# Modified GNN-SubNet: leveraging local versus global Graph Neural Network explanations for disease subnetwork detection

**Elena-Oana Milchi**
**Supervisor(s): Dr. Megha Khosla, Dr. Jana Weber**
[1]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: Elena-Oana Milchi
Final project course: CSE3000 Research Project
Thesis committee: Dr. Megha Khosla, Dr. Jana Weber, Dr. Thomas Abeel

**Abstract**

As graph neural networks (GNNs) become more frequently used in the biomedical field, there is a growing need to provide insight into how their predictions are made. An algorithm that does this is GNN-SubNet, developed with the aim of detecting disease subnetworks in protein-protein interaction (PPI) networks. GNN-SubNet makes use of a sampling scheme to generate a global explanation in the form of a node mask which indicates each node's importance for all of the GNN's predictions on a dataset. The aim of this study is to validate GNN-SubNet by comparing it with an alternative approach of obtaining global explanations. Instead of obtaining the node mask via a sampling scheme, multiple (local) explanations are optimized per dataset sample, then the node masks are aggregated by either the mean (Mean Aggregation) or the median value (Median Aggregation) per node.

GNN-SubNet is compared with its two modifications firstly by analyzing which disease subnetworks each algorithm detects, and secondly by leveraging metrics devised to assess explainers for GNNs. The results show that all algorithms detect subnetworks associated with cancer. In terms of the metric scores, Mean Aggregation obtains explanations with the highest fidelity, however no algorithm obtains sparse explanations. The study also indicates that GNN-SubNet obtains variate outcomes over multiple runs, and as such the results may not be reproducible.

# 1 Introduction

Protein-protein interactions serve a wide range of purposes in the human organism, including aiding locomotion and metabolism regulation [1]. The alteration or malforming of proteins can generate disease phenotypes, such as cancer phenotypes [2]. Because the alteration of a protein can propagate throughout the network it is a part of, the process of linking proteins to certain diseases is complex and cannot be performed in isolation [3]. Thus, the analysis of data contained by protein-protein interaction (PPI) networks has proved to be of great use in disease subnetwork detection, as it can lead to discovering potentially malformed subnetworks pertaining to diseases such as cancer or Alzheimer's [2], [4].

In order to efficiently analyse the data encompassed by PPI networks, the task of disease subnetwork detection can be automated with the use of machine learning algorithms. In particular, graph neural networks (GNNs) have shown significant potential in the biomedical and healthcare fields [5], as they are able to cope with data comprising complex relationships [6]. The advantages of GNNs over other deep learning models in the medical field also include their ability to capture "hidden information in biological networks", as well as graph structures in input data in which they are not apparent, such as images [7, p.2].

Nevertheless, one drawback of GNNs is their lack of interpretability, which is especially relevant in bioinformatics [7]. When devising an algorithm to explain GNNs' predictions, multiple challenges arise. For instance, this algorithm should adapt to multiple GNN architectures and multiple classification tasks on graphs [8]. Thus, the intrinsic black-box nature of these deep learning models has led to the proposal of multiple explainers for them [5], [8]. Some models, such as Guided Backpropagation or Sensitivity Analysis, return the gradient of the GNN's prediction with respect to its input as an explanation. Others, such as GNNExplainer and Zorro, return explanations in the form of node masks indicating for each node in the graph how important it was for the model's prediction [9].

Pfeifer, Saranti and Holzinger [10] are among the first to bring together explainable graph models and disease subnetwork detection with GNN-SubNet, which leverages an explainer for GNNs to detect subgroups associated with cancer. The algorithm operates on

PPIs, which are modeled as graphs, where a protein corresponds to a node, and an edge corresponds to an interaction between two proteins. The authors investigate three different types of cancer: kidney renal clear cell carcinoma (KIRC), breast invasive carcinoma (BRCA) and lung adenocarcinoma (LUAD). GNN-SubNet extends GNNExplainer, which returns an explanation for a single sample in the input dataset (a local explanation), by optimizing a node mask with a continuous sampling scheme. The final node mask contains values indicating for each node in the graph topology how important it was for a GNN's predictions for an entire graph dataset, i.e. a global explanation.

The purpose of this research project is to validate GNN-SubNet's method of obtaining a global explanation by comparing it with an alternative approach. As such, the main question to be answered is:

How does the global-level explanation of a GNN obtained by GNN-SubNet compare with an aggregation of local-level explanations of the same model?

To answer this question, the following three sub-questions are proposed:

1. How can GNN-SubNet be modified to return a local-level explanation of a GNN?
2. How can multiple local-level explanations of a GNN be aggregated to obtain a global explanation?
3. How does GNN-Subnet compare with its proposed modification in terms of metrics devised to compare explainers for GNNs?

This research paper is structured as follows: firstly, the background information that is necessary to understand the key concepts of the study is presented in Section 2. Afterwards, the modifications made to GNN-SubNet, as well as the experimental setup that compares the old and new explainers, are elaborated on in Section 3. Afterwards, Section 4 presents and discusses the results of the performed experiments. Lastly, the ethical aspects of the project are considered in Section 5.

# 2 Background

## 2.1 Graph Neural Networks (GNNs)

GNNs represent a class of powerful Machine Learning tools that operate on graph data, more specifically, on the information encoded in each node of the graph. At every iteration of the algorithm, the embedding of a node is updated with an aggregation of the information stored in its neighbourhood [8]. After $k$ iterations, the embedding of a node represents an aggregation of the embeddings of its $k$-hop neighbourhood, i.e. the set of nodes which are at most $k$ edges away. The final step of the algorithm is a classification task based on the learnt embeddings [11].

GNNs can be differentiated by the methods used to aggregate and update the node embeddings after each iteration [8]. The Graph Isomorphism Network (GIN), a specific type of GNN, draws insight from the Weisfehler-Lehman (WL) Graph Isomorphism Test [12]. The WL-test aims to assess whether two graphs are topologically identical by comparing the hash values of their aggregated node labels. GINs return different hash values for topologically different graphs. This GNN architecture has been shown to outperform others in terms of train and test set accuracy on multiple datasets [12], which prompted the choice to implement GNN-SubNet as an explainer for a GIN [10].

