Factorized Explainer for Graph Neural Networks

Rundong Huang[1], **Farhad Shirani**[2*], **Dongsheng Luo**[2*]

[1]Technical University of Munich, Munich, Germany
[2]Florida International University, Miami, U.S.
rundong.huang@tum.de, {fshirani,dluo}@fiu.edu

Abstract

Graph Neural Networks (GNNs) have received increasing attention due to their ability to learn from graph-structured data. To open the black-box of these deep learning models, post-hoc instance-level explanation methods have been proposed to understand GNN predictions. These methods seek to discover substructures that explain the prediction behavior of a trained GNN. In this paper, we show analytically that for a large class of explanation tasks, conventional approaches, which are based on the principle of graph information bottleneck (GIB), admit trivial solutions that do not align with the notion of explainability. Instead, we argue that a modified GIB principle may be used to avoid the aforementioned trivial solutions. We further introduce a novel factorized explanation model with theoretical performance guarantees. The modified GIB is used to analyze the structural properties of the proposed factorized explainer. We conduct extensive experiments on both synthetic and real-world datasets to validate the effectiveness of our proposed factorized explainer.

Introduction

Graph-structured data is ubiquitous in real-world applications, manifesting in various domains such as social networks (Fan et al. 2019), molecular structures (Mansimov et al. 2019; Chereda et al. 2019), and knowledge graphs (Liu et al. 2022). This has led to significant interest in learning methodologies specific to graphical data, particularly, graph neural networks (GNNs). GNNs commonly employ message-passing mechanisms, recursively transmitting and fusing messages among neighboring nodes on graphs. Thus, the learned node representation captures both node attributes and neighborhood information, thereby enabling diverse downstream tasks such as node classification (Kipf and Welling 2017; Veličković et al. 2018), graph classification (Xu et al. 2019), and link prediction (Lu et al. 2022).

Despite the success of GNNs in a wide range of domains, their inherent "black-box" nature and lack of interpretability, a characteristic shared among many contemporary machine learning methods, is a major roadblock in their utility in sensitive application scenarios such as autonomous decision systems. To address this, various GNN explanation methods have been proposed to understand the graph-structured data and associated deep graph learning models (Ying et al. 2019; Luo et al. 2020; Yuan et al. 2022). In particular, post-hoc instance-level explanation methods provide an effective way to identify determinant substructures in the input graph, which plays a vital role in trustworthy deployments (Ying et al. 2019; Luo et al. 2020). In the context of graph classification, the objective of graph explanation methods is, given a graph G, to extract a *minimal* and *sufficient* subgraph, G^*, that can be used to determine the instance label, Y. The Graph Information Bottleneck principle (GIB) (Wu et al. 2020) provides an intuitive principle that is widely adopted as a practical instantiation. At a high level, the GIB principle finds the subgraph G^* which minimizes the mutual information between the original graph G and the subgraph G^* and maximizes the mutual information between the subgraph G^* and instance label Y by minimizing $I(G, G^*) - \alpha I(G^*, Y)$, where the hyperparameter $\alpha > 0$ captures the tradeoff between minimality and informativeness of G^* (Miao, Liu, and Li 2022). As an example, the GNNExplainer method operates by finding a learnable edge mask matrix, which is optimized by the GIB objective (Ying et al. 2019). The PGExplainer also uses a GIB-based objective and incorporates a parametric generator to learn explanatory subgraphs from the model's output (Luo et al. 2020).

There are several limitations in the existing explainability approaches. First, as shown analytically in this work, existing GIB-based methods suffer from perceptually unrealistic explanations. Specifically, we show that in a wide-range of statistical scenarios, the original GIB formulation of the explainability problem has a trivial solution where the achieved explanation G^* *signals* the predicted value of Y, but is independent of the input graph G, otherwise. That is, the Markov chain $G^* \leftrightarrow Y \leftrightarrow G$ holds. As a result, the explanation G^* optimizing the GIB objective may consist of a few disconnected edges and fails to align with the high-level notion of explainability. To alleviate this problem, PGExplainer includes an ad-hoc connectivity constraint as the regularization term (Luo et al. 2020). However, without theoretical guarantees, the effectiveness of the extra regularization is marginal in more complicated datasets (Shan et al. 2021). Second, although previous parametric explanation methods, such as PGExplainer (Luo et al. 2020) and ReFine (Wang et al. 2021),

[*]Corresponding authors.

are efficient in the inductive setting, these methods neglect the existence of multiple motifs, which is routinely observed in real-life datasets. For example, In the MUTAG dataset (Debnath et al. 1991), both chemical components NO_2 and NH_2, which can be considered as explanation subgraphs, contribute to the positive mutagenicity. Existing methods over-simplify the relationship between motifs and labels to one-to-one, leading to inaccuracy in real-life applications.

