On the Probability of Necessity and Sufficiency of Explaining Graph Neural Networks: A Lower Bound Optimization Approach

Ruichu Cai[a,b], Yuxuan Zhu[a,*], Xuexin Chen[a], Yuan Fang[c], Min Wu[d], Jie Qiao[a,b], Zhifeng Hao[e]

[a]*School of Computer Science, Guangdong University of Technology, Guangzhou 510006, China*
[b]*Peng Cheng Laboratory, Shenzhen 518066, China*
[c]*School of Computing and Information Systems, Singapore Management University, 178902, Singapore*
[d]*Institute for Infocomm Research (I^{2}R), A*STAR, 138632, Singapore*
[e]*College of Science, Shantou University, Shantou 515063, China*

Abstract

The explainability of Graph Neural Networks (GNNs) is critical to various GNN applications, yet it remains a significant challenge. A convincing explanation should be both necessary and sufficient simultaneously. However, existing GNN explaining approaches focus on only one of the two aspects, necessity or sufficiency, or a heuristic trade-off between the two. Theoretically, the Probability of Necessity and Sufficiency (PNS) holds the potential to identify the most necessary and sufficient explanation since it can mathematically quantify the necessity and sufficiency of an explanation. Nevertheless, the difficulty of obtaining PNS due to non-monotonicity and the challenge of counterfactual estimation limit its wide use. To address the non-identifiability of PNS, we resort to a lower bound of PNS that can be optimized via counterfactual estimation, and propose a framework of Necessary and Sufficient Explanation for GNN (NSEG) via optimizing that lower bound. Specifically, we depict the GNN as a structural causal model (SCM), and estimate the probability of counterfactual via the intervention under the SCM. Additionally, we leverage continuous masks with a sampling strategy to optimize the lower bound to enhance the scalability. Empirical results demonstrate that NSEG outperforms state-of-the-art methods, consistently generating the most necessary and sufficient explanations. The implementation of our NSEG is available at `https://github.com/EthanChu7/NSEG`.

Keywords:
Explainability, Graph Neural Networks, Causality, Explainable AI, Interpretability

1. Introduction

Graph Neural Networks (GNNs) differentiate themselves from neural networks designed for Euclidean data by not only learning feature information but also capturing graph structures through the message-passing mechanism [23, 18, 55, 49, 60, 24, 17, 53, 10]. This unique characteristic has facilitated the successful application of GNNs in various domains, including social recommendation [15, 19], molecule discovery [44, 45], and fraud detection [54, 14]. However, GNNs with high complexity are still considered black-box models [4, 35], which limits their applications in many real-life related domains like

medicine and healthcare [39]. Although numerous studies have been proposed to explain neural networks for Euclidean data [41, 31, 46, 21], such approaches are usually not suitable for GNNs as they cannot explain the graph structures well. Hence, the explainability of GNNs remains an open challenge.

Current approaches for explaining GNNs mainly search for three types of explanation, i.e., necessary explanation, sufficient explanation, and the heuristic trade-off explanation between necessity and sufficiency. First, the approaches [27, 51, 29] searching for a necessary explanation seek to identify a group of necessary features that will change the prediction if one performs a perturbation. Although necessity is important for the explanation, the lack of sufficiency can result in the incompleteness of the explanations. For instance, consider a chat group classification task in which the hobby and social connections of each member are provided and we aim to explain why the instance given in Figure 1 is predicted as "Sport Lover Group". For the necessary explanation shown on the top-right of Figure 1, only a small set of soccer lovers or basketball lovers are considered as explanations. However, it is insufficient as some basketball lovers are missing. In
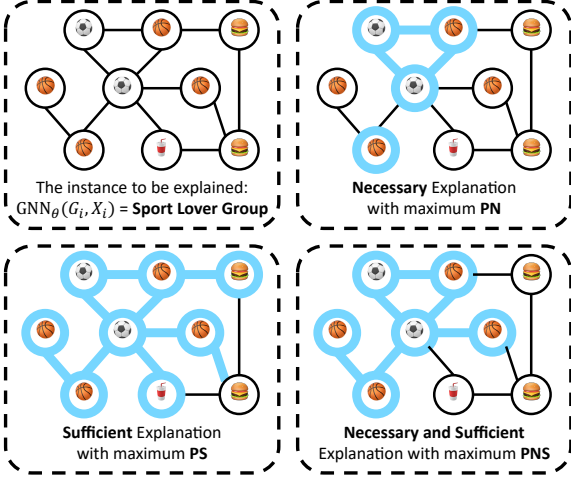
Figure 1: Explanations for the prediction "Sport Lover Group", which are highlighted in blue. Each node is a member in the group whose node features are their hobby, denoted by the icons. PN, PS, PNS refer to the probability of necessity, the probability of sufficiency, and the probability of necessity and sufficiency, respectively.

dition, to enable tractable optimization, continuous masks with a sampling strategy are used to optimize the lower bound of PNS. By combining these techniques, we propose a framework of **N**ecessary and **S**ufficient **E**xplanation for **G**NNs (NSEG), via maximizing the lower bound of PNS.

Our contributions can be summarized as follows:

- We propose NSEG to generate necessary and sufficient explanations for GNNs, by optimizing a lower bound of PNS via counterfactual estimation.

- We depict the GNN as an SCM such that the counterfactual estimation is tractable via intervention on the SCM. We further leverage a continuous mask with a sampling strategy to optimize the lower bound of PNS, making the optimization tractable by relaxing the discrete explanation to a continuous case.

- Our experiments show that the explanations from NSEG are the most necessary and sufficient, and both aspects are critical to the generation of explanations.

2. Related Work

2.1. Graph Neural Networks

Graph Neural Networks (GNNs) [23, 18, 55, 49, 60, 24, 17, 53, 10, 12, 59, 28, 13, 5, 26] have demonstrated tremendous success in various real-world applications, e.g., social recommendation [15, 19], molecule discovery [44], etc. Inspired by Convolutional Neural Networks (CNN) [25], graph convolution is applied in graph data to make the networks more efficient and convenient. Over the years, various convolutional GNNs have been proposed, including spectral-based and spatial-based approaches. Graph Convolutional Networks (GCN) later bridge the gap between spectral-based approaches and spatial-based [23], and then spatial-based approaches become more popular as they are efficient, flexible and general. For example, GraphSAGE with its proposed sampling and aggregation strategies [18] can be used for inductive learning and large-scale graph learning. Graph Attention Networks (GAT) adopt the self-attention mechanism to differentiate the importance of neighbors [49]. Graph Isomorphism Networks (GIN) introduce a pooling architecture so that they have expressive power as Weisfeiler-Lehman test [55].

2.2. Explainability of GNN

Most of the GNN explanation approaches can be categorized into four types: perturbation-based, gradient-based, decomposition, and surrogate approaches [58].

Our proposed approach is closely aligned with perturbation-based methods, which study the outcome changes w.r.t. different input perturbations. GNNExplainer [57] employs a trainable mask to perturb the data in the input space, to maximize the mutual information between perturbed input data and model outcome, to obtain a subgraph explanation that is relevant

contrast, the approaches [57, 32] searching for a sufficient explanation seek to locate a subset of the graph that can sufficiently cause the outcome by maximizing the mutual information between the input and outcome. For such approaches, the sufficient explanation might not be concise enough for people to understand, e.g., in the bottom-left of Figure 1, the explanation covers almost the whole graph including both sports and snack lovers and their relationships. Furthermore, a recent approach [47] considers a trade-off between necessity and sufficiency, whereas the trade-off is heuristically determined by hyper-parameters so that the explanation obtained might not be the *most* necessary and sufficient.

Different from either sufficient or necessary explanations, a both necessary and sufficient explanation offers completeness without sacrificing conciseness. As illustrated in the bottom-right of Figure 1, the most necessary and sufficient explanation includes all sport lovers, ensuring completeness, while excluding the redundant snack lovers, ensuring conciseness. The necessity and sufficiency of the explanation deserve a privileged position in the theory and practice of explainable AI [52, 37], and we argue that *an ideal explanation should be most necessary and sufficient.* A formal way to quantify the necessity and sufficiency of an explanation is through the use of the *Probability of Necessity and Sufficiency* (PNS) [38]. However, there are two main challenges in identifying PNS: 1) the violation of the assumption of monotonicity making PNS not identifiable [37, 48] and 2) the difficulty of counterfactual estimation. On one hand, to address the above identifiability issue of PNS, we derive a lower bound of PNS that can be estimated via counterfactual estimation. On the other hand, for counterfactual estimation, we depict the GNN as a structural causal model (SCM) and estimate the counterfactual probability by intervention under the SCM. In ad-

for the particular prediction. Though PGExlainer [32] shares the same objective with GNNExplainer, it achieved faster inference time by learning a parameterized mapping from the graph representation space to subgraph space. CF-GNNExplainer generates a counterfactual explanation that can flip the model prediction subject to a minimal perturbation [29]. RG-Explainer [43] leverages Reinforcement Learning (RL) algorithm to generate the explanation by sequentially adding nodes (action) based on the current generated explanation (state), which has the similar objective (reward) with [32]. RC-Explainer [51] also employs a RL algorithm to search for the explanation that maximizes the causal effect obtained by the edge perturbation. CF^2 [47], arguably the most closely related work to our approach, utilizes both factual and counterfactual reasoning to generate heuristic necessary and sufficient explanations. The main difference between CF^2 and ours is that the former searches for a trade-off explanation between necessity and sufficiency, while our work searches for the most necessary and sufficiency explanation. Besides, CF^2 only samples one counterfactual (necessary) for computing the necessary strength, which is a degradation to ours.

Gradient-based approaches approximate the importance of an input using the gradients of its outcome obtained by back-propagation. The saliency map approach, which is used to indicate the input importance, is obtained by computing the squared norm of the gradients [6]. In another method called Guided Backpropagation (GBP), the negative gradients are clipped during back-propagation as negative gradients are challenging to explain [6].

Decomposition approaches aim to decompose the model outcome into several terms as the importance scores of the corresponding input feature. Layer-wise Relevance Propagation (LRP) decomposes the GNN output into node importance scores, whereas the edge importance scores cannot be provided [6]. EBP [40] shares a similar idea with LRP, while it is based on the law of total probability.

Surrogate approaches leverage a simple and interpretable model to approximate the behavior of a complex model locally. GraphLIME [20] extends the LIME algorithm [41], and employs a Hilbert-Schmidt Independence Criterion Lasso as a surrogate to approximate the GNN instance. In particular, the weights of the surrogate model indicate the importance scores of the nodes. In PGM-Explainer [50], a probabilistic graphical model is utilized as surrogate for explaining the GNN instance.

2.3. Casual Explainability

Causal inference has a long history in statistics [38] and there is now an increasing interest in solving crucial problems of machine learning that benefit from causality [42, 56, 62, 7], including explainability. The well-known perturbation-based approaches such as LIME [41], Shapley values [31] implicitly use causal inference to estimate the attribution scores, which can be viewed as a special case of causal effect. Besides, Chattopadhyay et al. [11] views the

neural network architecture as a Structural Causal Model (SCM) and estimates the average causal effect upon it. Also, there are a few works [52, 16, 61, 8] utilizing the aspects of necessity and sufficiency for explainability via causal interpretations, which are the most related works to ours.

3. Preliminary

3.1. Causality

Here we introduce the basic causality preliminaries to enhance the understanding of this work. The identification of PNS, PN, and PS requires the counterfactual estimation. In the literature of [38], counterfactual is obtained by intervention under the structural causal model (SCM), which is defined in Definition 1.

Definition 1. *(Structural Causal Model). A structural causal model is a triple:*

$$M = (\mathbf{V}, \mathbf{U}, F),$$

where

1. $\mathbf{V}$ *is a set of variables called* endogenous, *that are determined by variables in the model, i.e.,* $\mathbf{U} \cup \mathbf{V}$.
2. $\mathbf{U}$ *is a set of variables called* exogenous, *that are determined by factors outside the model.*
3. F *is a set of functions where each* f_i *is a mapping from* $(\mathbf{V} \setminus \mathbf{v}_i) \times \mathbf{U}$ *to* $\mathbf{v}_i$, *i.e.,*

$$\mathbf{v}_i = f_i(Pa(\mathbf{v}_i), \mathbf{u}_i),$$

where $Pa(\mathbf{v}_i)$ *are the parent variables of* $\mathbf{v}_i$, *and* $\mathbf{u}_i$ *is the exogenous of* $\mathbf{v}_i$.

The SCM can be associated with a directed acyclic graph (DAG), where each node corresponds to variables in V and the directed edges point from members of $Pa(\mathbf{v}_i)$ toward $\mathbf{v}_i$. A SCM can properly model the data-generating process through functional mechanisms, that is, an endogenous $\mathbf{V}_i$ is determined by its parents $Pa(\mathbf{v}_i)$ and exogenous $\mathbf{u}_i$ via the function f_i, denoted as $\mathbf{v}_i = f_i(Pa(\mathbf{v}_i), \mathbf{u}_i)$. The functional characterization in SCM provides a convenient language for specifying how the resulting distribution would change in response to interventions. Regarding intervention, the simplest intervention such as an intervention of $do(\mathbf{v}_i = v'_i)$, amounts to removing the old generating mechanism $\mathbf{v}_i = f_i(Pa(\mathbf{v}_i), \mathbf{u}_i)$ from the SCM and substituting $\mathbf{v}_i = v'_i$ in the remaining generating equations. The concept of counterfactual refers to the consequences of the interventions, given certain facts. In particular, the given certain facts provide evidence about the actual state of the world, which is exogenous in the literature of [38]. The counterfactual is obtained by intervening on some variables under the SCM while keeping the actual state (exogenous) the same.

3.2. Graph Neural Networks

Graph neural networks are capable of incorporating both graph structure and node features into representations in an end-to-end fashion, to facilitate downstream tasks such as node classification task and graph classification task. In particular, in the k-th layer of GNNs, the learning process of the representation of node v can be divided into the following three steps:

- First, obtaining message $\mathbf{m}_{v,u}^{(k)}$ for any node pair (v, u) through a message function MSG:

$$\mathbf{m}_{v,u}^{(k)} = MSG(\mathbf{h}_v^{(k-1)}, \mathbf{h}_u^{(k-1)}, \mathbf{e}_{v,u}),$$

where $\mathbf{h}_v^{(k-1)}$ and $\mathbf{h}_u^{(k-1)}$ denote the representations of nodes v and u in the $(k-1)$-the layer, and $\mathbf{e}_{v,u}$ denotes the entry (relation) between nodes v and u.

- Second, aggregating messages from node v's neighbors $\mathcal{N}_v$ and calculating the an aggregated message $\mathbf{M}_v^{(k)}$ via a aggregating function AGG:

$$\mathbf{M}_v^{(k)} = AGG(\mathbf{m}_{v,u}^{(k)} | u \in \mathcal{N}_v).$$

- Third, updating node v's representation $\mathbf{h}_v^{(k)}$ using the aggregated messages $\mathbf{M}_v^{(k)}$ and node v's representation in the previous layer $\mathbf{h}_v^{(k-1)}$ via a update function $UPDATE$:

$$\mathbf{M}_v^{(k)} = UPDATE(\mathbf{M}_v^{(k)}, \mathbf{h}_v^{(k-1)}).$$

After obtaining the representation of each node, a node-level read-out and a graph-level read-out can be applied for node classification task and graph classification task respectively.

3.3. Probability of Necessity and Sufficiency

As we discussed before, a convincing explanation should be necessary and sufficient. To quantify the degree of necessity and sufficiency of an explanation to the model outcome, the Probability of Necessity and Sufficiency (PNS) is formally defined as follows.

Definition 2. (Probability of necessity and sufficiency [38]).

$$PNS(\xi) = P(\mathbf{Y}_{\xi^c} \neq \hat{y}, \mathbf{Y}_\xi = \hat{y}), \qquad (1)$$

where $\mathbf{Y}_{\xi^c}$ and $\mathbf{Y}_\xi$ are the potential outcome variables under the treatments ξ^c and ξ respectively, and ξ^c is the complementary event of ξ.

