in comparison to the other two models, the explanation performance is similar across the models, with GraphSAGE being the best across all metrics but Sparsity and almost tying with GCN on Validity+.

For RDT-Fidelity, GraphSAGE and GIN both obtain very high results, close to each other. GCN performed a bit worse, about one standard deviation under GIN, but the score obtained is still quite high.

GIN performed the worst for Validity+, although with high variance of the score no clear conclusions can be drawn.

All models achieved very high Validity-, showing their resilience to random perturbations in unimportant node features. Especially interesting was GraphSAGE, which achieved a Validity- score of 1 across all runs, meaning that it didn't change a prediction due to the perturbations even once.

Finally, Sparsity was the best for GIN, with a noticeable dropoff for GraphSAGE and GCN.

Local explanations

The evaluation of the models' local explanations can be seen in Table 4. The results are very similar to global explanations, with all but one value in the table being within one standard deviation from its global explanation counterpart.

The only difference is in the Sparsity of GCN, which worse for local explanations. This result did not occur for the other two models, however.

4.1 Discussion

The results show that all models are explainable, proving the generalizability of GNNSubnet. The first three metrics are similar across models, but Sparsity is better for GIN, making it overall the best model for explainability performance.

High Validity-

Validity- was extremely high for all models, especially GraphSAGE, for which it achieved a score of 1 across all runs. This shows that the models are very resistant to small perturbations in unimportant nodes. Paired with Strong RDT-Fidelity and good Validity+ scores, it is visible that the explainer correctly identifies the nodes important for GNN decisions across all models.

Sparsity differences

The logarthmic nature of sparsity means that the difference in scores between architectures is significant. The reason for GIN Sparsity being the best can be attributed to it mostly pooling information from its neighbours. Meanwhile, GraphSAGE was trained with global sampling, and GCN's learnable matrix W increases each nodes dependence on the entire graph structure. This hypothesis could be verified by testing

Table 3: Bagel metric evaluation of each model using global explanations over 10 training attempts (mean/stddev). The models were trained with KIRC data from the TCGA dataset.

Global Explanations	GIN	GCN	GraphSage
RDT-Fidelity	0.88/0.10	0.77/0.04	0.90/0.10
Normalized Validity+	0.58/0.26	0.75/0.13	0.74/0.10
Validity-	0.87/0.11	0.89/0.07	1.0/0.0
Sparsity	0.034/0.014	0.025/0.006	0.018/0.006

Table 4: Bagel metric evaluation of each model with local explanations over 10 training attempts (mean/stddev). The only major difference between the two explanations is Sparsity of GCN (bold text), which is significantly lower with local explanations.

Local Explanations	GIN	GCN	GraphSage
RDT-Fidelity	0.89/0.05	0.75/0.04	0.83/0.13
Normalized Validity+	0.49/0.23	0.57/0.10	0.75/0.12
Validity-	0.90/0.05	0.96/0.06	1.0/0.0
Sparsity	0.035/0.008	**0.0036**/0.0020	0.01/0.004

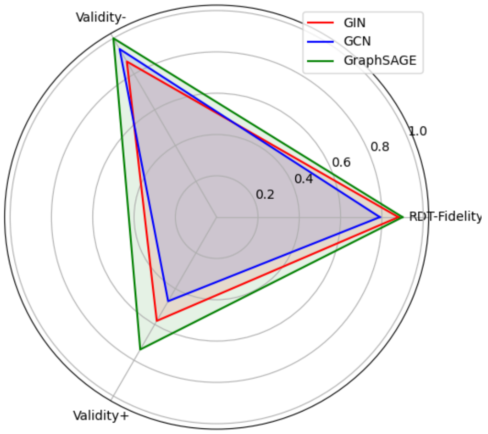

Figure 2: Radar comparison of mean Validity-, Validity+ and RDT-Fidelity calculated over global explanations. GraphSAGE shows an overall superior performance, although not significantly, which can be attributed partially to its better accuracy.

GraphSAGE with different neighbour sampling sizes, as described in section 7.

The reproducibly lower sparsity of local-explanation GCN compared to all other models (both global and local) shows that GNNSubnet can occasionally behave unpredictably under specific circumstances, which warrants further research.

5 Responsible Research

Research into the field of biomedical AI, while potentially extremely beneficial to society, carries significant amount of risk. The results of this and related pieces of research could lead to more widespread adoption of explainers, progressively removing human agency from protein-protein interaction research, which could have dangerous consequences if their predictions are incorrect. Moreover, explainability of protein-protein interaction modelling could be used to progress research in different fields than disease detection, such as genetic engineering, which raises major ethical concerns. Our research has been conducted specifically with disease detection in mind, but any research into biomedical engineering or artificial intelligence can have unexpected consequences and carries risk for society, and as such should be implemented responsibly.

This research has been conducted with reproducibility and transparency in mind. The datasets used are listed and available to the general public and have not been modified. The code written to benchmark GNNSubnet against different models is also publicly available. All data used is either synthetic (Barabasi networks) or fully anonymised (TCGA data), carrying no risk of discovery for patients who are part of the database.