## 2.2 Explainers for GNNs

In machine learning, explainability aims to understand why a model generates a certain prediction for a specified task [8]. For deep network models, this translates to insight either into correspondences between certain inputs and outputs, or into how data is represented within the network [13]. In a graph classification task, a *local explanation* relates to the prediction of a single graph instance from the dataset. On the other hand, *a global explanation* clarifies the predictions of all input graphs [8].

GNN-SubNet obtains a global explanation for a GIN by expanding on GNNExplainer, which returns a local explanation [10]. GNNExplainer finds the most relevant input features for a GNN's prediction as subgraphs, by perturbing the inputs and analysing the resulting outputs [8]. The explainer masks the node features of all nodes in the graph topology, and optimizes this mask using Gradient Descent. A threshold is then applied on the mask to find the most relevant input features for the prediction of a single dataset entry. GNNExplainer has been shown to outperform other alternatives in node classification tasks, as well as two graph classification tasks based on real-world datasets [14].

GNN-SubNet leverages an explainer for a GIN that is trained on a graph classification task, where all input graphs are PPI networks with the same topology. The node mask optimization is similar to that of GNNExplainer, however instead of optimizing on a single input graph it optimizes on a sample of dataset entries, which is continously updated. The node mask is then used to obtain weighted edges, which are processed by an algorithm that discovers communities in the PPI network. The final disease subnetwork is contained by the community ranked highest by this algorithm [10].

## 2.3 Evaluating explainers for GNNs

A significant proportion of the datasets which are mainly used in classification tasks by GNNs do not have a ground-truth explanation. Thus, it is difficult to evaluate explainers for such models [15], and multiple metrics were proposed to this end. Validity is one such metric. An explanation is considered valid if the prediction of a GNN does not change when taking into account only the nodes in the graph topology that were considered important by the explainer [16]. Another example is Fidelity+, which measures the difference in a GNN's accuracy when using all features versus removing those deemed important by the explanation [9]. On the other hand, Fidelity- implies removing the unimportant features and measuring the difference in the accuracy of the model [9].

Lastly, the BAGEL benchmark is meant to assess and compare different explainers for GNNs [5]. Four metrics for evaluating explainers for GNNs are proposed:

- *Faithfulness:* the ability to accurately explain the model's predictions. A high fidelity can mean either that the GNN's prediction does not change when only the features considered relevant by the explainer are selected, or that the explainer reports all, or enough relevant features for a GNN's prediction.

- *Sparsity:* the ability to generate non-trivial explanations. An explanation which reports the entire feature set as relevant for a prediction would have the lowest sparsity score.

- *Correctness:* the ability to recognize external correlations introduced in the network. This translates to how many decoys are detected by the explainer in a GNN that is re-trained with perturbed input data.

- *Plausibility:* how closely the explanations align with human rationale. This metric treats data on people's underlying opinions as the ground truth for evaluating an explanation.

# 3 Methods

This section elaborates on the changes made to GNN-SubNet as well as the experiments devised to compare it with the new explainers. Subsection 3.1 describes two alternatives to GNN-SubNet's method of obtaining a global explanation. The evaluation metrics chosen for comparing the three explainers are presented in Subsection 3.2. Finally, the experimental pipeline by which GNN-SubNet is compared with its modifications is elaborated on in Subection 3.3.

## 3.1 Modification of GNN-SubNet

### 3.1.1 RQ1: Obtaining a local explanation

In order to obtain a global explanation of a GNN, GNN-SubNet utilizes a node mask on the entire list of nodes. The node mask contains a single value for each node in the graph topology, which signifies how important the node was in the GNN's computation of predictions for the entire input dataset. For a trained GNN $\phi$, the goal is to minimize the expected value of the conditional entropy of its predicted label distribution $Y$ [14]. GNNExplainer [14] and, by extension, GNN-SubNet [10], achieve this by optimizing the node mask $N$ to minimize the following value via gradient descent:

$$\min_{N} - \sum_{c=1}^{C} \mathbb{1}[y = c] \log P_{\phi}(Y = y | G, X = X \times \sigma(N)). \tag{1}$$

Here, $C$ represents the set of all classification classes, and $G$ is an input graph. Moreover, $\mathbb{1}[y = c]$ is an indicator function equal to 1 for all predictions of the classification class $c$ and 0 otherwise. Lastly, $X$ is the node feature matrix, and $X \times \sigma(N)$ illustrates the application of the node mask, which is passed through the sigmoid function, to the node feature set [10].

When optimizing the node mask, the following loss function is employed:

$$\text{loss} = -\log P_{\theta}(Y = y | G, X = X \times \sigma(N)) + s \times \text{node\_feat\_reduce} - \frac{\sum_{e \in \text{entropy}} e}{|\text{entropy}|}, \tag{2}$$

where $N$ represents the node mask as in Equation 1 and $s$ is a configurable hyperparameter, in this case set to 1. For all arrays $A$, $|A|$ represents their length. Moreover, $node\_feat\_reduce$ represents the mean value of the node mask passed through the sigmoid function and is computed as:

$$\text{node\_feat\_reduce} = \frac{\sum_{n \in \sigma(N)} n}{|N|}. \tag{3}$$

Lastly, *entropy* measures the entropy of the node feature mask:

$$\text{entropy} = \text{ent} \times (\sigma(N) \times \log(\sigma(N) + \epsilon) + (1 - \sigma(N)) \times \log(1 - \sigma(N) + \epsilon)). \tag{4}$$

In this equation, ent is a configurable hyperparameter, set to 0.1 in GNN-SubNet. Lastly, $\epsilon$ is a positive value close to 0 that was introduced to avoid computing log(0). As observed, the loss takes into account the prediction of the model, as well as the values and entropy of the node feature mask.

GNN-SubNet performs gradient descent on a restricted sample of the input data. The size of this sample is customizable. Every 50 epochs, the sample is reinitialized with random input graphs from the dataset. The loss value in Equation 2 is computed for each entry in the current sample, and the sum of all of the loss values per entry is the final value used in backpropagation. Thus, the algorithm ensures that the value of the feature mask converges to a global explanation representing the entire dataset [10].