PNS measures the necessity and sufficiency of treatment ξ to model outcome $\hat{y}$ in probability space. Intuitively, PNS indicates the probability that the outcome $\hat{y}$ responds to both treatments ξ and ξ^c. However, direct optimization of the objective is intractable, given the non-identifiability of PNS shown in Eq. (1) due to the potential violation of monotonicity as defined in Definition 3, as well as the challenges of counterfactual estimation [38].

Definition 3. (Monotonicity). The model outcome $\mathbf{Y}$ is monotonic relative to the explanation event ξ if and only if:

$$(\mathbf{Y}_\xi \neq \hat{y}) \wedge (\mathbf{Y}_{\xi^c} = \hat{y}) = false.$$

Monotonicity indicates that a change from ξ^c to ξ cannot assure the outcome also changes from $\mathbf{Y} = \hat{y}$ to $\mathbf{Y} \neq \hat{y}$ [37]. However, the assumption of monotonicity will not always hold during the explanation searching stage. Instead of addressing the non-monotonic issue to identify the exact PNS, it is reasonable to maximize a lower bound of PNS(ξ) as shown in Lemma 1 for our objective optimization.

Lemma 1. The lower bound of PNS(ξ) is:

$$\max\{0, P(\mathbf{Y}_{\xi^c} \neq \hat{y}) + P(\mathbf{Y}_\xi = \hat{y}) - 1\}. \qquad (2)$$

In particular, the lower bound is tight if the assumption of monotonicity holds, as shown in Lemma 2. The proof of Lemma 2 is given in Appendix A.2.

Lemma 2. When the outcome $\mathbf{Y}$ is monotonic relative to explanation event ξ, the lower bound in Eq. (2) equals to the exact PNS(ξ).

4. Methodology

In this section, we develop our approach NSEG to generate the most **N**ecessary and **S**ufficient **E**xplanations for **G**NNs by optimizing our objective, the Probability of Necessity and Sufficiency (PNS). We maximize a derived lower bound of PNS since the non-identifiability of PNS and incorporate a proposed SCM of GNN for the identification of the lower bound. Additionally, we introduce a continuous optimization method that utilizes continuous masks and a sampling strategy. This approach enables tractable optimization of large-scale graphs, ensuring the scalability and efficiency of our approach.

4.1. Problem Definition

Given a trained GNN model f_θ parameterized by θ for graph classification task, our task is to explain a specific instance $\mathcal{I} : \hat{y} = f_\theta(E_i, X_i)$ produced by the model f_θ by generating a necessary and sufficient explanation in a post-hoc manner, where $\hat{y}$ is the predicted label, $X_i = \{x_v | v \in V_i\}$, and E_i, V_i are edges and vertices of graph G_i. In our framework, we consider the explanation of a specific instance $\mathcal{I}$ as an event, i.e., $(\mathbf{E} = E_i', \mathbf{X} = X_i')$, that most necessarily and sufficiently causes the model outcome $\mathbf{Y} = \hat{y}$ (with maximum PNS), where $E_i' \subset E_i$ and $X_i' = \{x_v | v \in V_i'\}$ with $V_i' \subset V_i$. Note that our main focus is to identify a set of node features instead of a set of features in the feature dimensions, where the latter has been well investigated in [11]. Similarly, as for the formulation of node classification task, our task is to explain a node-level instance $\mathcal{I} : \hat{y}_i = f_\theta^{(i)}(E, X)$ for node i by generating a necessary and sufficient explanation $\mathbf{E} = E', \mathbf{X} = X'$, where $\hat{y}_i$ is the predicted label of node i. Without loss of generality, we

formulate our approach in a graph classification fashion in the following paper. Regarding our notations, we use the bold font notation for random variable to emphasize the distinction between r.v. and its realization.

4.2. Lower Bound of the Explanation's PNS on GNN

Although PNS has been formally defined, applying it to GNNs is not a trivial task due to: 1) the combined influence of both structural information (edges) and feature information (node features) on GNN predictions, 2) GNN takes continuous inputs rather than a binary variable ξ. To overcome these challenges, we extend PNS to the graph domain by defining a joint event $(\mathbf{E} = E_i', \mathbf{X} = X_i')$ as the explanation, and its complement event $(\mathbf{E} = E_i', \mathbf{X} = X_i')^c$ which can be derived as $(\mathbf{E} \neq E_i', \mathbf{X} \neq X_i') \vee (\mathbf{E} \neq E_i', \mathbf{X} = X_i') \vee (\mathbf{E} = E_i', \mathbf{X} \neq X_i')$. Hence, our objective becomes maximizing the probability of necessity and sufficiency w.r.t. E_i' and X_i' as:

$$\max_{E_i', X_i'} \text{PNS}^{e,f}(E_i', X_i'), \tag{3}$$

where $\text{PNS}^{e,f}(E_i', X_i')$ is defined as:

$$\begin{aligned} &\text{PNS}^{e,f}(E_i', X_i') \\ &= P_\theta(\mathbf{Y}_{(\mathbf{E}=E_i', \mathbf{X}=X_i')^c} \neq \hat{y}, \mathbf{Y}_{\mathbf{E}=E_i', \mathbf{X}=X_i'} = \hat{y}), \end{aligned} \tag{4}$$

where θ denotes the model parameter. The lower bound of $\text{PNS}^{e,f}(E_i', X_i')$ is:

$$\begin{aligned} \max\{0, &P_\theta(\mathbf{Y}_{(\mathbf{E}=E_i', \mathbf{X}=X_i')^c} \neq \hat{y}) \\ &+ P_\theta(\mathbf{Y}_{\mathbf{E}=E_i', \mathbf{X}=X_i'} = \hat{y}) - 1\}. \end{aligned} \tag{5}$$

Note that our approach to optimize the above lower bound in Eq. (5) provides a joint explanation of both the edge and node feature. Single edge explanation and single node feature explanation are special cases of our joint explanation. Specifically, the formulations of single-edge and single-node features are familiar with the joint formulation, which are given by:

$$\text{PNS}^e(E_i', X_i) = P_\theta(\mathbf{Y}_{\mathbf{E} \neq E_i'} \neq \hat{y}, \mathbf{Y}_{\mathbf{E}=E_i'} = \hat{y}|\mathbf{X} = X_i),$$
$$\text{PNS}^f(E_i, X_i') = P_\theta(\mathbf{Y}_{\mathbf{X} \neq X_i'} \neq \hat{y}, \mathbf{Y}_{\mathbf{X}=X_i'} = \hat{y}|\mathbf{E} = E_i),$$

where the lower bound of each is shown as follows respectively.

$$\max\{0, P_\theta(\mathbf{Y}_{\mathbf{E}=E_i'} \neq \hat{y}|X_i) + P_\theta(\mathbf{Y}_{\mathbf{E}=E_i'} = \hat{y}|X_i) - 1\},$$
$$\max\{0, P_\theta(\mathbf{Y}_{\mathbf{X}=X_i'} \neq \hat{y}|E_i) + P_\theta(\mathbf{Y}_{\mathbf{X}=X_i'} = \hat{y}|E_i) - 1\}.$$

4.3. Estimating the Lower Bound via Counterfactual Estimation

4.3.1. GNN as Structural Causal Model

Obtaining the probability term in Eq. (5) requires generating counterfactual by intervention under a specific structural causal model (SCM) [38]. In terms of counterfactual, it answers what would the outcome $\mathbf{Y}$ be if $(\mathbf{E} = E_i', \mathbf{X} = X_i')^c$ or $(\mathbf{E} = E_i', \mathbf{X} = X_i')$. Building upon the interpretation of feed-forward neural networks as SCM

for investigating causal attribution [2, 11], we extend the SCM interpretation to GNNs. In particular, GNNs can be interpreted as directed acyclic graphs with directed edges from the lower layer to the layer above.

Proposition 1. *An $(l + 1)$-layer GNN corresponds to an SCM $M(\{\mathbf{X}, \mathbf{E}, \mathbf{H}^{(1)}, ..., \mathbf{H}^{(l)}, \mathbf{Y}\}, \mathbf{U}, \{f^{(0)}, f_\theta^{(1)}, ..., f_\theta^{(l+1)}\})$, where $\mathbf{H}^{(k)}$ denotes a set of node hidden representation after the k-th graph convolution layer, i.e., $\mathbf{H}^{(k)} = f_\theta^{(k)}(\mathbf{E}, \mathbf{H}^{(k-1)})$. $f_\theta^{(l+1)}$ denotes the read-out layer, which can be a node-level read-out for node classification tasks or a graph-level read-out for graph classification tasks, with $\mathbf{Y} = f_\theta^{(l+1)}(\mathbf{H}^{(l)})$. $\mathbf{U}$ refers to a set of exogenous which act as causal factors for $\mathbf{X}$ and $\mathbf{E}$, i.e., $\mathbf{E}, \mathbf{X} = f^{(0)}(\mathbf{U})$.*

The proof of Proposition 1 is provided in Appendix A.3. For a more intuitive understanding, the left side of Figure 2 illustrates M's corresponding causal graph. In the k-th layer of GNN, the node v's representation $\mathbf{h}_v^{(k)}$ is determined by all nodes' representations $\mathbf{H}^{(k-1)}$ in $(k-1)$-th layer and the entry of the graph $\mathbf{E}$, specifically, aggregating node u's representation $\mathbf{h}_u^{(k-1)}$ to obtain $\mathbf{h}_v^{(k)}$ if the entry $\mathbf{e}_{v,u}$ is not zero. Since our focus is on the mapping from the input to output rather than the full intrinsic mappings in the hidden layer, the SCM of GNN can be reduced to SCM $M(\{\mathbf{E}, \mathbf{X}, \mathbf{Y}\}, \mathbf{U}, \{f^{(0)}, f_\theta\})$ by marginalizing the hidden representations.

Proposition 2. *The SCM of an $(l + 1)$-layer GNN, $M(\{\mathbf{X}, \mathbf{E}, \mathbf{H}^{(1)}, ..., \mathbf{H}^{(l)}, \mathbf{Y}\}, \mathbf{U}, \{f^{(0)}, f_\theta^{(1)}, ..., f_\theta^{(l+1)}\})$ can be reduced to $M(\{\mathbf{E}, \mathbf{X}, \mathbf{Y}\}, \mathbf{U}, \{f^{(0)}, f_\theta\})$.*

The proof of Proposition 2 is presented in Appendix A.4. Intuitively, marginalizing the hidden representations is analogous to deleting edges connecting the hidden representations and creating new directed edges from the parents of the deleted to their respective child vertices in the causal diagram depicted on the left side of Figure 2. The corresponding causal diagram of the reduced SCM in Proposition 2 is shown on the right side of Figure 2.

4.3.2. Counterfactual Estimation

Given the reduced SCM depicted in Proposition 2, we can obtain the interventional probability via *do*-calculus, by controlling $\mathbf{E}$ and $\mathbf{X}$ for edges and node features respectively. To facilitate understanding, we present the formulations separately for edges and node features before combining them to give the joint formulation.

Counterfactual for Edges. Regarding edges, the interventional probability of the counterfactual $(\mathbf{E} = E_i')$ can be obtained by intervening on $\mathbf{E}$, i.e., replacing the causal mechanism from exogenous $\mathbf{U}$ to the edge $\mathbf{E}$ with the intervention $do(\mathbf{E} = E_i')$ while keeping other mechanisms
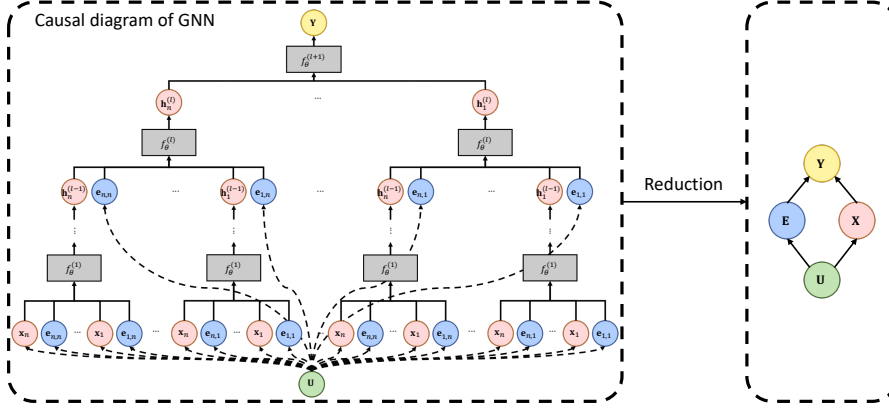
Figure 2: The causal diagram of corresponding SCM of GNN. $\mathbf{h}_v^k$ denotes the hidden representation of node v in k-th layer, $\mathbf{e}_{v,u}$ denotes the entry between v and u, $\mathbf{x}_v$ denotes the node feature of node v, $\mathbf{H}^{(k)} = \{\mathbf{h}_v^{(k)} | v \in V\}$ denotes a set of node representations in k-th layer, $\mathbf{X} = \{\mathbf{x}_v | v \in V\}$ denotes a set of node features, $\mathbf{E} = \{\mathbf{e}_{v,u} | v, u \in V\}$ denotes a set of graph entries.

unperturbed, i.e., $\mathbf{X} = X_i$:

$$
\begin{aligned}
& P_\theta(\mathbf{Y}_{\mathbf{E}=E_i'} = \hat{y} | \mathbf{X} = X_i) \\
=& P_\theta(\mathbf{Y} = \hat{y} | do(\mathbf{E} = E_i'), \mathbf{X} = X_i) \\
=& f_\theta^{\hat{y}}(E_i', X_i),
\end{aligned}
\tag{6}
$$

where $f_\theta^{\hat{y}}(.,.)$ outputs the probability of the class $\hat{y}$ and $\hat{y}$ is the label predicted by the model. Note that we only alter the generating mechanism $f^{(0)}$ for $\mathbf{E}$ and the GNN model f_θ remains unchanged. As for the interventional probability of the counterfactual $(\mathbf{E} \neq E_i')$,

$$
\begin{aligned}
& P_\theta(\mathbf{Y}_{\mathbf{E}\neq E_i'} \neq \hat{y} | \mathbf{X} = X_i) \\
=& P_\theta(\mathbf{Y} \neq \hat{y} | do(\mathbf{E} \neq E_i'), \mathbf{X} = X_i) \\
=& P_\theta(\mathbf{Y} \neq \hat{y} | \mathbf{E} \in \mathcal{E}_i \setminus E_i', \mathbf{X} = X_i) \\
=& 1 - \mathbb{E}_{\overline{E}_i'}[f_\theta^{\hat{y}}(\overline{E}_i', X_i)],
\end{aligned}
\tag{7}
$$

where $\overline{E}_i' \sim p(\overline{E}_i' | \mathcal{E}_i \setminus E_i')$ which can be specified by $p(\mathbf{U})$ according to the corresponding SCM, and $\mathcal{E}_i$ is the sub-edge space of graph G_i. In the absence of any prior knowledge of $p(\overline{E}_i' | \mathcal{E}_i \setminus E_i')$, it is reasonable to assume $p(\overline{E}_i' | \mathcal{E}_i \setminus E_i')$ is uniformly distributed over the sub-edge space $\mathcal{E}$, which encourage the exploration of all $(\mathbf{E} \neq E_i')$. Prior knowledge $p(\mathbf{U})$ can also be incorporated to refine the estimation of counterfactual.