To address these issues, we first analytically investigate the pitfalls of the application of the GIB principle in explanation tasks from an information theoretic perspective, and propose a modified GIB principle that avoids these issues. To further improve the inductive performance, we propose a new framework to unify existing parametric methods and show that their suboptimality is caused by their locality property and the lossy aggregation step in GNNs. We further propose a straightforward and effective factorization-based explanation method to break the limitation of existing local explanation functions. We summarize our main contributions as follows.

- For the first time, we point out that the gap between the practical objective function (GIB) and high-level objective is non-negligible in the most popular post-hoc explanation framework for graph neural networks.
- We derive a generalized framework to unify existing parametric explanation methods and theoretically analyze their pitfalls in achieving accurate explanations in complicated real-life datasets. We further propose a straightforward explanation method with a solid theoretical foundation to achieve better generalization capacity.
- Comprehensive empirical studies on both synthetic and real-life datasets demonstrate that our method can consistently improve the quality of the explanations.

Preliminary

Notations and Problem Definition

A graph G is parameterized by a quadruple $(\mathcal{V}, \mathcal{E}; \mathbf{Z}, \mathbf{A})$, where i) $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ is the node/vertex[1] set, ii) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, iii) $\mathbf{Z} \in \mathbb{R}^{n\times d}$ is the feature matrix, where the ith row of $\mathbf{Z}$, denoted by $\mathbf{z}_i \in \mathbb{R}^{1\times d}$, is the d-dimensional feature vector associated with node $v_i, i \in [n]$, and iv) the adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$ is determined by the edge set $\mathcal{E}$, where $A_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, $A_{ij} = 0$, otherwise. We write $|G|$ and $|\mathcal{E}|$ interchangeably to denote the number of edges of G.

For graph classification task, each graph G_i has a label $Y_i \in \mathcal{C}$, with a GNN model f trained to classify G_i into its class, i.e., $f : G \mapsto \{1, 2, \cdots, |\mathcal{C}|\}$. For the node classification task, each graph G_i denotes a K-hop sub-graph centered around node v_i, with a GNN model f trained to predict the label for node v_i based on the node representation of v_i learned from G_i.

Problem 1 (Post-hoc Instance-level GNN Explanation (Yuan et al. 2022; Luo et al. 2020; Ying et al. 2019))**.** *Given a trained GNN model f, for an arbitrary input graph $G = (\mathcal{V}, \mathcal{E}; \mathbf{Z}, \mathbf{A})$, the goal of post-hoc instance-level GNN explanation is to find a subgraph $G^* = \Psi(G)$ that can 'explain'[2] the prediction of f on G. The mapping $\Psi : G \mapsto G^*$ is called the explanation function.*

Informative feature selection has been well studied in non-graph structured data (Li et al. 2017), and traditional methods, such as concrete autoencoder (Abid, Balin, and Zou 2019), can be directly extended to explain features in GNNs. In this paper, we focus on discovering important typologies. Formally, the obtained explanation G^* is characterized by a binary mask $\mathbf{M} \in \{0,1\}^{n\times n}$ on the adjacency matrix, e.g., $G^* = (\mathcal{V}, \mathcal{E}, \mathbf{A} \odot \mathbf{M}; \mathbf{Z})$, where $\odot$ is elements-wise multiplication. The mask highlights components of G which influence the output of f.