GNN-SubNet can be modified to return a local explanation by removing the sampling scheme mentioned above. The new node mask is optimized for a single graph input from the dataset, instead of computing a node mask over multiple graph instances at once. Thus, for a dataset of size $D$, the new algorithm optimizes $D$ node masks instead of a single one. In this context, the node mask reflects how important the nodes in the graph topology are in the prediction of a single input.

#### 3.1.2 RQ2: Obtaining a global explanation

Once all local node masks are computed, the global explanation is obtained by aggregating all local mask values per node. To this end, multiple methods can be employed. However, it is important to ensure that the aggregated node mask reflects all of the local explanations and, by exension, the entire input dataset. This project explores two different methods of local node mask aggregation:

- **Mean Aggregation:** Here, the final node mask value is computed as the mean of all local mask values per node. The mean is more suitable for situations in which these values form a normal distribution. However, there exists no guarantee that this will be the case for multiple datasets. Moreover, this method can be sensitive against outliers as it is much more easily influenced by extreme values.
- **Median Aggregation:** Instead of taking the mean of all node mask values, this method calculates the median of the distribution of mask values for a single node. This method is better suited for skewed distributions and more robust against outliers, making it more adaptable to multiple datasets.

For clarity purposes, the two modifications listed above will be referred to as *Mean Aggregation* and *Median Aggregation* throughout the rest of the document.

### 3.2 RQ3: Evaluation metrics

As stated in Subsection 2.3, the BAGEL benchmark provides four metrics by which to assess the performance of an explainer for a GNN. However, not all metrics are suitable for evaluating GNN-SubNet. For example, plausibility is not applicable in this context, as collecting data from experts in disease subnetwork detection is out of scope for this project. As such, the selected methods of assessment for GNN-SubNet as well as its modifications are Rate Distortion Based (RDT) Fidelity, Sparsity, Validity+ and Validity-.

### 3.2.1 Rate Distortion Based (RDT) Fidelity

This metric indicates how well a generated explanation can clarify a GNN's prediction. RDT Fidelity is measured by selecting a number of node features from the input graph and randomizing the rest. If the GNN's prediction shows no significant change, then the selected features are relevant in the prediction [5].

To compute this metric, the input graph features are perturbed a specific number of times. For the context of this project, the empirical choice is made to perturb the features 10 times. Thus, the score can be analysed over multiple runs and the computational time remains manageable. The final RDT fidelity score represents the proportion of times in which the prediction does not change for a distorted input graph. The perturbation of the input graph with the feature set $X$ and the explanation in the form of a node mask *M(S)* is performed using the following formula:

$$Y_S = X \odot M(S) + Z \odot (\mathbb{1} - M(S)). \tag{5}$$

Here, "$\odot$ denotes an element-wise multiplication, and $\mathbb{1}$ a matrix of ones with the corresponding size" [5, p.4]. Moreover, $Z$ represents a noise distribution that perturbs the input graph. An explanation that is stable in the face of input perturbation will have a high RDT fidelity score [5].

It should be noted that RDT fidelity is computed over a single input graph. For a dataset of $D$ graphs, $D$ values for this metric will be generated.

### 3.2.2 Sparsity

When evaluating the explanation of a GNN, sparsity refers to how succint it is. To this end, the sparsity of an explanation is computed as its "entropy over the mask distribution" [5, p.5]:

$$H(p) = -\sum_{\phi \in M} p(\phi) \log(p(\phi)). \tag{6}$$

Here, $M$ represents the set of all node features, and p is the normalized distribution of the node mask. To limit the sparsity score between 0 and 1, the final score is calculated with the formula:

$$1 - \frac{H(p)}{\text{max\_entropy}}, \tag{7}$$

where *max_ entropy* denotes the maximum possible value that the entropy of a node mask can have. This corresponds to an explanation consisting of all nodes in the graph topology, which is computed as $-\log(\frac{1}{|M|})$. It is more favourable that explanations have a lower entropy and, by extension, a higher sparsity score. For $k$ runs of the explainer on the input dataset, the sparsity score will be evaluated $k$ times.

### 3.2.3 Validity+ and Validity-

Validity+ and Validity- are adapted from the notions of validity, as well as Fidelity+ and Fidelity-, which are presented in Subsection 2.3. Taking this information into account, Validity+ averages the unimportant feature values, whereas Validity- averages the important feature values. In order to differentiate important features from unimportant features, a

*threshold* $t$ is employed such that the top $t\%$ of the node mask values are considered important, and the rest, unimportant.

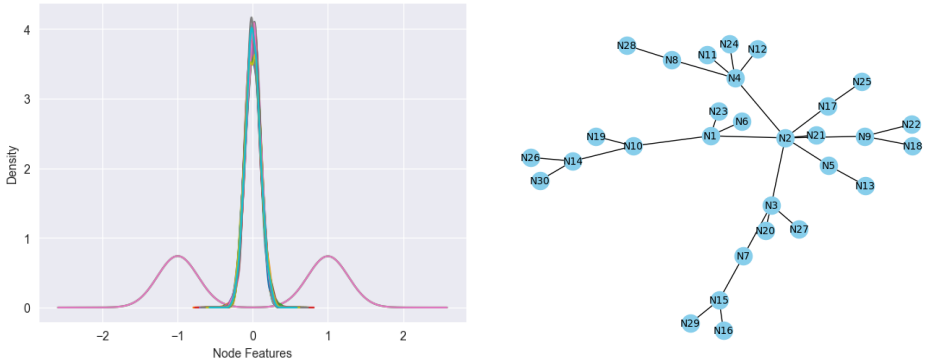
A good explanation should have a low Validity+ score and a high Validity- score. This is because the prediction should change considerably more when removing the important nodes than when removing those that are deemed unimportant.

## 3.3 Experimental Setup

### 3.3.1 Input data used

Throughout the experimental pipeline, two datasets are used: one containing synthetically generated graphs, and another containing data on PPI networks retrieved from The Cancer Genome Atlas (TCGA). It is important to note that, in both datasets, one entry comprises the node features of a single graph instance. Moreover, all graph entries have the same topology, and it is only the node feature values that differ among multiple graph instances. This facilitates the explainers' process of reaching a global mask that is representative of the entire dataset.

The first dataset represents a series of graph samples used to perform sanity checks on GNN-SubNet. Each graph entry is a synthetically generated network of 30 nodes [10]. In this study, the dataset is employed to verify the implementation of Gradient Descent by which a local explanation is obtained. Figure 1 visualizes this data.