Counterfactual for Node Features. Our goal is to identify a subset of node features that cause the prediction, that is, features in a subset of nodes V_i', instead of identifying a subset of feature in feature dimensions. One way to incorporate sub-node structure with the node features is to set the features outside V_i' to $\mathbf{0}$ while keeping features inside V_i' the same, since the zero-valued features have no impact in feed-forward process. Thus, the interventional probability of counterfactual $(\mathbf{X} = X_i')$ can be obtained by intervening on $\mathbf{X}$, i.e., replacing the causal mechanism from $\mathbf{U}$ to $\mathbf{X}$ with the intervention $do(\mathbf{X} = X_i')$ while keep-

ing other mechanisms unperturbed, which is given by,

$$
\begin{aligned}
& P_\theta(\mathbf{Y}_{\mathbf{X}=X_i'} = \hat{y} | \mathbf{E} = E_i) \\
=& P_\theta(\mathbf{Y} = \hat{y} | do(\mathbf{X} = X_i'), \mathbf{E} = E_i) \\
=& f_\theta^{\hat{y}}(E_i, X_i') \\
& with \quad X_i' = \{x_v | v \in V_i'\} \cup \{\mathbf{0} | v \notin V_i'\}.
\end{aligned}
\tag{8}
$$

Likewise, the interventional probability of counterfactual $(\mathbf{X} = X_i')$ is given by

$$
\begin{aligned}
& P_\theta(\mathbf{Y}_{\mathbf{X}\neq X_i'} \neq \hat{y} | \mathbf{E} = E_i) \\
=& P_\theta(\mathbf{Y} \neq \hat{y} | do(\mathbf{X} \neq X_i'), \mathbf{E} = E_i) \\
=& \mathbb{E}_{\overline{X}_i'}[P_\theta(\mathbf{Y} \neq \hat{y} | \mathbf{X} = \overline{X}_i', \mathbf{E} = E_i)] \\
=& 1 - \mathbb{E}_{\overline{X}_i'}[f_\theta^{\hat{y}}(E_i, \overline{X}_i')] \\
& with \quad \overline{X}_i' = \{x_v | v \in \overline{V}_i'\} \cup \{\mathbf{0} | v \notin \overline{V}_i'\},
\end{aligned}
\tag{9}
$$

where $\overline{X}_i' \sim p(\overline{X}_i' | \overline{V}_i') p(\overline{V}_i' | \mathcal{V}_i \setminus V_i') = p(\overline{V}_i' | \mathcal{V}_i \setminus V_i')$, and $\mathcal{V}_i$ is sub-node space of graph G_i.

Joint Formulation of Edge and Node Feature. The joint formulation of both edge and node feature explanation in Eq. (5) requires us to generate counterfactual for edge and node feature by intervention on both $\mathbf{E}$ and $\mathbf{X}$. Notably, the event $(\mathbf{E} = E_i', \mathbf{X} = X_i')^c$ in Eq. (5) can be divided into three sub-events, $(\mathbf{E} \neq E_i', \mathbf{X} \neq X_i')$, $(\mathbf{E} \neq E_i', \mathbf{X} = X_i')$, and $(\mathbf{E} = E_i', \mathbf{X} \neq X_i')$. According to the interventional probability in Eqs. (6), (7), (8), and (9), the lower bound of PNS in Eq. (5) can be derived as follows.

$$
\begin{aligned}
\max\{ & 0, -P_{00}\mathbb{E}_{\overline{E}_i', \overline{X}_i'}[f_\theta^{\hat{y}}(\overline{E}_i', \overline{X}_i')] \\
& -P_{01}\mathbb{E}_{\overline{E}_i'}[f_\theta^{\hat{y}}(\overline{E}_i', X_i')] \\
& -P_{10}\mathbb{E}_{\overline{X}_i'}[f_\theta^{\hat{y}}(E_i', \overline{X}_i')] \\
& +f_\theta^{\hat{y}}(E_i', X_i') \},
\end{aligned}
\tag{10}
$$

with:

$$
\begin{aligned}
P_{00} &= P(\mathbf{E} \neq E_i', \mathbf{X} \neq X_i' | (\mathbf{E} = E_i', \mathbf{X} = X_i')^c), \\
P_{01} &= P(\mathbf{E} \neq E_i', \mathbf{X} = X_i' | (\mathbf{E} = E_i', \mathbf{X} = X_i')^c), \\
P_{10} &= P(\mathbf{E} = E_i', \mathbf{X} \neq X_i' | (\mathbf{E} = E_i', \mathbf{X} = X_i')^c),
\end{aligned}
\tag{11}
$$

Note that the values of P_{00}, P_{01} and P_{10} need to be specified manually. Since these values are interpretable, their ranges can be flexibly defined based on prior knowledge. In this work, we simply assume all possible events derived from $(E_i, X_i)^c$ are equally likely, setting $P_{00}=P_{01}=P_{10}=1/3$.

4.4. Generating the Explanation via Lower Bound Optimization

Continuous mask. Enumerating all possible E_i' and X_i' for objective optimization in large-scale graphs is infeasible. To enhance the scalability of our approach, we adopt a continuous relaxation approach, as in [57], using continuous masks that allow for optimization through gradient descent. In particular, we design two masks $M_e \in [0,1]^m$ and $M_f \in [0,1]^n$ to mask the edges E_i and the node feature X_i to obtain E_i' and X_i' respectively, where m is the number of edges, and n is the number of nodes. Intuitively, $M_e^k = 0$ indicates that deleting the k-th edge from the full edges E_i, while $M_e^k = 1$ indicates retaining the k-th edge.

$$E_i' = M_e \odot E_i,$$
$$X_i' = M_f \odot X_i + (1 - M_f) \odot \mathbf{0} = M_f \odot X_i,$$

where $\odot$ denotes the Hadamard multiplication, and $\mathbf{0} \in 0^{n \times d}$ denotes a zero matrix. After masking, the term $f_\theta^{\hat{y}}(E_i', X_i')$ in Eq. (10) can be derived as follows.

$$f_\theta^{\hat{y}}(E_i', X_i') = f_\theta^{\hat{y}}(M_e \odot E_i, M_f \odot X_i), \quad (12)$$

Sampling strategy. Incorporating the masks to generate samples from $p(\overline{E}_i' | \mathcal{E}_i \setminus E_i')$ and $p(\overline{V}_i' | \mathcal{V}_i \setminus V_i')$ in Eq. (10), a heuristic sampling strategy is proposed such that the Monte Carlo estimation of the expectations in Eq. (10) are differentiable w.r.t. M_e and M_f. Inspired by the reparameterization trick proposed in [22], an auxiliary variable ϵ is used in our sampling strategy for the sample generation.

Specifically, the generating process of edge sample $\overline{E}_i'$ from $p(\overline{E}_i' | \mathcal{E}_i \setminus E_i')$ can be expressed as a deterministic function with auxiliary variable ϵ_e, which is:

$$\overline{E}_i' = (1 - M_e + \epsilon_e) \odot E_i, \quad \epsilon_e \sim p(\epsilon_e). \quad (13)$$

Intuitively, the term $1 - M_e$ aims to satisfy the given condition of $\mathbf{E} \in \mathcal{E}_i \setminus E_i'$, and a positive value of ϵ_e^k increases the weight of the existence of the k-th edge, while a negative value decreases the weight. Without any prior knowledge of $p(\overline{E}_i' | \mathcal{E}_i \setminus E_i')$, it is reasonable to assume ϵ_e is uniformly distributed to encourage the fair exploration. Similarly, the generating process of node feature sample $\overline{X}_i'$ from $p(\overline{V}_i' | \mathcal{V}_i \setminus V_i')$ can be expressed as follows.

$$\overline{X}_i' = (1 - M_f + \epsilon_f) \odot X_i, \quad \epsilon_f \sim p(\epsilon_f). \quad (14)$$

Thus, combining with Eqs. (13) and (14) the term $\mathbb{E}_{\overline{E}_i', \overline{X}_i'}[f_\theta^{\hat{y}}(\overline{E}_i', \overline{X}_i')]$ in Eq. (10) can be derived as:

$$\mathbb{E}_{\epsilon_e, \epsilon_f}[f_\theta^{\hat{y}}((1 - M_e + \epsilon_e) \odot E_i, (1 - M_f + \epsilon_f) \odot X_i)], \quad (15)$$
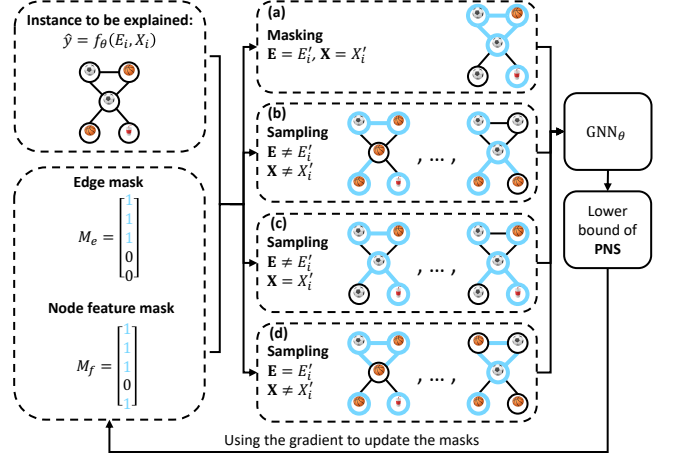

Figure 3: Illustration of the overall framework of NSEG. (a) is the process to obtain Eq. (12); (b), (c), and (d) are the processes to obtain Eqs. (15), (16), and (17), respectively.

similarly, the term $\mathbb{E}_{\overline{E}_i'}[f_\theta^{\hat{y}}(\overline{E}_i', X_i')]$ is given by:

$$\mathbb{E}_{\epsilon_e}[f_\theta^{\hat{y}}((1 - M_e + \epsilon_e) \odot E_i, M_f \odot X_i)], \quad (16)$$

also, the term $\mathbb{E}_{\overline{X}_i'}[f_\theta^{\hat{y}}(E_i', \overline{X}_i')]$ is given by:

$$\mathbb{E}_{\epsilon_f}[f_\theta^{\hat{y}}(M_e \odot E_i, (1 - M_f + \epsilon_f) \odot X_i)]. \quad (17)$$

Overall, the final optimizable lower bound of PNS combined with continuous masks, $\text{PNS}_{lb}^{e,f}$, is given as follows.

$$\max\{0, -P_{00}\mathbb{E}_{\epsilon_e, \epsilon_f}[f_\theta^{\hat{y}}((1 - M_e + \epsilon_e) \odot E_i, (1 - M_f + \epsilon_f) \odot X_i)]$$
$$-P_{01}\mathbb{E}_{\epsilon_e}[f_\theta^{\hat{y}}((1 - M_e + \epsilon_e) \odot E_i, M_f \odot X_i)]$$
$$-P_{10}\mathbb{E}_{\epsilon_f}[f_\theta^{\hat{y}}(M_e \odot E_i, (1 - M_f + \epsilon_f) \odot X_i)]$$
$$+f_\theta^{\hat{y}}(M_e \odot E_i, M_f \odot X_i)\}, \quad (18)$$

Despite joint explanation, for single edge explanation, the final objective is $\max\{0, \mathbb{E}_{\epsilon_e}[-f_\theta^{\hat{y}}((1 - M_e + \epsilon_e) \odot E_i, X_i)] + f_\theta^{\hat{y}}(M_e \odot E_i, X_i)\}$. Similarly for single node feature explanation, the final objective is $\max\{0, \mathbb{E}_{\epsilon_f}[f_\theta^{\hat{y}}(E_i, (1 - M_f + \epsilon_f) \odot X_i)] + f_\theta^{\hat{y}}(E_i, M_f \odot X_i)\}$.

4.5. Model Summary

The overall framework of NSEG is depicted in Figure 3. To explain an instance $\mathcal{I} : \hat{y} = f_\theta(E_i, X_i)$, an edge mask M_e and a node feature mask M_f are used to obtain and maximize the lower bound of PNS in Eq. (18). Then the gradients of the overall loss w.r.t. to the masks are employed to update the masks and obtain the necessary and sufficient explanation. Specifically, we utilize a *mask size* regularization term to enhance the optimization, which is also adopted in prior works [57, 32]. Intuitively, the regularization operates as an L_1 penalization, compelling the explanation to concentrate on the most important portions of the input. Besides, a *mask entropy* regularization term is added to discretize the mask, i.e., the values of

the mask are concentrated around a few scalars when the *mask entropy* is low [3]. Formally, the overall loss $\mathcal{L}$ is:

$$\mathcal{L} = -\text{PNS}_{lb} + \alpha_e ||M_e||_1 + \beta_e Ent(M_e) + \alpha_f ||M_f||_1 + \beta_f Ent(M_f),$$ (19)

where PNS_{lb} is the lower bound of PNS shown in Eq. (18), and $Ent()$ is the element-wise entropy to encourage the discretization of the mask.

Formally, the NSEG algorithm is outlined in Algorithm 1. After obtaining the final masks M_e and M_f, the explanation (E_i', X_i') can be obtained via *extract_explanation* in Algorithm 1. The possible choices for *extract_explanation* can be, extracting the top-K explanation based on the weights of the mask [57], or employing a pre-defined threshold [47].

Algorithm 1 The NSEG algorithm

Input: The trained GNN model f_θ, the instance to be explained $\mathcal{I} : \hat{y} = f_\theta(E_i, X_i)$, the hyper-parameters $(\alpha_e, \beta_e, \alpha_f, \beta_f)$, lr γ, and the # epochs n.
Output: The explanation (E_i', X_i').
1: Randomly initialize the M_e and M_f
2: **for** $j = 1, ..., n$ **do**
3:   $\text{PNS}_{lb}(M_e^{(j)}, M_f^{(j)}) \leftarrow$ the lower bound of PNS calculated by Eq.(18)
4:   $\mathcal{L}(M_e^{(j)}, M_f^{(j)}) \leftarrow$ the overall loss calculated by Eq. (19)
5:   $(M_e^{(j)}, M_f^{(j)}) \leftarrow$ Update the masks via $\nabla_{M_e, M_f} \mathcal{L}(M_e^{(j-1)}, M_f^{(j-1)})$
6: **end for**
7: $(M_e, M_f) \leftarrow (M_e^{(n)}, M_f^{(n)})$
8: $(E_i', X_i') \leftarrow extract\_explanation(M_e, M_f)$

---

5. NSEG Variants: Optimizing Only PN or PS

In this section, we present two variants of NSEG(PNS): NSEG(PN) and NSEG(PS). NSEG(PN) focuses on optimizing the Probability of Necessity (PN), considering only the necessity of the explanation. On the other hand, NSEG(PS) concentrates on optimizing the Probability of Sufficiency (PS), considering only the sufficiency of the explanation. These two variants are utilized in our ablation study.

5.1. PN and PS

In the NSEG variants, NSEG(PN) and NSEG(PS), our goal is to generate the most necessary explanation via NSEG(PN) and the most sufficient explanation via NSEG(PS). Theoretically, the probability of necessity, PN, and the probability of sufficiency, PS, which can quantify the necessity and sufficiency of explanation (ξ) to the model outcome $\hat{y}$ respectively [37], are defined in Definition 4 and 5.

Definition 4. *(Probability of necessity).*

$$PN(\xi) = P(\mathbf{Y}_{\xi^c} \neq \hat{y}|\xi, \mathbf{Y} = \hat{y}).$$ (20)

Definition 5. *(Probability of sufficiency).*

$$PS(\xi) = P(\mathbf{Y}_\xi = \hat{y}|\xi^c, \mathbf{Y} \neq \hat{y}).$$ (21)

PN captures the probability that the model outcome $\hat{y}$ changes with the absence of the event ξ, given the fact that the event ξ happens and the model outcome is $\hat{y}$. In similar, PS captures the probability that the model outcome is $\hat{y}$ with the existence of the event ξ, given the fact that the event ξ does not happen, and the model outcome is not $\hat{y}$.

Similar to the generalized formulation in Eq. (3), both PN and PS in Eqs. (20) and (21) can be generalized to GNN explanation formulations, which are given as follows.

$$\begin{aligned} &\text{PN}^{e,f}(E_i', X_i') \\ &= P_\theta(\mathbf{Y}_{(\mathbf{E}=E_i', \mathbf{X}=X_i')^c} \neq \hat{y}|\mathbf{E} = E_i', \mathbf{X} = X_i', \mathbf{Y} = \hat{y}), \end{aligned}$$ (22)

$$\begin{aligned} &\text{PS}^{e,f}(E_i', X_i') \\ &= P_\theta(\mathbf{Y}_{\mathbf{E}=E_i', \mathbf{X}=X_i'} = \hat{y}|(\mathbf{E} = E_i', \mathbf{X} = X_i')^c, \mathbf{Y} \neq \hat{y}), \end{aligned}$$ (23)

For NSEG(PN), to generate the most necessary explanation of GNN, the objective to maximize the PN defined in Eq. (22):

$$\max_{E_i', X_i'} \text{PN}^{e,f}(E_i', X_i').$$ (24)

Similar for NSEG(PS), to generate the most sufficient explanation, the objective to maximize the PS defined in Eq. (23):

$$\max_{E_i', X_i'} \text{PS}^{e,f}(E_i', X_i').$$ (25)

5.2. Optimization of PN and PS

The both PN and PS in Eqs. (22) and (23) are formulated as the probability of counterfactual in the literature of [38]. The identification of both PN and PS requires incorporating with SCM for the recovery of the exogenous [38]. However, the deterministic nature of GNN renders the given condition in the formulation of PN and PS not guaranteed, resulting in the non-identifiability of these probabilities. To better incorporate with the probability outputted by GNN, we derive lower bounds of both PN and PS that can be optimized similarly to PNS.