Graph Information Bottleneck

The GIB principle refers to the graphical version of the Information Bottleneck (IB) principle (Tishby and Zaslavsky 2015) which offers an intuitive measure for learning dense representations. It is based on the notion that an optimal representation should contain *minimal* and *sufficient* information for the downstream prediction task. Recently, a high-level unification of several existing post hoc GNN explanation methods, such as GNNExplainer (Ying et al. 2019), and PG-Explainer (Luo et al. 2020) was provided using this concept (Wu et al. 2020; Miao, Liu, and Li 2022; Yu et al. 2021). Formally, prior works have represented the objective of finding an explanation graph G^* in G as follows:

$$G^* \triangleq \underset{P_{G'|G}:\mathbb{E}(|G'|)\leq\gamma}{\arg\min} \quad I(G, G') - \alpha I(G', Y), \tag{1}$$

where G^* is the explanation subgraph, $\gamma \in \mathbb{N}$ is the maximum expected size (number of edges) of the explanation, Y is the original or ground truth label, and α is a hyper-parameter capturing the trade-off between minimality and sufficiency constraints. At a high level, the GIB formulation given in equation 1 selects the minimal explanation G', by minimizing $I(G, G')$ and imposing $\mathbb{E}(|G'|) \leq \gamma$, that inherits only the most indicative information from G to predict the label Y, by maximizing $I(G', Y)$, while avoiding imposing potentially biased constraints, such the connectivity of the selected subgraphs and exact maximum size constraints (Miao, Liu, and Li 2022). Note that from the definition of mutual information, we have $I(G', Y) = H(Y) - H(Y|G')$, where the entropy $H(Y)$ is static and independent of the explanation process. Thus, minimizing the mutual information between the explanation subgraph G' and Y can be reformulated as maximizing the conditional entropy of Y given G'. That is:

$$G^* = \underset{P_{G'|G}:\mathbb{E}(|G'|)\leq\gamma}{\arg\min} \quad I(G, G') + \alpha H(Y|G'). \tag{2}$$

Graph Information Bottleneck for Explanation

In this section, we study several pitfalls arising from the application of the GIB principle to explanation tasks. We

[1] We use node and vertex interchangeably.

[2] The notion of explainability is made precise in the sequel, where the modified GIB principle is introduced.

demonstrate that, for a broad range of learning tasks, the original GIB formulation of the explainability problem has a trivial solution that does not align with the intuitive notion of explainability. We propose a modified version of the GIB principle that avoids this trivial solution and is applicable in constructing GNN explanation methods. The analytical derivations in subsequent sections will focus on this modified GIB principle. To elaborate, we argue that the optimization given in equation 2 is prone to *signaling issues* and, in general, does not fully align theoretically with the notion of explainability. More precisely, the GIB formulation allows for an explanation algorithm to output G^* which *signals* the predicted value of Y, but is independent of the input graph G otherwise. To state this more concretely, we consider the class of statistically degraded classification tasks defined in the following.

Definition 1 (Statistically Degraded Classification). *Consider a classification task characterized by the triple $(\mathcal{X}, \mathcal{Y}, P_{\mathbf{X},Y})$, where $\mathcal{X}$ represents the feature space, $\mathcal{Y}$ denotes the set of output labels, and $P_{\mathbf{X},Y}$ characterizes the joint distribution of features and labels. The classification task is called statistically degraded*[3] *if there exists a function $h : \mathcal{X} \to \mathcal{Y}$ such that the Markov chain $\mathbf{X} \leftrightarrow h(\mathbf{X}) \leftrightarrow Y$ holds. That is, $h(\mathbf{X})$ is a sufficient statistic for $\mathbf{X}$ w.r.t. Y.*

Remark 1. *Any deterministic classification task, where there exists a function $h : \mathcal{X} \to \mathcal{Y}$ such that $h(\mathbf{X}) = Y$, is statistically degraded.*

Remark 2. *There are classification tasks that are not statistically degraded. For instance, let us consider a classification task in which the feature vector is $\mathbf{X} = (X_1, X_2)$, where X_1 and X_2 are independent binary symmetric variables. Let the label Y be equal to X_1 with probability $p \in (0, 1)$ and equal to X_2, otherwise. By exhaustively searching over all 16 possible choices of $h(\mathbf{X})$, it can be verified that no Boolean function $h(\mathbf{X})$ exists such that the relationship $\mathbf{X} \leftrightarrow h(\mathbf{X}) \leftrightarrow Y$ holds. Consequently, the classification task $(\mathcal{X}, \mathcal{Y}, P_{\mathbf{X},Y})$ is not statistically degraded.*

Remark 3. *Note that for the statistically degraded task defined in Definition 1, the optimal classifier $f^*(\mathbf{X})$ is equal to the sufficient statistic $h(\mathbf{X})$.*