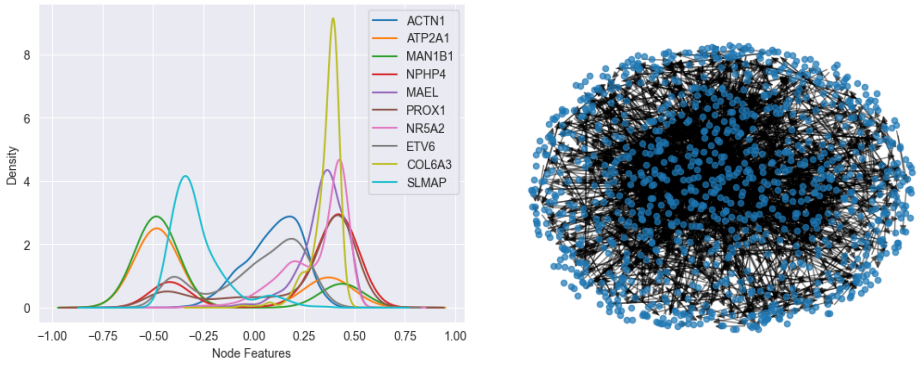



(a) (b)

Figure 1: A visualization of the synthetic dataset. (a) plots the numeric distribution of the feature values of all 30 nodes. The y-axis indicates the estimated probability density of a node's features. (b) shows the graph topology corresponding to each dataset sample.

The TCGA dataset contains mRNA and protein methylation features [17], which are retrieved from The Cancer Genome Atlas[1], and is used in GNN-SubNet's task of disease subnetwork detection [10]. The node features contain genetic profiles relating to Kidney Renal Cell Carcinoma (KIRC) and are much more complex and variate than the synthetic dataset. As such, the graph models a PPI with cca. 4780 proteins and more than one million protein interactions. In this study, the dataset is used when comparing GNN-SubNet and its two modifications in terms of their performance in disease subnetwork detection, as well as the BAGEL metrics. A visualization of the data is shown in Figure 2.

[1] `https://portal.gdc.cancer.gov/`




(a) (b)

Figure 2: A visualization of the TCGA KIRC dataset. (a) plots the numeric distribution of the node feature values of the first 10 proteins. As before, the y-axis indicates the estimated probability density. (b) illustrates a visualization of a subset of the graph. Each node represents a protein, and each edge represents a protein interaction.

### 3.3.2 Experimental Pipeline

The comparison of GNN-SubNet with the proposed modifications is achieved in multiple steps. Firstly, the correctness of the method by which a local explanation is obtained is verified by plotting the loss function value from performing gradient descent. The GIN model is trained on the synthetic dataset for 20 epochs. Then, GNN-SubNet and the Mean Aggregation modification are run for 300 epochs and the gradient descent loss value per epoch is recorded. The implementation is deemed functional when the plot shows a gradual decrease in the loss value and eventual convergence.

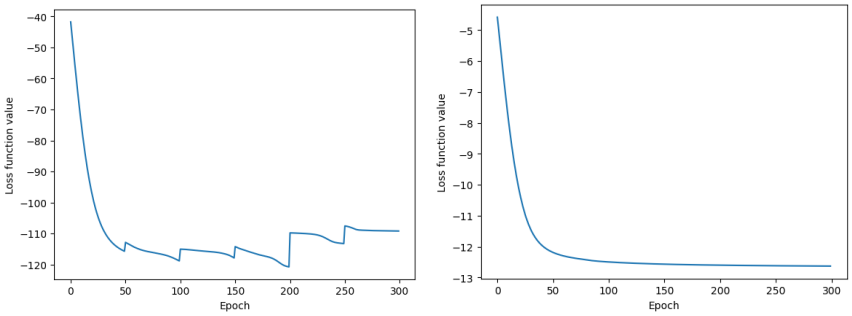
To appraise the global explanations obtained by the new explainers, the node masks are employed for disease subnetwork detection. Firstly, a GIN model is trained for 20 epochs on the test set of the TCGA dataset, which contains 200 proteins. The procedure of obtaining this test set is described more in detail by Pfeifer, Saranti and Holzinger [10]. Then, the selected explainer is run for 300 epochs in 10 iterations. The resulting global explanation is employed in a community detection algorithm [10], which results in a series of subnetworks, each with an associated importance score. This entire process is repeated 10 times, and for each iteration the obtained subnetwork with the highest importance score is recorded. Afterwards, for each reported subnetwork, its frequency among the ten runs is analysed and compared to that of other protein subgroups.

The pipeline that assesses the scores for RDT fidelity, Sparsity, Validity+ and Validity- is similar to that of the experiment performing disease subnetwork analysis. However, instead of analysing the obtained subnetworks, the metric scores are evaluated on the resulting node masks. Firstly, the mean RDT Fidelity over the input dataset, as well as the mean Sparsity over the explainer runs, are computed. Validity+ and Validity- are evaluated at the thresholds $t = 30.0$ and $t = 50.0$. This choice was made to appraise a hard masking approach that is more lenient, which considers the proteins ranked in the top half as important, as well as a stricter approach, which only considers the top 30%. This entire process is run 10 times for all three explainers and the scores for each metric are stored per run in separate files. After all iterations are finished, the scores of all runs are aggregated (for both Validity

scores, the aggregation is done per threshold) and their mean value is obtained.

# 4 Results and Discussion

## 4.1 RQ1: Obtaining a local explanation




(a) (b)

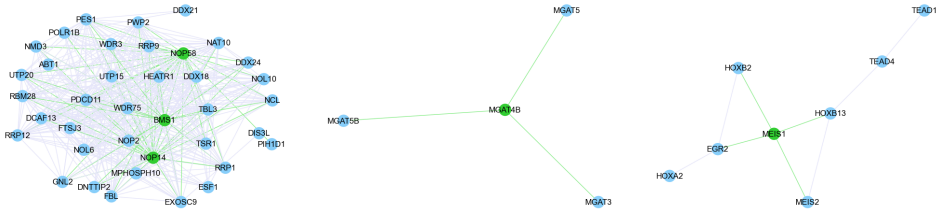
Figure 3: Side-by-side comparison of the values of the loss function when performing Gradient Descent on the synthetic dataset. The plot shown in (a) corresponds to GNN-SubNet, and the plot shown in (b) corresponds to the Mean Aggregation modification. The x-axis indicates the epoch for which the value is recorded. The y-axis represents the value of the loss function recorded for the corresponding epoch.