Proposition 3. *Given the SCM M depicted in Proposition 2, the lower bounds of $PN^{e,f}$ and $PS^{e,f}$ can be derived as:*

$$PN_{lb}^{e,f}(E_i', X_i') = \max\{0, \frac{P_\theta(\mathbf{Y} \neq \hat{y}|(\mathbf{E} = E_i', \mathbf{X} = X_i')^c) - 1}{P_\theta(\mathbf{Y} = \hat{y}|\mathbf{E} = E_i', \mathbf{X} = X_i')} + 1\},$$

$$PS_{lb}^{e,f}(E_i', X_i') = \max\{0, \frac{P_\theta(\mathbf{Y} = \hat{y}|\mathbf{E} = E_i', \mathbf{X} = X_i') - 1}{P_\theta(\mathbf{Y} \neq \hat{y}|(\mathbf{E} = E_i', \mathbf{X} = X_i')^c)} + 1\}.$$

For the proof of Proposition 3 and the optimization details of the PN and PS lower bounds, please refer to Sections Appendix A.5 and Appendix B.

8

Table 1: Dataset statistics.

	BA-Shapes	Tree-Cycles	Tree-Grid	BA2Motif	Mutagenicity	MSRC_21
#graphs	1	1	1	1000	4337	563
#avg. nodes	700	871	1231	25000	30.32	77.52
#avg. edges	2055	967	1705	51392	30.77	198.32
#classes	4	2	2	2	2	20

6. Empirical Study

In this section, we quantitatively and qualitatively evaluate our NESG(PNS) on both synthetic and real-world data to tell:

- **(RQ1)** Are the explanations necessary and sufficient?

- **(RQ1)** Are the necessary and sufficient explanations accurate?

6.1. Dataset

In this subsection, we will introduce the datasets we used in our experiments, and the statistics of the dataset are presented in Table 1.

Synthetic Datasets. We follow the graph generation process in [57, 32] and adopt four datasets, BA-Shapes, Tree-Cycles, Tree-Grid, and BA2Motif. Each dataset consists of a base graph and a set of motifs, where for node classification, the class label of each node is determined by its role in the motif, and for graph classification, the class label of each graph is determined by the type of motif in the graph. For instance, in BA-Shapes, the motifs are the "house-shaped" subgraphs, and the class labels of the nodes are "bottom", "middle", "top" and "outside". For a node instance, its ground-truth explanation is given by all edges in the motif to which it belongs. Regarding the explaining indices, we follow the settings in [50] for BA-Shapes, Tree-Cycles, and Tree-Grid, and the settings in [32] for BA2Motif.

Real-world Datasets. Two real-world datasets called Mutagenicity and MSRC_21 [36] are used for *graph classification* in our experiment. Mutagenicity contains chemical compounds that belong to two classes: either mutagenic or not. Each compound is a graph, in which each node is an atom and node features are one-hot encodings of the node atom types. We follow [32, 57, 47, 33] and treat the amino-group (-NH2) and nitro-group (-NO2) as the ground-truth explanations for the mutagenic compounds. MSRC_21 is derived from MSRC-v2 [34], a benchmark dataset in semantic image processing, where each image belongs to one of the 20 classes describing the scene of the image. A graph is constructed based on the semantic segmentation of each image, in which each node is a super-pixel whose feature is the one-hot embedding of the object semantic type. We evaluate on 250 graphs for both Mutagenicity and MSRC_21.

6.2. Experimental Setup

GNN Training Setup. For both node and graph classification tasks, we employ three layers of Graph Convolutional Networks (GCNs) [23] with ReLU activations. For node classification task, we apply node-level read-out by stacking a fully connected classification layer after the last GCN layer. For graph classification task, a sum-based read-out is used to obtain a graph representation after the last GCN layer, followed by a fully connected classification layer. The detailed training setup and results are provided in Appendix C.1.

Evaluation Metrics. We quantitatively evaluate the explanations from two aspects: (1) **the necessity and sufficiency (RQ1)**; (2) **the accuracy when the ground truth explanations are available (RQ2)**. Specifically:

- **Necessity and Sufficiency:** We utilize *Fidelity+* and *Fidelity-* (abbreviated as *Fid+* and *Fid-*) to quantify the necessity and sufficiency of the explanations, respectively [3]. The higher *Fid+*, the more necessary the explanation, on the contrary, the lower *Fid-*, the more sufficient the explanation. Additionally, we use the *charact* score, which combines both *Fid+* and *Fid-*, to measure the overall performance on both necessary and sufficient aspects [3]. The definitions of *Fid+*, *Fid-* and *charact* scores are shown as follows.

$$Fid+ = 1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\mathbf{Y}^{1-M_i} = \hat{y}_i),$$
$$Fid- = 1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\mathbf{Y}^{M_i} = \hat{y}_i),$$
$$charact = \frac{2 \times Fid+ \times (1 - Fid-)}{Fid+ + (1 - Fid-)},$$

where M is the explanation mask, and $\mathbf{Y}^{1-M} = f_\theta(E \odot (1 - M_e), X \odot (1 - M_f))$, and $\mathbf{Y}^M = f_\theta(E \odot M_e, X \odot M_f)$. We use $Fid+^c$, $Fid-^c$ and $charact^c$ to denote these scores for continuous mask explanations. Since the discrete nature of graph, we further discretize the explanations mask via threshold, and compute the $Fid+^d$, $Fid-^d$ and $charact^d$ for discrete mask explanations.

- **Accuracy:** We use *Recall@K* and *ROC-AUC* to evaluate the explanation prediction accuracy on all datasets with ground-truths: BA-Shapes, Tree-Cycles, Tree-Grid, BA2Motif, and Mutagenicity. In particular, for most methods, K corresponds to the number of edges in the ground-truth explanations, with values of 6, 6, 12, 5, 15 for these datasets, respectively. For PGM-Explainer, which generates node-level explanation, K is based on the number of nodes

Table 2: The hyper-parameters setting of NSEG(PNS^e) and NSEG($PNS^{e,f}$) among experimented datasets. The sub-script (e) denotes the hyper-parameters for M_e, and the sub-script (f) denotes the hyper-parameters for M_f.

	BA-Shapes	Tree-Cycles	Tree-Grid	BA2Motif	Mutagenicity	MSRC_21
PNS^e						
α_e	5.0e-3	1.0e-2	5.0e-2	1.0e-2	1.0e-4	1.0e-3
β_e	1.0	1.0	1.0	1.0	1.0e-3	1.0
$PNS^{e,f}$						
α_e	5.0e-3	1.0e-2	1.0e-2	1.0e-2	1.0e-4	5.0e-4
β_e	1.0	1.0	1.0	1.0	1.0	1.0
α_f	5.0e-3	1.0e-3	1.0e-2	1.0e-2	1.0e-4	5.0e-4
β_f	1.0	1.0	1.0	1.0	1.0	1.0

in the ground-truths, with values of 5, 6, 9, 5 and 10 for these datasets, respectively.

Hyper-parameter Setting. We apply a grid search to tune the hyper-parameters for our NSEG. The detailed hyper-parameters settings of NSEG(PNS^e) and NSEG($PNS^{e,f}$) are shown in Table 2. Regarding the baselines, we carefully tune the hyper-parameters based on their respective reported settings.

Baselines. To verify the effectiveness of PNS, we consider three variants of NSEG, named NSEG(PN), NSEG(PS), and NSEG(PNS), which optimize the lower bound of the probability of necessity, the probability of sufficiency, and the probability of necessity and sufficiency respectively. Furthermore, we use a superscript to indicate if edge (e) or feature (f) is considered in the explanations, e.g., NSEG(PNS^e) means only edge explanations are considered by our full PNS model. We compare our NSEG with the state-of-the-art baselines including the following:

- **Random** [1] generates random edge explanations.

- **GuidedBP** [12] generates explanation via gradient with negative gradients clipped during backpropagation.

- **GNNExplainer** [57] generates explanations by maximizing the mutual information between explanation subgraph and model prediction.

- **PGExplainer** [32] generates explanations from parameterized networks whose objective is to maximize the mutual information, similar to GNNExplainer.

- **PGM-Explainer** [50] generates Bayesian networks as explanations upon the perturbation-prediction data to identify significant nodes.

- **CF-GNNExplainer** [29] generates counterfactual explanations capable of flipping the model prediction subject to minimal perturbation.

- **CF**^{2} [47] generates explanations based on factual and counterfactual reasoning.

Note that for those baseline approaches that only generate node explanations (Grad, GuidedBP, and PGM-Explainer), we we will not report the *fidelity* since there is no edge explanation obtained, and we only report the *Recall@K*. Regarding the implementations of baselines, we adopt their original settings, as detailed in Appendix C.2. Since all baseline approaches generate only edge explanations instead of the joint explanations of edge and node features, for a fair comparison, we mainly compare the results of NSEG(PNS^e), NSEG(PN^e), and NSEG(PS^e) (which only generate edge explanations) with baseline methods in quantitative analysis, and we also showcase the results of NSEG($PNS^{e,f}$), which generate joint explanations, in both quantitative and qualitative analyses.

6.3. Are the Explanations Necessary and Sufficient? (RQ1)

We evaluate the necessity and sufficiency aspects of the explanations based on the *Fid+*, *Fid-*, and *charact* metrics [3] in Table 3 and Figure 4 for continuous and discrete explanations, respectively.

First, when comparing with the baselines, NSEG(PNS) (including NSEG(PNS^e) and NSEG($PNS^{e,f}$)) consistently outperforms the other methods, achieving the highest $charact^c$ scores for continuous explanations in most cases and the second winner on MSRC_21 dataset. When discrete explanations are obtained by thresholding, NSEG(PNS) achieves the best and relatively stable $Fid+^d$, $Fid-^d$, and $charact^d$ at most cases and the second best on MSRC_21. The second best results on MSRC_21 can be due to, the winner GNNExplainer that searches for sufficient explanations yields the extremely lowest $Fid-^c$ and $Fid-^d$, thus leads to the highest $charact^c$ and $charact^d$, respectively.

Secondly, among the variants of NSEG, we observe that NSEG(PN^e) provides higher $Fid+^c$ and $Fid+^d$ scores compared to NSEG(PS^e), indicating that NSEG(PN^e) focuses more on necessity. Conversely, NSEG(PS^e) emphasizes sufficiency, as reflected by its lower $Fid-^c$ and $Fid-^d$ scores compared to NSEG(PN^e). The full model, NSEG(PNS^e), combines the benefits of both aspects, achieving the highest $charact^c$ and $charact^d$ scores in most cases, with one exception on $charact^d$ on Tree-Cycles datasets. This can be due to, the discretization process damages the necessary aspect of the explanation leading to extremely low $Fid+^d$, thus causes low $charact^d$.

In summary, NSEG(PNS) consistently generates the most necessary and sufficient explanations in most cases

Table 3: $Fid+^c$(%), $Fid\text{-}^c$ (%) and $charact^c$ (%) of the explanations on GCN. Mean and standard deviation are reported. The best result of each metric is bolded.

	Node Classification								
	BA-Shapes			Tree-Cycles			Tree-Grid		
	$Fid+^c$ (↑)	$Fid\text{-}^c$ (↓)	$charact^c$ (↑)	$Fid+^c$ (↑)	$Fid\text{-}^c$ (↓)	$charact^c$ (↑)	$Fid+^c$ (↑)	$Fid\text{-}^c$ (↓)	$charact^c$ (↑)
Random(3-hops)	57.25±1.97	58.20±1.12	48.30±1.01	77.22±1.82	78.28±1.28	33.88±1.55	87.22±0.55	86.28±0.65	23.71±0.98
GuidedBP	47.75±0.00	77.00±0.00	31.05±0.00	21.67±0.00	85.56±0.00	17.33±0.00	94.17±0.00	26.94±0.00	82.28±0.00
GNNExplainer	46.95±0.83	72.00±0.00	35.08±0.23	77.61±1.34	18.22±0.72	79.63±0.81	90.03±0.90	40.58±0.87	71.58±0.68
PGExplainer	58.45±0.90	59.45±0.10	47.88±0.26	93.61±3.80	95.56±1.24	8.46±2.27	88.06±10.12	94.63±1.41	10.10±2.53
CFGNNExplainer	33.85±0.25	79.75±0.00	25.34±0.07	78.39±0.62	6.11±0.00	85.44±0.37	43.83±0.38	59.53±0.57	42.08±0.44
CF^2	46.60±0.25	72.00±0.00	34.98±0.07	78.89±0.00	21.11±0.00	78.89±0.00	43.89±0.00	56.11±0.00	43.89±0.00
NSEG(PN^e)	**100.00±0.00**	66.60±0.12	50.07±0.14	**100.00±0.00**	**0.00±0.00**	**100.00±0.00**	99.31±0.00	11.53±0.00	93.58±0.00
NSEG(PS^e)	62.75±0.00	**0.00±0.00**	77.11±0.00	98.89±0.00	**0.00±0.00**	99.44±0.00	95.50±0.07	**0.00±0.00**	97.70±0.04
NSEG(PNS^e)	99.60±0.12	**0.00±0.00**	99.80±0.06	**100.00±0.00**	**0.00±0.00**	**100.00±0.00**	**100.00±0.00**	**0.00±0.00**	**100.00±0.00**
NSEG(PNS^{e,f})	**100.00±0.00**	**0.00±0.00**	**100.00±0.00**	**100.00±0.00**	**0.00±0.00**	**100.00±0.00**	**100.00±0.00**	**0.00±0.00**	**100.00±0.00**
	Graph Classification								
	BA2Motif			Mutagenicity			MSRC_21		
	$Fid+^c$ (↑)	$Fid\text{-}^c$ (↓)	$charact^c$ (↑)	$Fid+^c$ (↑)	$Fid\text{-}^c$ (↓)	$charact^c$ (↑)	$Fid+^c$ (↑)	$Fid\text{-}^c$ (↓)	$charact^c$ (↑)
Random	50.50±0.00	50.50±0.00	49.99±0.00	94.64±0.70	94.48±0.30	10.43±0.54	80.16±0.78	80.16±0.60	31.80±0.79
GuidedBP	50.50±0.00	50.50±0.00	50.00±0.00	88.40±0.00	58.80±0.00	56.20±0.00	79.60±0.00	77.60±0.00	34.96±0.00
GNNExplainer	46.40±1.20	50.50±0.00	47.89±0.64	12.80±1.54	89.44±1.42	11.52±1.24	87.36±0.54	**2.40±0.00**	**92.20±0.30**
PGExplainer	50.50±0.00	50.50±0.00	49.99±0.00	95.20±0.00	95.20±0.00	9.14±0.00	80.64±1.78	80.96±2.00	30.77±2.71
CFGNNExplainer	**97.50±1.10**	50.50±0.00	65.66±0.25	93.76±0.93	95.20±0.00	9.13±0.00	76.16±1.49	91.60±0.00	15.13±0.03
CF^2	50.50±0.00	50.50±0.00	49.99±0.00	98.80±0.00	0.80±0.00	99.00±0.00	**97.20±0.00**	52.40±0.00	63.90±0.00
NSEG(PN^e)	81.90±0.92	18.30±0.75	81.80±0.58	**99.90±0.17**	83.30±0.17	28.62±0.25	96.40±0.00	51.70±0.17	64.36±0.15
NSEG(PS^e)	0.00±0.00	50.50±0.00	0.00±0.00	91.33±0.50	**0.27±0.19**	95.35±0.21	43.60±0.33	55.87±0.19	43.86±0.25
NSEG(PNS^e)	83.50±1.70	**17.60±0.37**	**82.94±1.01**	99.60±0.00	0.80±0.00	**99.40±0.00**	96.00±0.00	33.12±0.47	78.84±0.33
NSEG(PNS^{e,f})	73.50±1.38	47.90±0.73	60.97±0.69	95.12±0.16	86.48±0.78	23.67±1.20	93.52±0.39	61.44±0.20	54.61±0.24

compared to the baselines and other variants of our approach.