The authors have no personal or financial affiliation with people or companies in the biomedical field or with the developers of GNNSubnet.

6 Conclusion

In order to assess the difference in GNNSubNet performance with change in underlying architecture, three models were tested: GIN, GCN and GraphSAGE. The explanation performance for each model was measured using four metrics: RDT-Fidelity, Validity-, Validity+ and Sparsity. Fidelity and Validity metrics concerned themselves with assessing if the explainer chose the correct important nodes for the decision, while the Sparsity metric informs how concise the result was.

Overall, there is no strong difference in explainability of different models. The Fidelity and Validity scores were all similar, with GraphSAGE slightly outperforming the other models, likely due to its higher accuracy. Such result shows that GNNSubnet is able to correctly identify disease subnetworks regardless of the techniques employed by a model.

The only variable factor for the models is Sparsity. Due to differences in the way that data is aggregated between models, GNNSubnet requires more features to provide a good explanation for some models, such as GCN, than for others.

The study shows that GNNSubnet is highly generalizable and can perform well under multiple models with very different internal architectures. However, due to its better Sparsity score, GIN explanations will generally be more concise and relevant in comparison to the two other models tested. For future disease subnetwork detection tasks GIN is the recommended architecture of choice.

7 Limitations and Future work

Due to the time constraints of the project, only three architectures were tested, which limits the generalizability of the results provided. Moreover, on the provided datasets, the mean accuracy of some models differs significantly. An additional controlled study would account any potential difference in metric comparison.

7.1 Models

This paper shows that GNNSubnet is highly generalizable across the three training models chosen for the experimental process. However, it does not cover the entire GNN model space. While the chosen architectures provide a wide range of approaches to GNN model design, multiple other options should be tested before certain conclusion can be drawn. Particularly two areas of GNN research deserve additional research:

Attentional models
Attentional GNNs, such as a Graph Attention Network (GAT)[14] are able to process variable sized data by focusing on relevant parts of a network. This approach could have an impact on explainability.

Sampling models
While GraphSAGE samples nodes in a neighbourhood, some sampling techniques, such as GraphSAINT[15], sample entire subgraphs. This technique was shown by its authors to rival GraphSAGE in accuracy on multiple models, but due to a minibatch sampling approach it could raise the error rates of explainers.

Moreover, in this study, GraphSAGE was tested using global sampling, which is shown to be the most effective. However, on many datasets, this sampling technique would prove ineffective. PPI networks can be extremely big and complex, which could make sampling techniques necessary for such tasks. As such, explainability loss under lower neighbourhood sizes should be investigated.

7.2 Sparsity exploration

The study shows that the only metric that varies significantly between architectures is Sparsity. With mean values ranging from 0.035 for locally-explained GIN to 0.0036 for locally-explained GCN on a logarithmic scale, the results vary a lot in their usefulness. In order to ensure concise and correct explainer predictions, causes for those differences should be explored further. For example, the impact of aggregation globality (such as GraphSAGE sampling size) on Sparsity requires additional research.

8 Acknowledgments

I would like to thank Sucharitha Rajesh for providing me with insight and implementation for the BAGEL metrics used in this study. I would also like to express my gratitude to Elena-Oana Milchi for helping me with implementation of the local explanations for GNNSubnet.

References

[1] Andreas Holzinger, Matthias Dehmer, Frank Emmert-Streib, Rita Cucchiara, Isabelle Augenstein, Javier Del Ser, Wojciech Samek, Igor Jurisica, and Natalia Díaz-Rodríguez. Information fusion as an integrative cross-cutting enabler to achieve robust, explainable, and trustworthy medical artificial intelligence. *Information Fusion*, 79:263–278, March 2022.

[2] Jaykumar Kakkad, Jaspal Jannu, Kartik Sharma, Charu Aggarwal, and Sourav Medya. A Survey on Explainability of Graph Neural Networks. 2023.

[3] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[4] Luca Nannini, Agathe Balayn, and Adam Leon Smith. Explainability in AI Policies: A Critical Review of Communications, Reports, Regulations, and Standards in the EU, US, and UK. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '23, pages 1198–1212, New York, NY, USA, June 2023. Association for Computing Machinery.

[5] Mandeep Rathee, Thorben Funke, Avishek Anand, and Megha Khosla. BAGEL: A Benchmark for Assessing Graph Neural Network Explanations. 2022.

[6] Bastian Pfeifer, Anna Saranti, and Andreas Holzinger. GNN-SubNet: disease subnetwork detection with explainable graph neural networks. *Bioinformatics*, 38(Supplement_2):ii120–ii126, September 2022.

[7] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating Explanations for Graph Neural Networks. 2019.

[8] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? 2018.

[9] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. 2016.

[10] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. 2017.

[11] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. 2021.

[12] Sucharitha Rajesh. Evaluating the explainability of graph neural networks for disease subnetwork detection. 2024.

[13] Milchi Elena Oana. Modified gnn-subnet: leveraging local versus global graph neural network explanations for disease subnetwork detection. 2024.

[14] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. 2017.

[15] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. GraphSAINT: Graph Sampling Based Inductive Learning Method. 2019.