Upon analysing the gradient descent loss value plots shown in Figure 3, it can be noticed that, whereas the loss function decreases gradually in the plot corresponding to the modified GNN-SubNet, that of the original algorithm does not. This is likely due to the sampling scheme employed by GNN-SubNet. The sample that aids in the optimisation of the node mask is reset every 50 epochs, but the optimized node mask only changes to accomodate to one sample at a time. As a result, upon reinitialization, the loss function loses its monotony.

It can be observed that the loss is negative in both plots. This is likely due to the manner in which the loss function is computed in the Gradient Descent algorithm, which is shown in Equation 2. The plotted loss function value is likely negative due to the fact that the mean value of the node mask is smaller than the sum of the forward function value and the node mask entropy. Moreover, the loss values showcased by GNN-SubNet are far lower than those of its modification. This is due to the fact that GNN-SubNet computes its final loss as the sum of the losses for each entry in its current sample, which leads to a much lower final value.

## 4.2 RQ2: Performance in disease subnetwork detection

A subnetwork analysis as described in Subsection 3.3 shows that GNN-SubNet and its two modifications return multiple protein groups over ten runs. Three of the most frequently discovered subnetworks are illustrated in Figure 4. Among all algorithms, the most common subset of the returned subnetworks is a group of 38 proteins, more specifically:




(a) (b) (c)

Figure 4: Plots of three of the most frequently discovered subgroups among ten runs of GNN-SubNet and its two modifications. Each node models a protein and an edge models a protein interaction. Note that only the edges with an importance score above the 95th percentile are shown. The nodes with a connectivity above the 95th percentile, as well as their connected edges, are highlighted in green. For (a), these are NOP14, NOP58 and BMS1. For (b) this is MGAT4B and, for (c), MEIS1.

*ABT1, BMS1, DCAF13, DDX18, DDX21, DDX24, DIS3L, DNTTIP2, ESF1, EXOSC9, FBL, FTSJ3, GNL2, HEATR1, MPHOSPH10, NAT10, NCL, NMD3, NOL10, NOL6, NOP14, NOP2, NOP58, PDCD11, PES1, PIH1D1, POLR1B, PWP2, RBM28, RRP1, RRP12, RRP9, TBL3, TSR1, UTP15, UTP20, WDR3, WDR75*

From this list, NOP14 is one of the proteins with the highest connectivity as noted in Figure 4. The protein has been shown to potentially cause some cancer types, and supress the tumor of others, and it has a significantly higher expression in tumor tissues compared to normal tissues. However, in terms of KIRC, NOP14 has been associated with a good prognosis in terms of the patients' overall survival rate [18]. On the other hand, NOP58, another protein with a high connectivity in the identified subnetwork, has shown a significant expression in multiple types of cancerous tumours, including KIRC tumours [19].

Another subnetwork common to GNN-SubNet and the Mean Aggregation Algorithm contains four proteins, i.e. *MGAT3, MGAT4B, MGAT5 and MGAT5B*. Among these proteins, MGAT5 is considered an important gene in cancer classification [20]. It has been shown that a high expression of MGAT5 can potentially indicate a poorer prognosis for the survival rate of clear-cell Renal Cell Carcinoma (ccRCC) patients [21], where ccRCC is a subtype of KIRC. MGAT4B, which has the highest connectivity in the subgraph, is also relevant in cancer classification. However, a differential expression level when compared with other genes such as MGAT5 is not detected in the case of KIRC [20].

The authors of GNN-SubNet have indicated that the subnetwork of greatest importance reported by them in one run was formed of the genes *EGR2, HOXA2, HOXB13, HOXB2, MEIS1, MEIS2, TEAD1 and TEAD4* [10]. Upon running the algorithm ten times on the TCGA dataset, this gene group was never reported as the most important subnetwork. On the other hand, both the Mean Aggregation and the Median Aggregation modifications have identified this subnetwork once in the ten runs. From this subnetwork, HOXB13 has been reported as potentially significant in the genesis and progression of Renal Cell Carcinoma (RCC) tumours. HOXB13 also directly interacts with MEIS1 and MEIS2, which aids tumour suppression, and the disruption of this linkage can generate cancerous cells [22].

Overall, the results indicate that GNN-SubNet and its two modified versions are able to identify subnetworks containing proteins that have a higher expression level in cancer

patients. Nevertheless, upon 10 iterations of this experiment, each explainer has obtained highly variate subnetworks. It is thus advised that the results shown in the section be further investigated, as the reproducibility of these three algorithms is not guaranteed. This limitation is elaborated on in Subsection 4.4.

## 4.3 RQ3: Comparison in terms of the selected metrics

The final experiment described in Subsection 3.3 resulted in a series of values corresponding to RDT Fidelity, Sparsity, Validity+ and Validity- scores. These metric scores are shown in Table 1.

Table 1: Mean RDT Fidelity, Sparsity and, for thresholds 30.0 and 50.0, Validity+ And Validity-, recorded on the TCGA (KIRC) dataset. The metric scores are computed for GNN-SubNet as well as its two modifications, which are described in Section 3. The numbers in bold show the best metric scores among the three different explainers.

| **Explainer** | **GNN Acc.** | **Mean Fidelity** | **Mean Sparsity** | **Thres. Value** | **Mean Validity+** | **Mean Validity-** |
|---|---|---|---|---|---|---|
| GNN-SubNet | 75.0 | 0.855625 | **0.042664** | 30.0 | **0.25875** | 0.73625 |
| | | | | 50.0 | 0.47125 | 0.85375 |
| Mean Aggregation | 75.75 | **0.861375** | 0.019166 | 30.0 | 0.32375 | **0.8150** |
| | | | | 50.0 | **0.39750** | **0.8725** |
| Median Aggregation | 73.0 | 0.837375 | 0.038665 | 30.0 | 0.39375 | 0.80750 |
| | | | | 50.0 | 0.45500 | 0.83375 |

Overall, the Mean Aggregation modification outperforms the other two explainers in terms of some metrics. For RDT Fidelity, the results show a 0.67% increase compared to GNN-SubNet. This signifies that a global explanation that is guaranteed to take into account all graph instances can have a higher fidelity than one that is aggregated via a sampling algorithm. However, the Median Aggregation variant performs the worst out of the three explainers, with Mean Aggregation outperforming it by 2.86%. This could indicate that the median value of all node masks may not be the most accurate aggregation method. The mean value is also usually employed when computing values with little chances of being influenced by outliers. As such, the results may reflect the distribution of the aggregated local explanations and, by extension, the node features in the input dataset.