6.4. Are the Necessary and Sufficient Explanations Accurate? (RQ2)

We employ *Recall@K* and *ROC-AUC* to judge if the necessary and sufficient explanations are accurate. As there is only edge-based ground truth on the first three synthetic datasets, we only generate edge explanations using NSEG(PNS^e) on those datasets compared with the other baselines, as shown in Table 4.

The results indicate that our approach generally achieves the highest *Recall@K* except on Tree-Cycles datasets, and slightly lower *ROC-AUC* on Tree-Cycles, BA2Motif, and Mutagenicity. The slightly lower *Recall@K* and *ROC-AUC* of our approach may be attributed to the fact that NSEG prioritizes finding the most necessary and sufficient explanations rather than focusing on ranking the explanations. However, *Recall@K* and *ROC-AUC* metrics are sensitive to ranking, which could explain the relative difference in performance on datasets like Tree-Cycles.

6.5. Qualitative Studies

Besides quantitative analysis, visual explanations can better help humans understand the decision-making process of GNNs. Hence, we present a series of qualitative studies as follows.

Explanations of NSEG and Baselines. Figure 5 illustrates explanations obtained by different approaches in Mutagenicity and MSRC_21 datasets. In Figure 5 (a), we observe that NSEG(PNS) successfully identifies the

well-known explanation, the nitro-group (-NO2), in the Mutagenicity dataset, while maintaining the integrity of the molecule by including the benzene ring and excluding the hydrogen bond (-H). Additionally, NSEG(PNS^{e,f}) generates a finer-grained explanation by identifying the node features of nitrogen (N), oxygen (O), and carbon (C) atoms. In Figure 5 (b), the prediction of the instance graph is a scene about "chair". For edge-level explanations, we observe that our approach NSEG(PNS) successfully identifies all edges among the node "chair" as the explanation, while GNNExplainer generates the explanation including not only edges among nodes "chair" but also edges among nodes "void", and CF-GNNExplainer only identifies the edges between nodes "chair" and nodes "void". For the node-level explanation generated by NSEG(PNS^{e,f}) (highlighted nodes in the rightmost column of Figure 5 (b)), the inclusion of nodes related to the chair ensures the explanation's sufficiency. However, some "void" nodes may also increase the probability of predicting "chair" because they are strongly associated with the "chair" nodes in the training set. These scattered "void" nodes are unlikely to be shared by other "chair"-predicted samples, making the explanation sufficient but unnecessary. This is expected, since the strength of our NSEG lies in its ability to generate explanations that are both as sufficient and necessary as possible.

Explanations of NSEG Variants. Figure 6 illustrates the explanations obtained by the variants of NSEG, which are NSEG(PN^e), NSEG(PS^e), and NSEG(PNS^e) respectively. Compared to the baselines, NSEG(PNS^e) identifies the well-known explanation, the amino-group (-NH2) [57, 32], while keeping the integrity of the molecule by in-
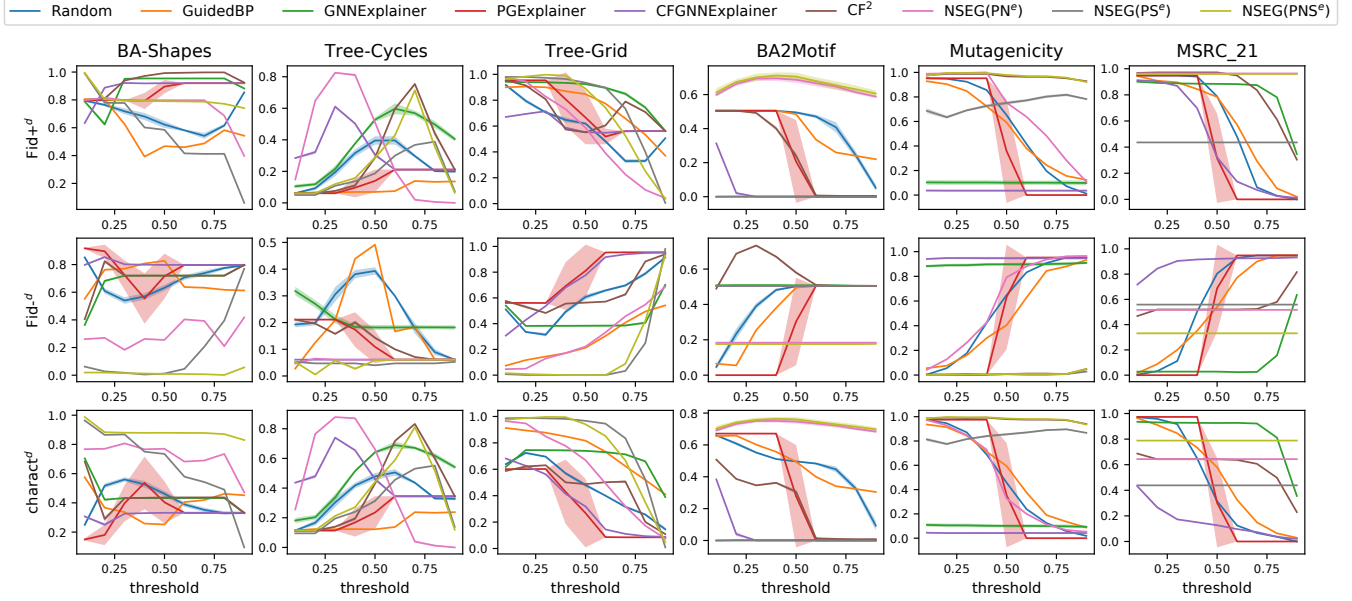
Figure 4: $Fid+^d$, $Fid\text{-}^d$, and $charact^d$ w.r.t. threshold on BA-Shapes, Tree-Cycles, Tree-Grid, Mutagenicity, and MSRC_21 datasets.

Table 4: *Recall@K* (%) and *ROC-AUC* (%) of the explanations on GCN. Mean and standard deviation are reported. The best result of each metric is bolded. Approach with symbol * outputs node-level explanations. The K values for *Recall@K* in the table are for methods generating edge-level explanations. For methods generating node-level explanations, the K values for these datasets are 5, 6, 9, 5, and 10.

	BA-Shapes		Tree-Cycles		Tree-Grid		BA2Motif		Mutagenicity	
	Recall@6	ROC-AUC	Recall@6	ROC-AUC	Recall@12	ROC-AUC	Recall@5	ROC-AUC	Recall@15	ROC-AUC
Random	33.33±0.38	50.09±0.29	66.94±0.31	50.82±0.38	64.04±0.26	49.82±0.21	22.19±0.97	50.45±0.80	40.20±0.87	49.71±0.85
GuidedBP	88.38±0.00	99.63±0.00	77.69±0.00	74.50±0.00	72.11±0.00	67.63±0.00	64.07±0.00	**88.44±0.00**	82.75±0.00	**84.06±0.00**
GNNExplainer	79.47±0.41	79.23±0.44	67.16±0.26	54.89±0.57	68.78±0.03	61.59±0.26	19.38±0.48	50.78±0.38	80.43±0.70	69.17±0.15
PGExplainer	79.89±11.35	96.31±4.04	**83.61±5.00**	**79.47±5.00**	66.43±1.17	54.09±6.17	49.43±4.30	82.59±2.37	78.81±6.75	79.66±5.19
PGM-Explainer*	65.01±0.69	-	82.35±0.69	-	72.91±0.20	-	26.48±1.15	-	46.55±1.63	-
CFGNNExplainer	75.42±0.07	95.40±0.11	73.41±0.05	66.97±0.13	72.52±0.07	68.21±0.10	60.25±0.18	85.52±0.11	58.64±0.26	59.64±0.07
CF²	86.07±0.20	99.31±0.00	68.71±0.06	59.51±0.09	69.54±0.08	62.56±0.14	27.57±0.44	66.13±0.24	70.58±0.50	76.50±0.08
NSEG(PNS^e)	86.48±0.12	98.44±0.08	71.92±0.16	66.57±0.08	**73.27±0.07**	**69.17±0.06**	**68.75±0.59**	85.06±0.23	67.95±0.19	74.03±0.12
NSEG(PNS^{e,f})	**89.96±0.07**	**99.67±0.00**	72.82±0.13	63.13±0.14	72.62±0.07	67.32±0.05	61.43±0.20	86.33±0.12	**84.69±0.69**	70.97±0.37

cluding the benzene ring and excluding the hydrogen bond (-H). In contrast, the ablation NSEG(PN^e) only identifies the benzene ring (which is necessary but not sufficient), while NSEG(PS^e) identifies almost the whole graph (which is sufficient but not necessary).

Explanations for Incorrect and Correct Predictions. To gain insights into the model's decision-making process, it is crucial to explain both correct and incorrect predictions. Hence, we showcase two explanations of the instances with correct and incorrect predictions. The first instance in Figure 7(a) is a scene about "water, boat...", but the GNN incorrectly predicts "road, building...". The generated explanation suggests the reason behind the misclassification—the GNN paid more attention to the nodes "road", "building", and the edges among them. The second instance in Figure 7(b) is a scene about "dog", which is correctly predicted. The generated explanation shows that more attention was paid to the nodes "dog" and edges among them, leading to the correct prediction. Additionally, more qualitative studies are shown in Appendix D.2.

Explanations with both Necessity and Sufficiency. In Figure 8 a compound is predicted as a "mutagenic", whose necessary and sufficient explanation $\mathbf{E} = E'_i, \mathbf{X} = X'_i$ can produce the prediction "mutagen" with a high probability 0.93, showing the sufficiency of the explanation. Also, in the last three columns of Figure 8, we present one example for each of the cases: $(\mathbf{E} \neq E'_i, \mathbf{X} \neq X'_i)$, $(\mathbf{E} \neq E'_i, \mathbf{X} = X'_i)$, and $(\mathbf{E} = E'_i, \mathbf{X} \neq X'_i)$, which degrades the probabilities of producing "mutagen" to 0.44, 0.41, and 0.38, respectively. Such changes imply the necessity of the explanation.

6.6. Sensitivity Analysis

Sensitivity of α and β. The hyper-parameter α and β in Eq. (19) modulates the *mask size* and *mask entropy* regularization terms, the components also utilized in prior works [57, 32, 47, 30]. Intuitively, the *mask size* regularization operates as an L_1 penalization, compelling the explanation to emphasize the most salient portions of the input, while the *mask entropy* forces the mask concentrating around a few scalars. Within our experimental setup, we scrutinized the impact of α_e and β_e on NSEG(PNS^e), where α_e and β_e govern the *mask size* and *mask entropy*

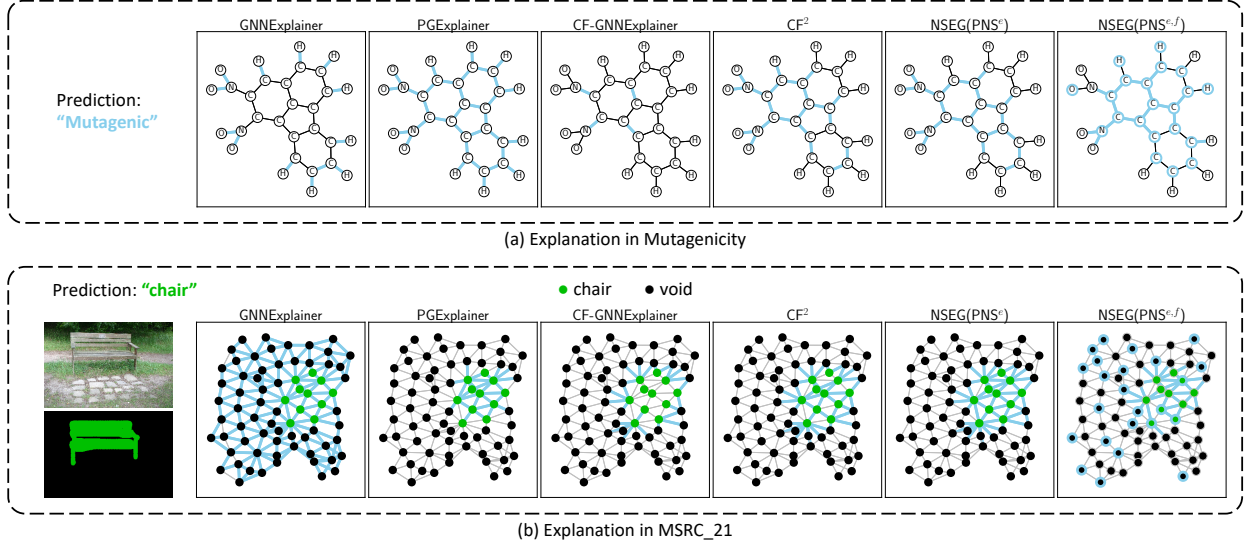(a) Explanation in Mutagenicity

(b) Explanation in MSRC_21

Figure 5: Explanations of GNNExplainer, PGExplainer, CF-GNNExplainer, CF^2, NSEG(PNS^e), and NSEG($PNS^{e,f}$) in Mutagenicity and MSRC_21, where the explanations are highlighted in blue.
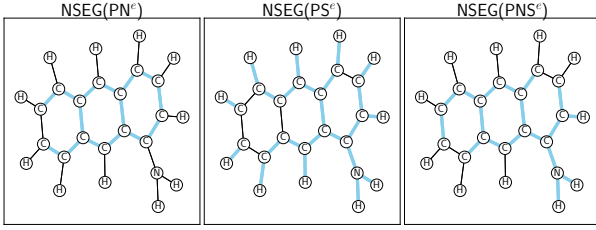

Figure 6: Explanations of NSEG variants, NSEG(PN^e), NSEG(PS^e), and NSEG(PNS^e) on a Mutagenicity instance, where the explanations are highlighted in blue.

regularization of the edge mask M_e, respectively. The implications of varying α_e on NSEG(PNS^e) across the metrics of *charact* and *Recall@K* are depicted in Figure 9. Two key observations emerge from our analysis:

- Regarding α_e, metrics $Fid+^c$, $Fid\text{-}^c$ and $charact^c$ remains largely impervious to alterations in α_e, while in contrast, metric AUC displays a pronounced sensitivity to α_e.

- Regarding β_e, metrics α_e, metrics $Fid+^c$, $Fid\text{-}^c$, $charact^c$, and AUC show non-sensitive to α_e, evidenced by achieving stable results on these metrics.

Sensitivity of the Priors. We present the sensitivity analysis for the prior probabilities P_{00}, P_{01}, and P_{10} (Eq. (11)) in our NSEG. Specifically, we first determined the appropriate ranges for P_{00}, P_{01}, and P_{10} around 1/3, based on the prior knowledge discussed in Section 4.3.2, using the range [0.2, 0.4]. We then fixed P_{00} at 0.4 and linearly sampled 5 values for P_{01} and P_{10} at equal intervals within their respective ranges. As shown in Figure 10, NSEG's performance, evaluated using $Fid+^c$, $Fid\text{-}^c$, $charact^c$, and AUC, remains stable. This indicates that as long
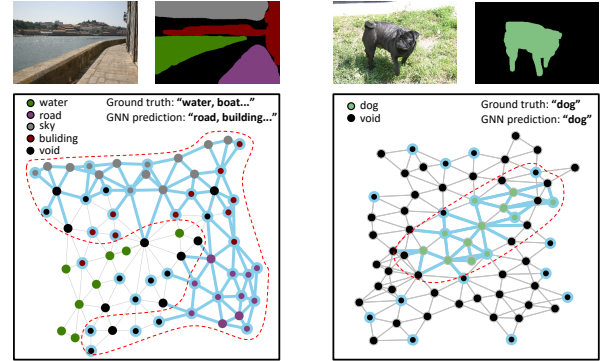
(a) Explanation of an incorrectly predicted instance. The GNN falsely pays more attention to the nodes "sky", "road", and "building" and the edges among them.

(b) Explanation of a correctly predicted instance. The GNN correctly pays more attention to the nodes "dog" and the edges among them.

Figure 7: Explanations of NSEG($PNS^{e,f}$) on two MSRC_21 instances, where the explanations are highlighted in blue and enclosed by a red dashed line.

as the priors are within an appropriate range, NSEG's performance is less sensitive to their values.

7. Conclusion and Discussion

In this paper, we proposed NSEG, a GNN explanation model that is able to provide necessary and sufficient explanations for GNNs. Different from the existing approaches that generate either necessary or sufficient explanations, or a heuristic trade-off explanation between the two aspects, our objective is to maximize PNS that ensures the explanation is both necessary and sufficient. To overcome the intractability of identifying PNS, we derived a lower bound of PNS, and further propose a gen-
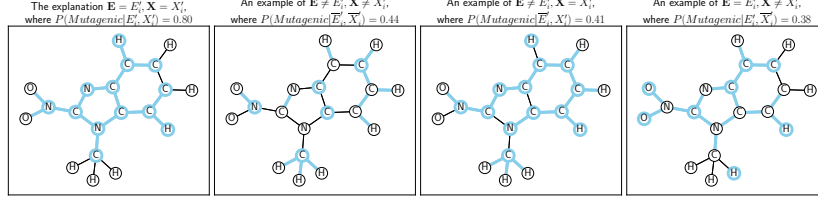
Figure 8: Explanation of $NSEG(PNS^{e,f})$ on a Mutagenicity instance, where the explanation is highlighted in blue.

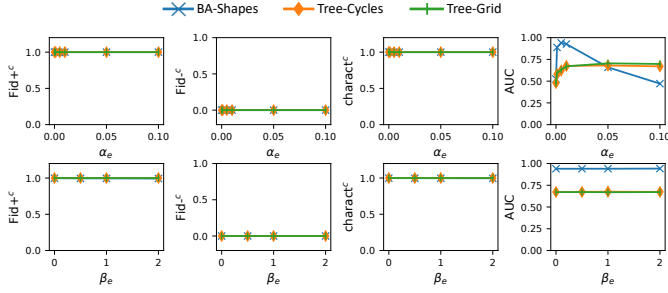Figure 9: $Fid+^c$, $Fid-^c$, $charact^c$ and AUC w.r.t. α_e on $NSEG(PNS^e)$ across BA-Shapes, Tree-Cycles, and Tree-Grid datasets.

eralized SCM of GNN for estimating this lower bound. For optimization, we leverage continuous masks with sampling strategy to optimize the lower bound. Our empirical study demonstrates that NSEG can provide the most necessary and sufficient explanations and achieves state-of-the-art performance on various datasets.

However, the objective function of NSEG depends on the selection of priors, even though an appropriate range can be ensured based on prior knowledge. In future work, we believe a more prior-free algorithm deserves further attention to address the limitations of NSEG. Despite these limitations, we believe our NSEG contributes to machine learning explainability, particularly in terms of necessity and sufficiency, and inspires the design of more interpretable models.

References

[1] Agarwal, C., Queen, O., Lakkaraju, H., Zitnik, M., 2023. Evaluating explainability for graph neural networks. Scientific Data 10, 144.

[2] Alvarez-Melis, D., Jaakkola, T.S., 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models, in: EMNLP, Association for Computational Linguistics. pp. 412–421.

[3] Amara, K., Ying, R., Zhang, Z., Han, Z., Shan, Y., Brandes, U., Schemm, S., Zhang, C., 2022. Graphframex: Towards systematic evaluation of explainability methods for graph neural networks.

[4] Arrieta, A.B., Rodríguez, N.D., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion 58, 82–115.

[5] Bai, L., Cui, L., Wang, Y., Li, M., Li, J., Yu, P.S., Hancock, E.R., 2024. Haqjsk: Hierarchical-aligned quantum jensen-shannon kernels for graph classification. IEEE Transactions
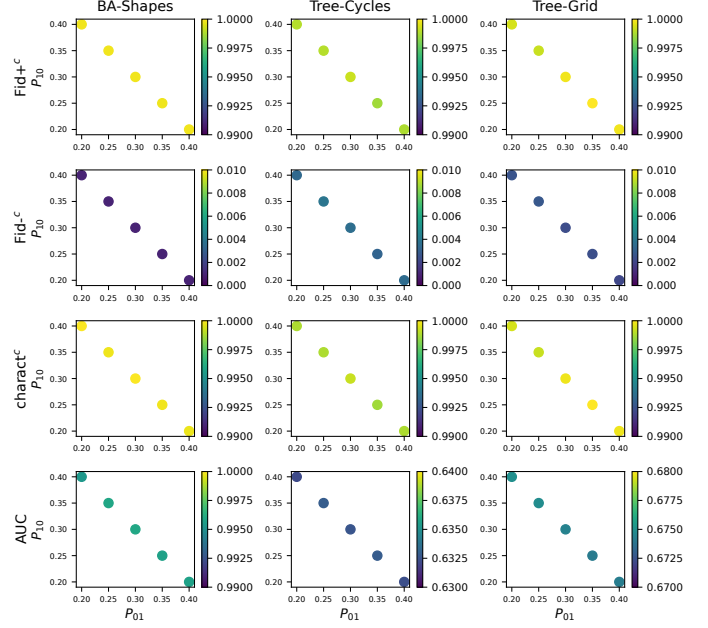
Figure 10: $Fid+^c$, $Fid-^c$, $charact^c$ and AUC w.r.t. P_{01} and P_{10} on $NSEG(PNS^{e,f})$ across BA-Shapes, Tree-Cycles, and Tree-Grid datasets, where $P_{00} = 1 - P_{01} - P_{10}$.

on Knowledge and Data Engineering , 1–14doi:10.1109/TKDE. 2024.3389966.

[6] Baldassarre, F., Azizpour, H., 2019. Explainability techniques for graph convolutional networks. doi:10.48550/ARXIV.1905. 13686, arXiv:1905.13686.

[7] Bareinboim, E., Forney, A., Pearl, J., 2015. Bandits with unobserved confounders: A causal approach, in: Advances in Neural Information Processing Systems, pp. 1342–1350.

[8] Beckers, S., 2022. Causal explanations and XAI, in: 1st Conference on Causal Learning and Reasoning, CLeaR 2022, PMLR. pp. 90–109.

[9] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine learning 3, 1–122.

[10] Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2014. Spectral networks and locally connected networks on graphs, in: ICLR.

[11] Chattopadhyay, A., Manupriya, P., Sarkar, A., Balasubramanian, V.N., 2019. Neural network attributions: A causal perspective, in: ICML, PMLR. pp. 981–990.

[12] Chen, M., Wei, Z., Ding, B., Li, Y., Yuan, Y., Du, X., Wen, J., 2020. Scalable graph neural networks via bidirectional propagation, in: NeurIPS.

[13] Chen, X., Cai, R., Fang, Y., Wu, M., Li, Z., Hao, Z., 2023. Motif graph neural network. IEEE Transactions on Neural Networks and Learning Systems , 1–15doi:10.1109/TNNLS.2023.3281716.

[14] Dou, Y., Shu, K., Xia, C., Yu, P.S., Sun, L., 2021. User preference-aware fake news detection, in: SIGIR, ACM. pp.

2051–2055.

[15] Fan, W., Ma, Y., Li, Q., He, Y., Zhao, Y.E., Tang, J., Yin, D., 2019. Graph neural networks for social recommendation, in: The World Wide Web Conference, ACM. pp. 417–426. doi:10.1145/3308558.3313488.

[16] Galhotra, S., Pradhan, R., Salimi, B., 2021. Explaining black-box algorithms using probabilistic contrastive counterfactuals, in: SIGMOD '21: International Conference on Management of Data, ACM. pp. 577–590.

[17] Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E., 2017. Neural message passing for quantum chemistry, in: ICML, PMLR. pp. 1263–1272.

[18] Hamilton, W.L., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, pp. 1024–1034.

[19] Huang, C., Xu, H., Xu, Y., Dai, P., Xia, L., Lu, M., Bo, L., Xing, H., Lai, X., Ye, Y., 2021. Knowledge-aware coupled graph neural network for social recommendation, in: AAAI Conference on Artificial Intelligence, AAAI Press. pp. 4115–4122.

[20] Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., Chang, Y., 2020. Graphlime: Local interpretable model explanations for graph neural networks. arXiv:2001.06216.

[21] Jung, Y., Kasiviswanathan, S.P., Tian, J., Janzing, D., Blöbaum, P., Bareinboim, E., 2022. On measuring causal contributions via do-interventions, in: ICML, PMLR. pp. 10476–10501.

[22] Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes, in: ICLR.

[23] Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, OpenReview.net.

[24] Klicpera, J., Bojchevski, A., Günnemann, S., 2019. Predict then propagate: Graph neural networks meet personalized pagerank, in: ICLR (Poster), OpenReview.net.

[25] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324. doi:10.1109/5.726791.

[26] Li, J., Zheng, R., Feng, H., Li, M., Zhuang, X., 2024. Permutation equivariant graph framelets for heterophilous graph learning. IEEE Transactions on Neural Networks and Learning Systems , 1–15doi:10.1109/TNNLS.2024.3370918.

[27] Lin, W., Lan, H., Li, B., 2021. Generative causal explanations for graph neural networks, in: International Conference on Machine Learning, PMLR. pp. 6666–6679.

[28] Liu, M., Gao, H., Ji, S., 2020. Towards deeper graph neural networks, in: KDD, ACM. pp. 338–348.

[29] Lucic, A., ter Hoeve, M.A., Tolomei, G., de Rijke, M., Silvestri, F., 2022a. Cf-gnnexplainer: Counterfactual explanations for graph neural networks, in: AISTATS, PMLR. pp. 4499–4511.

[30] Lucic, A., ter Hoeve, M.A., Tolomei, G., de Rijke, M., Silvestri, F., 2022b. Cf-gnnexplainer: Counterfactual explanations for graph neural networks, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 4499–4511.

[31] Lundberg, S.M., Lee, S., 2017. A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, pp. 4765–4774.

[32] Luo, D., Cheng, W., Xu, D., Yu, W., Zong, B., Chen, H., Zhang, X., 2020. Parameterized explainer for graph neural network, in: Advances in Neural Information Processing Systems.

[33] Magister, L.C., Kazhdan, D., Singh, V., Liò, P., 2021. Gc-explainer: Human-in-the-loop concept-based explanations for graph neural networks. arXiv:2107.11889.

[34] Microsoft, 2017. Image understanding. https://www.microsoft.com/en-us/research/project/image-understanding. Accessed: 2022-08-01.

[35] Molnar, C., 2022. Interpretable Machine Learning. 2 ed.

[36] Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M., 2020. Tudataset: A collection of benchmark datasets for learning with graphs arXiv:2007.08663.

[37] Pearl, J., 1999. Probabilities of causation: Three counterfactual interpretations and their identification. Synth. 121, 93–149.

[38] Pearl, J., 2009. Causality: Models, Reasoning and Inference. 2nd ed., Cambridge University Press, USA.

[39] Pfeifer, B., Saranti, A., Holzinger, A., 2022. GNN-SubNet: disease subnetwork detection with explainable graph neural networks. Bioinformatics 38, ii120–ii126. doi:10.1093/bioinformatics/btac478.

[40] Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H., 2019. Explainability methods for graph convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, Computer Vision Foundation / IEEE. pp. 10772–10781. doi:10.1109/CVPR.2019.01103.

[41] Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "why should I trust you?": Explaining the predictions of any classifier, in: International Conference on Knowledge Discovery and Data Mining, ACM. pp. 1135–1144. doi:10.1145/2939672.2939778.

[42] Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y., 2021. Towards causal representation learning. arXiv:2102.11107.

[43] Shan, C., Shen, Y., Zhang, Y., Li, X., Li, D., 2021. Reinforcement learning enhanced explainer for graph neural networks, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 22523–22533.

[44] Shi, C., Xu, M., Zhu, Z., Zhang, W., Zhang, M., Tang, J., 2020. Graphaf: a flow-based autoregressive model for molecular graph generation, in: International Conference on Learning Representations, OpenReview.net.

[45] Sun, M., Zhao, S., Gilvary, C., Elemento, O., Zhou, J., Wang, F., 2020. Graph convolutional networks for computational drug development and discovery. Briefings Bioinform. 21, 919–935.

[46] Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning, PMLR. pp. 3319–3328.

[47] Tan, J., Geng, S., Fu, Z., Ge, Y., Xu, S., Li, Y., Zhang, Y., 2022. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning, in: The ACM Web Conference, ACM. pp. 1018–1027. doi:10.1145/3485447.3511948.

[48] Tian, J., Pearl, J., 2000. Probabilities of causation: Bounds and identification, in: Conference in Uncertainty in Artificial Intelligence, Morgan Kaufmann. pp. 589–598.

[49] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks, in: International Conference on Learning Representations, OpenReview.net.

[50] Vu, M.N., Thai, M.T., 2020. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks, in: Advances in Neural Information Processing Systems.

[51] Wang, X., Wu, Y., Zhang, A., Feng, F., He, X., Chua, T.S., 2022. Reinforced causal explainer for graph neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence , 1–1doi:10.1109/TPAMI.2022.3170302.

[52] Watson, D.S., Gultchin, L., Taly, A., Floridi, L., 2021. Local explanations via necessity and sufficiency: unifying theory and practice, in: Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, AUAI Press. pp. 1382–1392.

[53] Wu, F., Jr., A.H.S., Zhang, T., Fifty, C., Yu, T., Weinberger, K.Q., 2019. Simplifying graph convolutional networks, in: ICML, PMLR. pp. 6861–6871.

[54] Xu, B., Shen, H., Sun, B., An, R., Cao, Q., Cheng, X., 2021. Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field, in: AAAI, AAAI Press. pp. 4537–4545.

[55] Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2019. How powerful are graph neural networks?, in: International Conference on Learning Representations, OpenReview.net.

[56] Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., Wang, J., 2021. Causalvae: Disentangled representation learning via neural structural causal models, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Computer Vision Foundation / IEEE. pp. 9593–9602.

[57] Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J., 2019.

Gnnexplainer: Generating explanations for graph neural networks, in: Advances in Neural Information Processing Systems, pp. 9240–9251.

[58] Yuan, H., Yu, H., Gui, S., Ji, S., 2020. Explainability in graph neural networks: A taxonomic survey. doi:10.48550/ARXIV.2012.15445, arXiv:2012.15445.

[59] Zeng, H., Zhou, H., Srivastava, A., Kannan, R., Prasanna, V.K., 2020. Graphsaint: Graph sampling based inductive learning method, in: ICLR, OpenReview.net.

[60] Zhang, M., Chen, Y., 2018. Link prediction based on graph neural networks, in: Advances in Neural Information Processing Systems, pp. 5171–5181.

[61] Zhang, W., Wu, T., Wang, Y., Cai, Y., Cai, H., 2023. Towards trustworthy explanation: On causal rationalization, in: International Conference on Machine Learning, ICML 2023, PMLR. pp. 41715–41736.

[62] Zhao, H., des Combes, R.T., Zhang, K., Gordon, G.J., 2019. On learning invariant representations for domain adaptation, in: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, PMLR. pp. 7523–7532.