The existing literature contains multiple assessments of GNN explainers' performances in terms of RDT Fidelity. For example, Rathee et al. [5] analyse the RDT Fidelity of a range of explanations for a GIN performing a graph classification task. The results show that GNNExplainer, which GNN-SubNet extends, is the best performing model when explaining the GNN's predictions of chemical compound mutagenic effects. However it showcases the worst performance when explaining the GIN's protein classification task [5]. This can indicate that GNNExplainer's RDT Fidelity score may differ greatly depending on the dataset that is used, and that GNN-SubNet, together with its modifications, inherit this quality as well.

In terms of Sparsity, it can be noted that all explainers have obtained very low scores on the TCGA dataset. This implies that the node mask value distributions for all nodes in the graph topology has a high entropy. A possible explanation for this result is represented by the fact that GNN-SubNet randomly initializes the node mask values for the Gradient Descent algorithm. This limitation is further detailed in subsection 4.4.

The Mean Aggregation modification showcases the worst sparsity score out of all explainers. Compared to the original algorithm, it shows a 55% decrease in score. In this respect, Median Aggregation obtains a much better score. This could be due to the fact that the median value is less likely to be influenced by exceedingly high node mask values, which could lead to lower node importance values overall. However GNN-SubNet is still shown to obtain the sparsest explanations.

The low Sparsity scores may also be an issue derived from GNNExplainer. Funke et al. [16] compare multiple explainers for GNNs performing a classification task on datasets of scientific publications. Their results report that, on all datasets, when explaining a GIN's decisions, GNNExplainer showcases the worst Sparsity scores.

In terms of Validity+, GNN-SubNet shows the best performance at threshold 30.0, but is outperformed by the Mean Aggregation modification at threshold 50.0. In terms of Validity- Mean Aggregation shows a better performance by 10% compared to GNN-SubNet at threshold 30.0, and 2.19% at threshold 50.0. On the other hand, Median Aggregation outperforms GNN-SubNet at threshold 30.0, but shows the worst performance at threshold 50.0. Thus, though Mean Aggregation showcases the best Validity- scores, there does not seem to be a clear winner in terms of the Validity+ metric.

It can be observed for all explainers that increasing the threshold value when computing validity scores (more concretely, an increase in the amount of nodes considered "important") will lead to an increase in the metric value, both for Validity+ and Validity-. For Validity+, the results indicate that the metric score worsens the more feature values remain unchanged. For Validity-, the score becomes better the less "unimportant" features maintain their original value.

It is important to note that, in most cases, the metric score distributions have showcased a high variance over ten runs of the three explainers. This can also be observed in the visualizations in Appendix A. As such, it is important that the presented results and their subsequent interpretation be viewed critically, and that the performance of the three explainers be further analysed.

## 4.4 Limitations

Since Mean Aggregation and Median Aggregation build upon the original implementation of GNN-SubNet, the limitations of this algorithm apply to its extensions as well. It is important to note that GNN-Subnet randomly initializes the values for the node mask to optimize via gradient descent. Different initial values will lead to different optimizations and as such the final node mask will differ from one iteration of the algorithm to another. Consequently, both the final obtained subnetworks and the metric scores will vary over multiple runs, which can affect the reporoducibility of the results. This becomes clearer when studying the plots in Appendix A; most of the distributions of the metric scores are skewed and indicate a high variance. This limitation can be overcome by setting a seed for the randomization of the initial node mask values, which will lead to a deterministic and reporoducible algorithm. This random seed will then need to be tuned in order to find the best value for the initial node mask.

Moreover, due to time constraints, the hyperparameters employed in the entire pipeline of the algorithm as described in Section 3 have been set to the same values as GNN-SubNet. It is recommended that a grid search be performed to find the best values for how many epochs gradient descent should be run for, or how many iterations of the three explainers will lead to the best metric scores. A tuning of the hyperparameters used to compute the

loss function could also be performed for the two modifications of GNN-SubNet.

Lastly, the Median Aggregation modification can be extended to take into account multiple quantile values. Currently, time constraints impede a further fine-tuning of the quantile value to be employed when aggregating all local node masks. The quantile value that leads to the best metric scores can be explored in a future extension of this project.

# 5 Responsible Research

Genomic research, which includes the study of genes coding for proteins [23], is a field which does not allow for errors. False positives or negatives of this kind of research incur a great toll on the individuals whose genome data is employed [23]. The usage of genomic data, which represents sensitive information, can also lead to data protection issues. Moreover, from a technological perspective, it is paramount that the developed code be correct as well as reproducible. It is also important to ensure that the process of obtaining results upholds academic integrity. The rest of this section will highlight the extent to which each of these matters has been addressed throughout the research pipeline.

Firstly, the data sources employed in the project are retrieved in compliance with principles of responsible research. As such, the data obtained from The Cancer Genome Atlas upholds the FAIR principle [24] in that its public accessibility [2] renders it findable as well as available. The dataset is also interoperable and reusable as the project repository contains a python notebook which explores it in depth. Additionally, The Cancer Genome Atlas highlight guidelines for ethical data access and usage. To access sensitive data that can be traced back to a specific person, researchers must agree to comply with norms on accessing secure and private information [25].

Secondly, the modification of GNN-SubNet is verified in the following manner: the process of obtaining a local explanation is assessed by analysing the loss value over multiple epochs. Then, the process of obtaining global explanations is verified by appraising their usability in disease subnetwork detection, as well as by leveraging a selected subset of metrics designed to assess explainers for GNNs.