Appendix A. Proof

Appendix A.1. Proof of Lemma 1

Proof. According to the definition of PNS in Definition 2, we have

$$\text{PNS}(\xi) = P(\mathbf{Y}_{\xi^c} \neq \hat{y}, \mathbf{Y}_\xi = \hat{y})$$
$$= P(\mathbf{Y}_{\xi^c} \neq \hat{y}) + P(\mathbf{Y}_\xi = \hat{y}) - P(\mathbf{Y}_{\xi^c} \neq \hat{y} \vee \mathbf{Y}_\xi = \hat{y}),$$

since $P(\mathbf{Y}_{\xi^c} \neq \hat{y} \vee \mathbf{Y}_\xi = \hat{y}) \leq 1$ and $P(\mathbf{Y}_{\xi^c} \neq \hat{y}, \mathbf{Y}_\xi = \hat{y}) \geq 0$, we have:

$$\text{PNS}(\xi) \geq \max\{0, P(\mathbf{Y}_{\xi^c} \neq \hat{y}) + P(\mathbf{Y}_\xi = \hat{y}) - 1\}.$$

$\square$

Appendix A.2. Proof of Lemma 2

Proof. Since $(\mathbf{Y}_{\xi^c} = \hat{y}) \vee (\mathbf{Y}_{\xi^c} \neq \hat{y}) = true$, we have:

$$(\mathbf{Y}_\xi = \hat{y})$$
$$= (\mathbf{Y}_\xi = \hat{y}) \wedge ((\mathbf{Y}_{\xi^c} = \hat{y}) \vee (\mathbf{Y}_{\xi^c} \neq \hat{y}))$$
$$= ((\mathbf{Y}_\xi = \hat{y}) \wedge (\mathbf{Y}_{\xi^c} = \hat{y})) \vee ((\mathbf{Y}_\xi = \hat{y}) \wedge (\mathbf{Y}_{\xi^c} \neq \hat{y})),$$

and since $(\mathbf{Y}_\xi = \hat{y}) \vee (\mathbf{Y}_\xi \neq \hat{y}) = true$, when monotonicity holds, i.e., $(\mathbf{Y}_\xi \neq \hat{y}) \wedge (\mathbf{Y}_{\xi^c} = \hat{y}) = false$, we have:

$$(\mathbf{Y}_{\xi^c} = \hat{y})$$
$$= (\mathbf{Y}_{\xi^c} = \hat{y}) \wedge ((\mathbf{Y}_\xi = \hat{y}) \vee (\mathbf{Y}_\xi \neq \hat{y}))$$
$$= ((\mathbf{Y}_{\xi^c} = \hat{y}) \wedge (\mathbf{Y}_\xi = \hat{y})) \vee ((\mathbf{Y}_{\xi^c} = \hat{y}) \wedge (\mathbf{Y}_\xi \neq \hat{y}))$$
$$= ((\mathbf{Y}_{\xi^c} = \hat{y}) \wedge (\mathbf{Y}_\xi \neq \hat{y})),$$

then, combining those 2 equations, we have:

$$(\mathbf{Y}_\xi = \hat{y})$$
$$= (\mathbf{Y}_{\xi^c} = \hat{y}) \vee ((\mathbf{Y}_\xi = \hat{y}) \wedge (\mathbf{Y}_{\xi^c} \neq \hat{y}))$$

then take the probability form and use the disjointness of $\mathbf{Y}_{\xi^c} = \hat{y}$ and $\mathbf{Y}_{\xi^c} \neq \hat{y}$:

$$P(\mathbf{Y}_\xi = \hat{y}) = P(\mathbf{Y}_{\xi^c} = \hat{y}) + P(\mathbf{Y}_\xi = \hat{y}, \mathbf{Y}_{\xi^c} \neq \hat{y})$$

then:
$$\text{PNS}(\xi) = P(\mathbf{Y}_\xi = \hat{y}) - P(\mathbf{Y}_{\xi^c} = \hat{y})$$
$$= P(\mathbf{Y}_\xi = \hat{y}) + P(\mathbf{Y}_{\xi^c} \neq \hat{y}) - 1$$

$\square$

Appendix A.3. Proof of Proposition 1

Proof. In a graph neural network, each node's hidden representation in each hidden layer can be written as functions of entries of graph and all nodes' representations in the previous layer, i.e.,

$$\forall v \forall k (\mathbf{h}_i^{(k)} = f_\theta^{(k)}(\mathbf{E}_{*,i}, \mathbf{H}^{(k-1)})),$$

where $\mathbf{h}_v^{(k)}$ denotes the hidden representation of node v in k-th layer, $\mathbf{H}^{(k-1)})$ denotes the hidden representations of all nodes in $(k-1)$-th layer, $\mathbf{E}_{*,i}$ denotes the entries from all nodes to node i. Then for all nodes' representations, we have,

$$\forall k (\mathbf{H}^{(k)} = f_\theta^{(k)}(\mathbf{E}, \mathbf{H}^{(k-1)})),$$

16

where $\mathbf{E}$ denotes the edges of the graph.

The output variable $\mathbf{Y}$ is determined by all nodes' representations in l-th layer by a read-out function $f_\theta^{(l+1)}$ (specifically, a node-level read-out for node classification task and a graph-level read-out for graph classification task), i.e.,

$$\mathbf{Y} = f_\theta^{(l+1)}(\mathbf{H}^{(l)}).$$

And for the input variables $\mathbf{E}$ and $\mathbf{X}$, they are determined by a set of exogenous $\mathbf{U}$, i.e.,

$$\mathbf{E}, \mathbf{X} = f^{(0)}(\mathbf{U}).$$

Thus, the structure of those variables can be equivalently expressed by a SCM $M(\{\mathbf{X}, \mathbf{E}, \mathbf{H}^{(1)}, ..., \mathbf{H}^{(l)}, \mathbf{Y}\}, \mathbf{U}, \{f^{(0)}, f_\theta^{(1)}, ..., f_\theta^{(l+1)}\})$. $\square$

Appendix A.4. Proof of Proposition 2

Proof. Start from the output variable $\mathbf{Y}$, the corresponding causal mechanism function $\mathbf{Y} = f_\theta^{(l+1)}(\mathbf{H}^{(l)})$ can be substituted as

$$\mathbf{Y} = f_\theta^{(l+1)}(f_\theta^{(l)}(\mathbf{E}, f_\theta^{(l-1)}(\mathbf{E}, ...f_\theta^{(1)}(\mathbf{E}, \mathbf{X})))).$$

Thus, we can obtain a new causal mechanism f_θ that is a mapping from $\mathbf{E}, \mathbf{X}$ to $\mathbf{Y}$, i.e.,

$$\mathbf{Y} = f_\theta(\mathbf{E}, \mathbf{X}).$$

Hence, we can obtain a reduced SCM $M(\{\mathbf{E}, \mathbf{X}, \mathbf{Y}\}, \mathbf{U}, \{f^{(0)}, f_\theta\})$. $\square$

Appendix A.5. Proof of Proposition 3

Proof. First, we can derive lower bounds of both PN and PS as follows.

$$
\begin{aligned}
\text{PN}(\xi) &= P(\mathbf{Y}_{\xi^c} \neq \hat{y} | \xi, \mathbf{Y} = \hat{y}) \\
&= \frac{P(\mathbf{Y}_{\xi^c} \neq \hat{y}, \xi, \mathbf{Y} = \hat{y})}{P(\xi, \mathbf{Y} = \hat{y})} \\
&\overset{Proposition\ 2}{=} \frac{P(\mathbf{Y}_{\xi^c} \neq \hat{y}, \xi, \mathbf{Y}_\xi = \hat{y})}{P(\xi, \mathbf{Y} = \hat{y})} \\
&= \frac{P(\mathbf{Y}_{\xi^c} \neq \hat{y}, \mathbf{Y}_\xi = \hat{y})P(\xi)}{P(\xi, \mathbf{Y} = \hat{y})} \\
&= \frac{P(\mathbf{Y}_{\xi^c} \neq \hat{y}, \mathbf{Y}_\xi = \hat{y})}{P(\mathbf{Y} = \hat{y}|\xi)} \\
&\geq \frac{\max\{0, P(\mathbf{Y}_{\xi^c} \neq \hat{y}) + P(\mathbf{Y}_\xi = \hat{y}) - 1\}}{P(\mathbf{Y} = \hat{y}|\xi)} \\
&= \max\{0, \frac{P(\mathbf{Y}_{\xi^c} \neq \hat{y}) - 1}{P(\mathbf{Y} = \hat{y}|\xi)} + 1\}
\end{aligned}
$$

$$
\begin{aligned}
\text{PS}(\xi) &= P(\mathbf{Y}_\xi = \hat{y} | \xi^c, \mathbf{Y} \neq \hat{y}) \\
&= \frac{P(\mathbf{Y}_\xi = \hat{y}, \xi^c, \mathbf{Y} \neq \hat{y})}{P(\xi^c, \mathbf{Y} \neq \hat{y})} \\
&\overset{Proposition\ 2}{=} \frac{P(\mathbf{Y}_\xi = \hat{y}, \xi^c, \mathbf{Y}_{\xi^c} \neq \hat{y})}{P(\xi^c, \mathbf{Y} \neq \hat{y})} \\
&= \frac{P(\mathbf{Y}_\xi = \hat{y}, \mathbf{Y}_{\xi^c} \neq \hat{y})P(\xi^c)}{P(\xi^c, \mathbf{Y} \neq \hat{y})} \\
&= \frac{P(\mathbf{Y}_\xi = \hat{y}, \mathbf{Y}_{\xi^c} \neq \hat{y})}{P(\mathbf{Y} \neq \hat{y}|\xi^c)} \\
&\geq \frac{\max\{0, P(\mathbf{Y}_{\xi^c} \neq \hat{y}) + P(\mathbf{Y}_\xi = \hat{y}) - 1\}}{P(\mathbf{Y} \neq \hat{y}|\xi^c)} \\
&= \max\{0, \frac{P(\mathbf{Y}_\xi = \hat{y}) - 1}{P(\mathbf{Y}_{\xi^c} \neq \hat{y})} + 1\}
\end{aligned}
$$

Altering the event ξ with the explanation event $(\mathbf{E} = E_i', \mathbf{X} = X_i')$, and under the SCM depict in Proposition 2, the lower bounds of PN and PS after intervening on both $\mathbf{E}$ and $\mathbf{X}$ (replacing causal mechanism from $\mathbf{U}$ to $\mathbf{E}$ and $\mathbf{X}$), we have:

$$\text{PN}_{lb}^{e,f}(E_i', X_i') = \max\{0, \frac{P_\theta(\mathbf{Y} \neq \hat{y}|(\mathbf{E} = E_i', \mathbf{X} = X_i')^c) - 1}{P_\theta(\mathbf{Y} = \hat{y}|\mathbf{E} = E_i', \mathbf{X} = X_i')} + 1\},$$

$$\text{PS}_{lb}^{e,f}(E_i', X_i') = \max\{0, \frac{P_\theta(\mathbf{Y} = \hat{y}|\mathbf{E} = E_i', \mathbf{X} = X_i') - 1}{P_\theta(\mathbf{Y} \neq \hat{y}|(\mathbf{E} = E_i', \mathbf{X} = X_i')^c)} + 1\}.$$

$\square$

Appendix B. Optimization Details of the PN and PS Lower Bounds

In practice, directly optimizing the lower bounds of PN and PS in Proposition 3 for $\text{NSEG}(\text{PN}^e)$ and $\text{NSEG}(\text{PS}^e)$ often leads to suboptimal solutions, manifested by the dominance of the denominator in the optimization, while the numerator remains nearly unchanged. To alleviate this issue, since these two bounds are not the primary focus of this work, we simply fix their denominators and optimize only the numerators to avoid suboptimal solutions. Mathematically, combining the previously discussed continuous mask and sampling strategy, the objective functions for optimizing $\text{PN}_{lb}^{e,f}$ and $\text{PS}_{lb}^{e,f}$ are as follows.

$$
\begin{aligned}
\text{PN}_{lb}^{e,f} = &- P_{00}\mathbb{E}_{\epsilon_e, \epsilon_f}[f_\theta^{\hat{y}}((1 - M_e + \epsilon_e) \odot E_i, (1 - M_f + \epsilon_f) \odot X_i)] \\
&- P_{01}\mathbb{E}_{\epsilon_e}[f_\theta^{\hat{y}}((1 - M_e + \epsilon_e) \odot E_i, M_f \odot X_i)] \\
&- P_{10}\mathbb{E}_{\epsilon_f}[f_\theta^{\hat{y}}(M_e \odot E_i, (1 - M_f + \epsilon_f) \odot X_i)] + 1,
\end{aligned}
$$

$$\text{PS}_{lb}^{e,f} = f_\theta^{\hat{y}}(M_e \odot E_i, M_f \odot X_i).$$

For tasks requiring higher precision in the PN and PS lower bounds, we can apply existing approaches such as alternating optimization of the numerator and denominator [9].

Appendix C. Details of Experiment Setup

Appendix C.1. GNN Training Setup

In this section, we introduce the training setup and training result of the GNN we explain. For the GNN

training, we employ three layers of Graph Convolutional Networks (GCNs) [23] with ReLU activation. For node classification task, the dimensions of the hidden layers are 16, 32, and 16 respectively, followed by a fully connected layer as the output layer. For graph classification task, the dimensions of the hidden layer in graph classification task are 16, 32, and 16, and after the last GCN layer, a sum-based read-out is used to obtain a graph representation, followed by a fully connected layer as the output layer. For all datasets, we split train/val/test with 80 %/10 %/10 %. The hyperparameter setting and test accuracy of the trained GNN model are shown in Table C.5.

Appendix C.2. Baselines

GNNExplainer. The goal of GNNExplainer is to identify a subgraph G_i' and the associated features X_i' that are important to GNN's prediction $\hat{y}$. The objective of GNNExplainer is to maximize the mutual information between the subgraph explanation and the GNN model outcome. Intuitively, if knowing the information of the subgraph G_i' and its associated features X_i' can reduce the uncertainty of $\mathbf{Y}$, then G_i' and X_i' are good explanations for the GNN prediction. Equivalently, the mutual information objective captures the sufficiency aspect in producing an explanation.

$$\max_{G_i', X_i'} MI(\mathbf{Y}; (G_i', X_i'))$$

where MI quantifies the change in the probability of the prediction $\hat{y}$ when the input is limited to (G_i', X_i'). In particular, GNNExplainer converts the discrete optimization problem into continuous optimization and leverages trainable edge mask M_e and feature mask M_f in optimization, where the objective is derived as follow.

$$\min_{M_e, M_f} H(\mathbf{Y}|\mathbf{G} = G_i \odot M_e, \mathbf{X} = X_i \odot M_f)$$

Notably, the X_i' in GNNExplainer is quite different from the X_i' in NSEG, where the former is a feature dimension-wise explanation while the latter is a node-wise explanation. In implementations, we adopt the original settings where for node classification task the l-hop subgraph of the target node is extracted, where l denotes the number of graph convolution layers. For hyper-parameter that controls the *mask size* regularization, we select [0.01, 0.01, 0.01, 0.005, 0.005] for BA-Shapes, Tree-Cycles, Tree-Grid, Mutagenicity, and MSRC_21, respectively. For hyper-parameter that controls the *entropy* regularization, we set it to 1.0 for all datasets.

PGExplainer. Though PGExplainer shares the same objective with GNNExplainer, i.e., maximizing the mutual information between the subgraph explanation and the GNN outcome, PGExplainer spends way less time in inference than GNNExplainer. To achieve this, the key idea of PGExplainer is to learn a mapping from the graph representation space to subgraph space parameterized by

ω, and the subgraphs are sampled from the distribution $q(\omega)$. The parameterized mapping gives PGExplainer the inductive ability, which means the explainer once trained, it does not need to be retrained to explain different instances. In particular, the objective of PGExplainer is shown as follows.

$$\min_{\omega} \mathbb{E}_{G_i' \sim q(\omega)} H(\mathbf{Y}|G_i'),$$

where the sampling process $G_i' \sim q(\omega)$ is approximated via determinant function of parameters ω, temperature τ, and an independent random variable $\epsilon \sim U(0,1)$:

$$G_i' = f_{\omega}(G_i, \tau, \epsilon).$$

In implementations, we adopt the original annealing schedule of τ, i.e., $\tau_t = (\tau_T/\tau_0)^t$, where τ_0 and τ_T are set to 5.0 and 2.0, respectively. For node classification task the l-hop subgraph of the target node is extracted, where l denotes the number of graph convolution layers. For the hyper-parameter that controls the *mask size* regularization, we select [0.01, 0.01, 0.01, 0.05, 0.05] for BA-Shapes, Tree-Cycles, Tree-Grid, Mutagenicity, and MSRC_21, respectively. For the hyper-parameter that controls the *entropy* regularization, we set it to 1.0 for all datasets.

PGM-Explainer. A Probabilistic Graphical Model (PGM) is utilized as a surrogate in PGM-Explainer to generate explanations for GNN. The goal of PGM-Explainer is to learn a Bayesian network upon the local perturbation-prediction dataset to identify significant nodes. Specifically, to obtain the local perturbation-prediction dataset, PGM-Explainer randomly perturbs the node features of several random nodes, then records a random variable indicating whether its features are perturbed and its influence on the GNN predictions. After obtaining the local dataset, the Grow-Shrink (GS) algorithm is leveraged to reduce the size of the local dataset, then an interpretable Bayesian network is employed to fit the local dataset and to explain the GNN model. For hyper-parameters settings, we set the probability of perturbation to 0.5, the prediction difference threshold to 0.1, the number of samples to 50, and the p-value for the conditional independence test to 0.05.

CF-GNNExplainer. Different from GNNExplainer and PGExplainer whose objective is to generate subgraphs that are especially relevant for a particular prediction, the goal of CF-GNNExplainer is to generate a counterfactual explanation that can flip the model prediction. The nature of counterfactual explanation [29] is that the explanation can flip the model prediction subject to a minimal perturbation on data, which is equivalent to the nature of the necessary explanation. To generate the counterfactual explanation G_i', the objective of CF-GNNExplainer is given as follows.

$$\max_{G_i'} \mathcal{L}_{pred}(G_i, G_i') + \beta \mathcal{L}_{dist}(G_i, G_i'),$$

Table C.5: The hyperparameter setting and test accuracy of GNN.

	BA-Shapes	Tree-Cycles	Tree-Grid	BA2Motif	Mutagenicity	MSRC_21
lr	0.001	0.001	0.001	0.01	0.001	0.001
# of GNN layers	3	3	3	3	3	3
# of epochs	2000	2000	2000	2000	500	500
dropout	0	0	0	0	0.5	0.5
optimizer	Adam	Adam	Adam	Adam	Adam	Adam
weight decay	0	0	0	0	5e-4	5e-4
accuracy	0.957	0.903	0.927	1.0	0.761	0.983

where $\mathcal{L}_{pred}(G_i, G_i')$ stands for the prediction loss aiming to obtain G_i' that can flip the model prediction, and $\mathcal{L}_{dist}(G_i, G_i')$ stands for the distance loss aiming to achieve minimum perturbation. In implementations, we adopt the original settings where for node classification task the $(l+1)$-hop subgraph of the target node is extracted, where l denotes the number of graph convolution layers. For β we select [0.05, 0.05, 0.05, 0.01, 0.01] for BA-Shapes, Tree-Cycles, Tree-Grid, Mutagenicity, and MSRC_21, respectively.

CF^2. CF^2 produces a trade-off explanation between necessity and sufficiency by taking insights into counterfactual and factual reasoning from causal inference theory. The objective of CF^2 is to minimize the complexity of the explanation, subject to the explanation strength being strong enough. The explanation strength can be divided into two parts, the counterfactual explanation strength S_c and the factual explanation strength S_f, and a hyperparameter is introduced to control the trade-off between the two. Formally, the objective of CF^2 is shown as follows.

$$\min_{G_i', X_i'} C(G_i') \quad s.t. \quad \alpha S_f(G_i', X_i') + (1-\alpha)S_c(G_i', X_i') > \lambda,$$

where $C(G_i', X_i')$ measures the complexity of the explanation G_i', and X_i' is similar to the X_i' of GNNExplainer as we mentioned before. For implementations, we adopt the original settings where α is set to 0.6 for all datasets, and λ is set to [500, 500, 500, 1000, 1000] for BA-Shapes, Tree-Cycles, Tree-Grid, Mutagenicity, and MSRC_21, respectively. Also for node classification task the l-hop subgraph of the target node is extracted, where l denotes the number of graph convolution layers.

Appendix C.3. Hardware and Software

For hardware, our experiments are conducted on on a Linux machine with Nvidia GeForce RTX 2080 Ti with 11 GB memory. For software, we implement our NSEG in Deep Graph Library (DGL) with Pytorch. For the implementations of the baseline approaches, we follow and modify the following codes: GNNExplainer[1], PGExplainer[2], PGM-Explainer[3], CF-GNNExplainer[4], and CF^2 [5].

---

[1] https://github.com/RexYing/gnn-model-explainer
[2] https://github.com/flyingdoog/PGExplainer
[3] https://github.com/mims-harvard/GraphXAI/tree/main/graphxai/explainers/pgm_explainer
[4] https://github.com/a-lucic/cf-gnnexplainer
[5] https://github.com/chrisjtan/gnn_cff

Appendix D. More Experimental Results

Appendix D.1. Result on GIN Architecture

In this subsection, we showcase the additional experimental results on GIN [55] architecture besides the GCN architects, where the results of $Fid+^c$, $Fid\text{-}^c$ and $charact^c$ are given in Table D.6, and the results of $Recall@K$ and $ROC\text{-}AUC$ are given in Table D.7. We observe that our NSEG achieves the best $Fid+^c$, $Fid\text{-}^c$ and $charact^c$ compared with other approaches except for $Fid\text{-}^c$ and $charact^c$ on Tree-Grid dataset (NSEG also achieve the second best). Besides, our NSEG achieves the best $Recall@K$ on BA-Shapes and Tree-Grid, while achieving the second winner on the remaining metrics.

Appendix D.2. More Qualitative Studies

We further illustrate the necessary and sufficient explanation obtained by our approach through visualization in more case studies. The visualization of explanations of Mutagenicity is shown in Figure D.11, while the visualization of explanations of MSRC_21 is shown in Figure D.12.

Table D.6: Comparison of $Fid+^c$(%), $Fid\text{-}^c$ (%) and $charact^c$ (%) of the explanations on GIN architecture. Mean and standard deviation are reported.

	Node Classification								
	BA-Shapes			Tree-Cycles			Tree-Grid		
	Fid+^{c} (↑)	Fid-^{c} (↓)	charact^c (↑)	Fid+^{c} (↑)	Fid-^{c} (↓)	charact^c (↑)	Fid+^{c} (↑)	Fid-^{c} (↓)	charact^c (↑)
Random	62.50±0.00	62.45±0.10	46.91±0.08	45.56±2.76	47.44±3.17	48.80±2.92	98.75±0.00	**98.75±0.00**	**2.47±0.00**
GuidedBP	61.75±0.00	62.50±0.00	46.66±0.00	61.39±0.00	63.61±0.00	45.69±0.00	98.75±0.00	**98.75±0.00**	**2.47±0.00**
GNNExplainer	62.00±0.00	62.00±0.00	47.12±0.00	96.48±0.69	9.07±1.25	93.62±0.98	98.75±0.00	**98.75±0.00**	**2.47±0.00**
PGExplainer	48.50±10.49	62.50±0.00	41.78±4.28	44.07±26.47	42.96±29.19	49.60±27.96	95.51±2.67	**98.75±0.00**	**2.47±0.00**
CFGNNExplainer	**62.75±0.50**	62.50±0.00	46.94±0.14	98.33±0.00	81.81±0.14	30.71±0.20	98.47±0.28	**98.75±0.00**	**2.47±0.00**
CF^2	62.50±0.00	**58.00±0.00**	**50.24±0.00**	100.00±0.00	0.00±0.00	100.00±0.00	98.75±0.00	**98.75±0.00**	**2.47±0.00**
NSEG(PNS^e)	62.50±0.00	**58.00±0.00**	**50.24±0.00**	100.00±0.00	0.00±0.00	100.00±0.00	100.00±0.00	98.75±0.00	2.47±0.00
	Graph Classification								
	BA2 Motif			Mutagenicity			MSRC_21		
	Fid+^{c} (↑)	Fid-^{c} (↓)	charact^c (↑)	Fid+^{c} (↑)	Fid-^{c} (↓)	charact^c (↑)	Fid+^{c} (↑)	Fid-^{c} (↓)	charact^c (↑)
Random	**49.50±0.00**	49.50±0.00	49.99±0.00	55.84±1.38	54.96±1.49	49.85±1.20	2.88±0.63	93.04±0.98	3.98±0.52
GuidedBP	**49.50±0.00**	49.50±0.00	49.99±0.00	66.80±0.00	29.60±0.00	68.55±0.00	39.2±0.00	34.00±0.00	49.19±0.00
GNNExplainer	43.67±0.24	49.50±0.00	46.84±0.14	92.67±0.50	0.53±0.19	95.95±0.27	59.47±0.69	2.80±0.45	73.79±0.59
PGExplainer	**49.50±0.00**	49.50±0.00	49.99±0.00	61.47±0.38	58.27±0.38	49.71±0.35	38.27±10.01	24.40±4.73	50.56±9.67
CFGNNExplainer	48.00±1.08	49.50±0.00	49.21±0.57	66.40±1.99	14.53±0.19	74.72±1.28	15.73±0.69	97.20±0.00	4.75±0.03
CF^2	**49.50±0.00**	49.50±0.00	49.99±0.00	100.00±0.00	0.00±0.00	100.00±0.00	87.60±0.00	1.20±0.00	92.86±0.00
NSEG(PNS^e)	**49.50±0.00**	**23.50±0.71**	**60.11±0.22**	96.13±0.19	0.40±0.00	97.84±0.10	**90.40±0.00**	**0.40±0.00**	**94.78±0.00**

Table D.7: Comparison of $Recall@K$ (%) and $ROC\text{-}AUC$ (%) of the explanations on GIN architecture. Mean and standard deviation are reported.

|	BA-Shapes		Tree-Cycles		Tree-Grid		BA2Motif		Mutagenicity	
	Recall@K	ROC-AUC	Recall@K	ROC-AUC	Recall@K	ROC-AUC	Recall@K	ROC-AUC	Recall@K	ROC-AUC
Random	33.33±0.38	50.09±0.29	66.94±0.31	50.82±0.38	64.04±0.26	49.82±0.21	22.22±1.23	50.46±1.00	40.20±0.87	49.71±0.85
GuidedBP	77.77±0.00	98.98±0.00	76.85±0.00	73.04±0.00	74.83±0.00	71.31±0.00	36.90±0.00	73.28±0.00	76.67±0.00	80.07±0.00
GNNExplainer	62.58±0.00	74.89±0.00	67.40±0.24	52.38±1.09	73.40±0.11	65.24±0.02	20.15±0.42	51.59±0.09	35.56±0.65	53.64±0.07
PGExplainer	82.54±8.43	99.14±0.47	78.53±5.87	75.68±5.39	67.20±1.21	59.48±7.59	24.63±10.33	63.66±5.49	40.84±24.44	50.06±25.12
CFGNNExplainer	68.59±0.24	62.57±0.11	70.98±0.24	64.91±0.11	69.82±0.18	62.69±0.17	27.50±0.10	57.31±0.02	76.66±0.20	78.54±0.07
CF^2	61.67±0.10	65.76±0.09	74.68±0.25	72.76±0.00	64.26±0.07	54.49±0.04	18.86±0.13	55.84±0.03	53.70±0.28	62.80±0.29
NSEG(PNS^e)	65.43±0.05	79.63±0.03	78.35±0.07	73.31±0.05	65.23±0.06	58.71±0.04	37.35±0.73	70.74±0.51	78.13±0.30	74.82±0.19
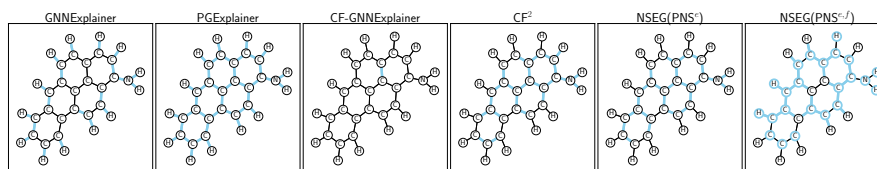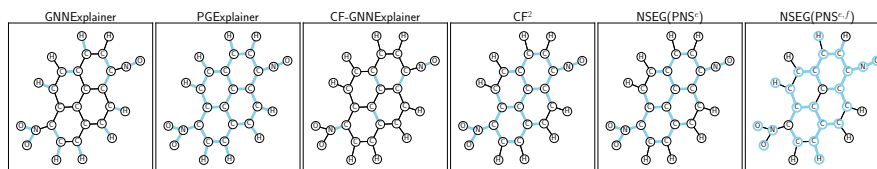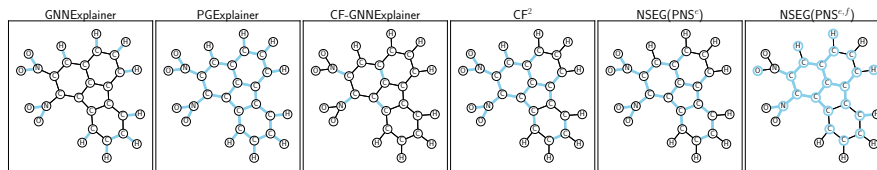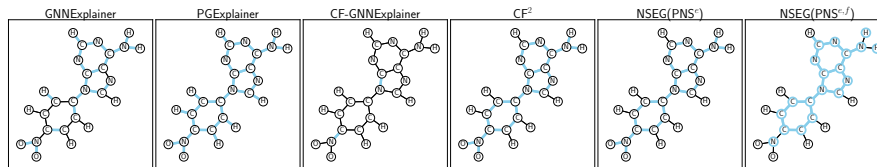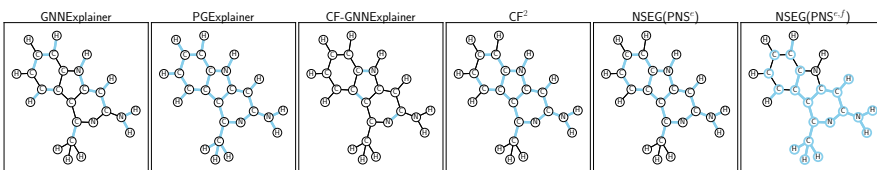
Figure D.11: Explanations of GNNExplainer, PGExplainer, CF-GNNExplainer, CF^2, NSEG(PNS^e), and NSEG(PNS^{e,f}) on various Mutagenicity instances obtained by threshold, where the explanations are highlighted in blue.
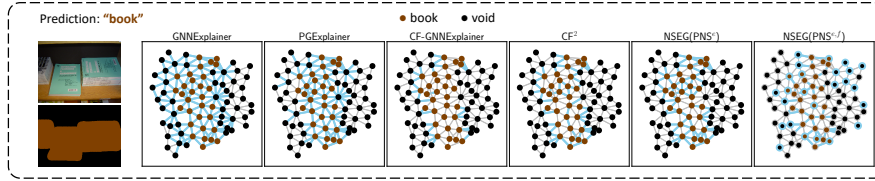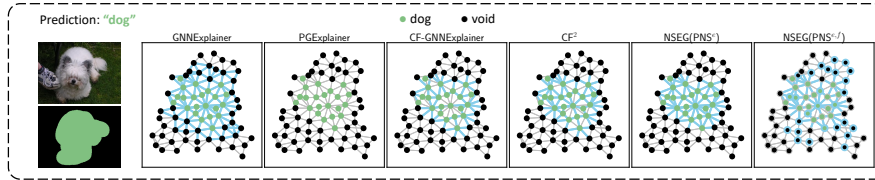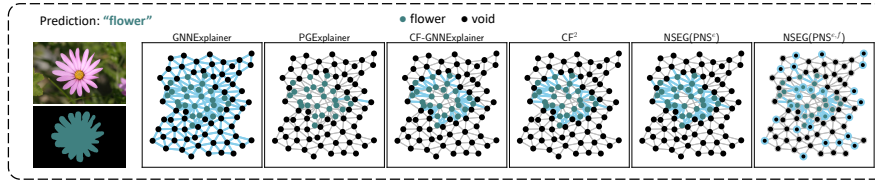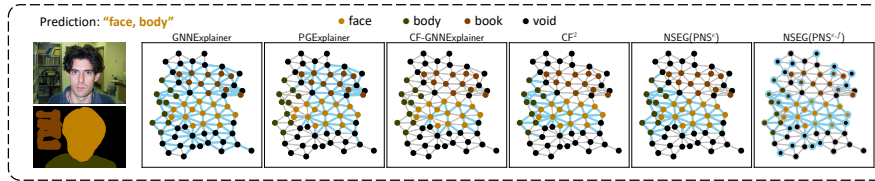
Figure D.12: Explanations of GNNExplainer, PGExplainer, CF-GNNExplainer, CF^2, NSEG(PNS^e), and NSEG($PNS^{e,f}$) on various MSRC_21 instances obtained by threshold, where the explanations are highlighted in blue.