One drawback of this study as described in Subsection 4.4 is its limited reproducibility. The high variance in results over multiple runs of GNN-Subnet gives rise to concerns with regard to how feasible it is to replicate the results obtained for the explainer's proposed modifications. Nevertheless, to the extent possible, the project is designed with careful consideration for reproducibility by future researchers. To make the code understandable and easy to reuse, all modifications to GNN-SubNet are properly documented. The python notebooks that run all experiments highlighted in section 3.3 and visualize the input data displayed in 3.3.1 are available in the repository to facilitate the replication of the experiments. Additionally, to maintain the transparency of the research process, version control is assured via the use of GIT, and the project repository is publicly available[3].

# 6 Conclusions and Future Work

The purpose of this research project is to investigate how to modify the means by which a global explainer for a GNN generates an output that is representative of an entire dataset. Thus, the main questions to be answered are firstly, how to modify GNN-SubNet such

[2] `https://portal.gdc.cancer.gov/`

[3] `https://github.com/Oana-MO3/GNN-SubNet_modified.git`

that it returns a local explanation and, secondly, how to aggregate the local explanations to obtain a global explanation. Moreover, the project aims to validate the algorithm's sampling method by comparing it with the new approach in terms of four metrics devised for assessing explainers for GNNs.

GNN-SubNet obtains a global explanation by optimizing a node mask via gradient descent with the use of a continuous sampling scheme. The modification of this algorithm lies in removing the sampling scheme such that a local node mask is now optimized per dataset sample. Then, all of the obtained local explanations are aggregated per node either by their mean or their median value to obtain a global explanation.

To investigate how GNN-SubNet compares with its modifications, the three algorithms are run on data provided by The Cancer Genome Atlas relating to Kidney Renal Carcinoma (KIRC) to generate disease subnetworks. An analysis of the obtained disease subnetworks reveals that all explainers uncover proteins related to cancer. Another experiment involves comparing the three algorithms in terms of metrics meant to assess explainers for GNNs. The results indicate that changing the method by which a global explanation is obtained can lead to a slight improvement in terms of RDT Fidelity scores, however it leads to significantly lower Sparsity scores.

One recommendation for future research lies in exploring whether GNN-Subnet can be modified to compute an edge mask, rather than a node mask. This would remove the need to aggregate the node mask values into an edge mask when performing disease subnetwork detection. As such, assessing the model's behaviour with such a change can prove to be an interesting direction for the future.

Lastly, the research described paves the way for further investigations into the workings of GNN-SubNet. This study shows that one limitation of the explainer lies in its highly variate results. It is thus recommended that a method of combating this limitation be investigated. In order for the research to be reproducible, ideally, GNN-SubNet should return a single subnetwork per dataset. A suggestion is to set a random seed for the generation of the initial node mask values in the optimization process. On a more general level, this issue poses the need for further research into low variance explainable graph models for biological applications.

# 7 Acknowledgements

I would like to extend my gratitude to my supervisors, Dr. Khosla and Dr. Weber, for their guidance throughout the duration of the project, without which I would not have been able to produce this paper. I would also like to thank my colleague, Sucharita Rajesh, for analysing the most suitable evaluation metrics to use when assessing GNN-SubNet, as well as providing the implementation of these metrics.

# References

[1] P. Braun and A. Gingras, "History of protein-protein interactions: From egg-white to complex networks", *Proteomics*, vol. 12, no. 10, pp. 1478–1498, May 2012, ISSN: 1615-9853, 1615-9861. DOI: 10.1002/pmic.201100563.

[2] R. Schulte-Sasse, S. Budach, D. Hnisz, and A. Marsico, "Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms", *Nat. Mach. Intell.*, vol. 3, no. 6, pp. 513–526, Apr. 2021, ISSN: 2522-5839. DOI: 10.1038/s42256-021-00325-y.

[3] M. Agrawal, M. Zitnik, and J. Leskovec, "Large-scale analysis of disease pathways in the human interactome", in *Pacific Symposium on Biocomputing 2018: Proceedings of the Pacific Symposium*, World Scientific, 2018, pp. 111–122.

[4] D. Armanious, J. Schuster, G. A. Tollefson, *et al.*, "Proteinarium: Multi-sample protein-protein interaction analysis and visualization tool", *Genomics*, vol. 112, no. 6, pp. 4288–4296, Nov. 2020, ISSN: 08887543. DOI: 10.1016/j.ygeno.2020.07.028.

[5] M. Rathee, T. Funke, A. Anand, and M. Khosla, "BAGEL: A Benchmark for Assessing Graph Neural Network Explanations", 2022, arXiv:2206.13983.

[6] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model", *IEEE T. Neural Networ.*, vol. 20, no. 1, pp. 61–80, 2009. DOI: 10.1109/TNN.2008.2005605.

[7] X.-M. Zhang, L. Liang, L. Liu, and M.-J. Tang, "Graph neural networks and their current applications in bioinformatics", *Front. Genet.*, vol. 12, p. 690 049, 2021.

[8] J. Kakkad, J. Jannu, K. Sharma, C. Aggarwal, and S. Medya, "A Survey on Explainability of Graph Neural Networks", 2023, arXiv:2306.01958.

[9] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in Graph Neural Networks: A Taxonomic Survey", *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–19, 2022, ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2022.3204236.

[10] B. Pfeifer, A. Saranti, and A. Holzinger, "GNN-SubNet: Disease subnetwork detection with explainable graph neural networks", *Bioinformatics*, vol. 38, no. Supplement_2, pp. ii120–ii126, Sep. 2022, ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btac478.

[11] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. Wiltschko, "A Gentle Introduction to Graph Neural Networks", *Distill*, vol. 6, no. 8, 10.23915/distill.00033, Aug. 2021, ISSN: 2476-0757. DOI: 10.23915/distill.00033.

[12] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?", 2019, arXiv:1810.00826.

[13] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining Explanations: An Overview of Interpretability of Machine Learning", in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, Turin, Italy: IEEE, Oct. 2018, pp. 80–89, ISBN: 978-1-5386-5090-5. DOI: 10.1109/DSAA.2018.00018.

[14] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks", *Adv. Neur. In.*, vol. 32, 2019.

[15] J. Tan, S. Geng, Z. Fu, *et al.*, "Learning and Evaluating Graph Neural Network Explanations based on Counterfactual and Factual Reasoning", in *Proceedings of the ACM Web Conference 2022*, Virtual Event, Lyon France: ACM, Apr. 2022, pp. 1018–1027, ISBN: 978-1-4503-9096-5. DOI: 10.1145/3485447.3511948.

[16] T. Funke, M. Khosla, M. Rathee, and A. Anand, "Zorro: Valid, Sparse, and Stable Explanations in Graph Neural Networks", Mar. 2022, arXiv:2105.08621.

[17] S. V. Vasaikar, P. Straub, J. Wang, and B. Zhang, "Linkedomics: Analyzing multi-omics data within and across 32 cancer types", *Nucleic acids res.*, vol. 46, no. D1, pp. D956–D963, 2018.

[18] C. Lu, W. Liao, Y. Huang, Y. Huang, and Y. Luo, "Increased expression of nop14 is associated with improved prognosis due to immune regulation in colorectal cancer", *BMC gastroenterol.*, vol. 22, no. 1, p. 207, 2022.

[19] J. Wang, R. Huang, Y. Huang, Y. Chen, and F. Chen, "Overexpression of nop58 as a prognostic marker in hepatocellular carcinoma: A tcga data-based analysis", *Adv. ther.*, vol. 38, no. 6, pp. 3342–3361, 2021.

[20] J. Ashkani and K. J. Naidoo, "Glycosyltransferase gene expression profiles classify cancer types and propose prognostic subtypes", *Sci. rep-UK*, vol. 6, no. 1, 2016.

[21] Y. Liu, H. Liu, W. Liu, W. Zhang, H. An, and J. Xu, "$\beta$1, 6-N-acetylglucosaminyltransferase V predicts recurrence and survival of patients with clear-cell renal cell carcinoma after surgical resection", *World j. urol.*, vol. 33, pp. 1791–1799, 2015.

[22] C. VanOpstall, S. Perike, H. Brechka, *et al.*, "Meis-mediated suppression of human prostate cancer growth and metastasis through hoxb13-dependent regulation of proteoglycans", *Elife*, vol. 9, 2020.

[23] J. Mathaiyan, A. Chandrasekaran, and S. Davis, "Ethics of genomic research", *Perspect. Clin. Res.*, vol. 4, no. 1, 2013, ISSN: 2229-3485. DOI: 10.4103/2229-3485.106405.

[24] "The FAIR Guiding Principles for scientific data management and stewardship", *Sci. Data.*, vol. 3, no. 1, Mar. 2016, ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18.

[25] N. C. I. (NCI). "Tcga ethics and policies". (Mar. 2019), [Online]. Available: https://www.cancer.gov/ccg/research/genome-sequencing/tcga/history/ethics-policies.

# A A visualization of the comparison metric scores

Figure 5 highlights a visual representation of the distribution of metric scores for GNN-SubNet, as well as its two modifications. The experimental pipeline that produces these results is described in Subsection 3.3.

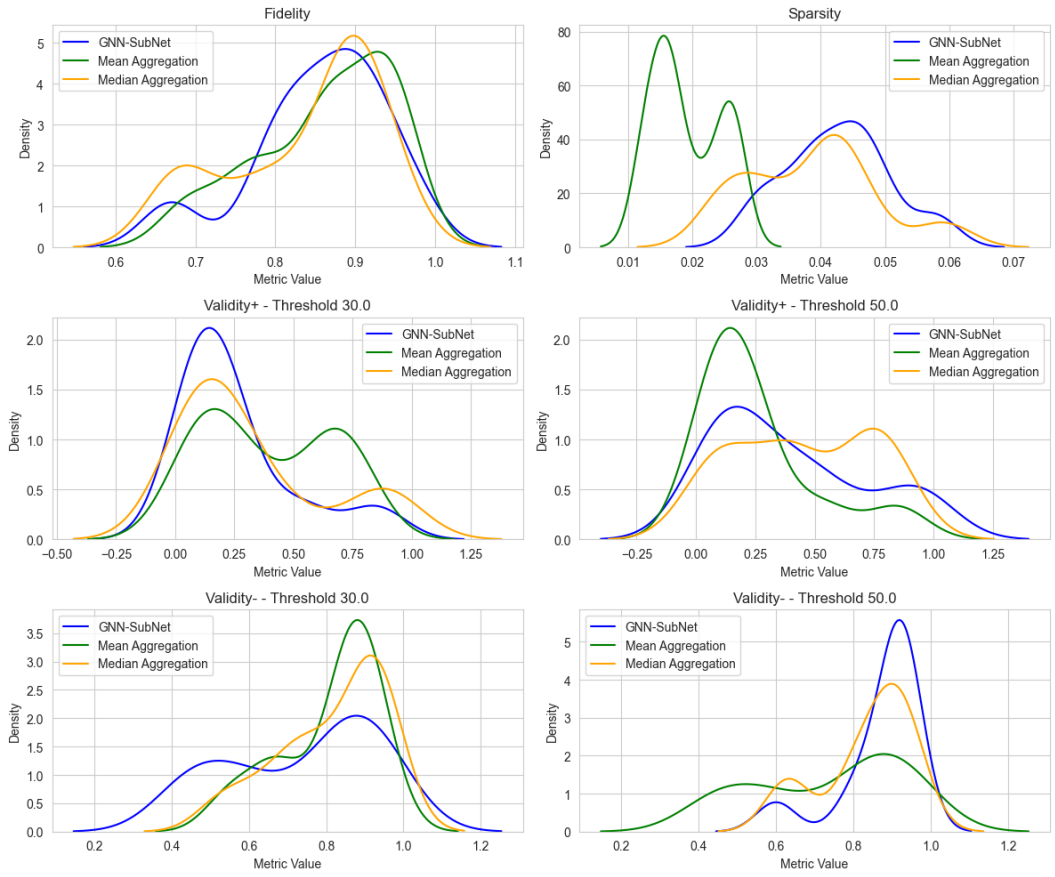



Figure 5: The numeric distributions of RDT Fidelity, Sparsity, Validity+ and Validity- scores (at thresholds 30.0 and 50.0) for GNN-SubNet and its two modified versions. The values are recorded over 10 runs of each explainer. For all plots, the x-axis indicates the metric values. The y-axis indicates the estimated probability density of the metric values.