In order to show the limitations of the GIB in fully encapsulating the concept of explainability, in the sequel we focus on statistically degraded classification tasks involving graph inputs. That is, we take $\mathbf{X} = G$, where G is the input graph. The next lemma shows that, for any statistically degraded task, there exists an explanation function $\Psi(\cdot)$ which optimizes the GIB objective function (equation 2), and whose output is independent of G given $h(\cdot)$. That is, although the explanation algorithm is optimal in the GIB sense, it does not provide any additional information about the input of the classifier, in addition to the information that the classifier output label $h(G)$ readily provides.

Theorem 1. *Consider a statistically degraded graph classification task, parametrized by $(P_{G,Y}, h(\cdot))$, where $P_{G,Y}$ is the joint distribution of input graphs and their labels, and $h : \mathcal{G} \to \mathcal{Y}$ is such that $G \leftrightarrow h(G) \leftrightarrow Y$ holds. For any $\alpha > 0$, there exists an explanation algorithm $\Psi_\alpha(\cdot)$ such that $G' \triangleq \Psi_\alpha(G)$ optimizes the objective function in equation 2 and $\Psi_\alpha(G) \leftrightarrow h(G) \leftrightarrow G$ holds.*

The proof relies on the following modified data processing inequality.

Lemma 1 (Modified Data Processing Inequality). *Let A, B and C be random variables satisfying the Markov chain $A \leftrightarrow B \leftrightarrow C$. Define the random variable A' such that $P_{A'|C} = P_{A|C}$ and $A, B \leftrightarrow C \leftrightarrow A'$. Then,*

$$I(A', B) \leq I(A, B).$$

The proof of Lemma 1 and Theorem 1 are included in Appendix.

As shown by Theorem 1, the original GIB formulation does not fully align with the notion of explainability. Consequently, we adopt the following modified objective function:

$$G^* \triangleq \underset{P_{G'|G}:\mathbb{E}(|G'|)\leq\gamma}{\arg\min} \quad I(G, G') + \alpha CE(Y, Y'), \qquad (3)$$

where $Y' \triangleq f(G')$ is the predicted label of G' made by the model to be explained f, and the cross-entropy $\mathrm{CE}(Y, Y')$ between the ground truth label Y and Y' is used in place of $H(Y|G')$ in the original GIB. The modified GIB avoids the signaling issues in Theorem 1, by comparing the correct label Y with the prediction output Y' based on the original model $f(\cdot)$. This is in contrast with the original GIB principle which measures the mutual information $I(Y, G')$, which provides a general measure of how well Y can be predicted from G' (via Fano's inequality (El Gamal and Kim 2010)), without relating this prediction to the original model $f(\cdot)$. It should be mentioned that several recent works have also adopted this modified GIB formulation (Ying et al. 2019; Luo et al. 2020). However, the rationale provided in these earlier studies was that the modified GIB serves as a computationally efficient approximation for the original GIB, rather than addressing the limitations of the original GIB shown in Theorem 1.

K-FactExplainer for Graph Neural Networks

In this section, we first theoretically show that existing parametric explainers based on the GIB objective, such as PGExplainer (Luo et al. 2020), are subject to two sources of inaccuracies: locality and lossy aggregation. Then we propose a straightforward and effective approach to mitigating the problem. In the subsequent sections, we provide simulation results that corroborate these theoretical predictions.

Theoretical Analysis

We first define the general class of *local explanation methods*.

Definition 2 (Geodisc Restricted Graph). *Given a graph $G = (\mathcal{V}, \mathcal{E}; \mathbf{Z}, \mathbf{A})$, node $v \in \mathcal{V}$, and a radius $r \in \mathbb{N}$, the (v, r)-restriction of G is the graph $G_{v,r} = (\mathcal{V}_{v,r}, \mathcal{E}_{v,r}; \mathbf{Z}_{v,r}, \mathbf{A}_{v,r})$, where*

- *$\mathcal{V}_{v,r} \triangleq \{v' | d(v, v') \leq r\}$, where $d(\cdot, \cdot)$ is the geodisc distance.*

[3] Statistical degradedness has its origins in the field of information theory, particularly in communication and estimation applications (El Gamal and Kim 2010)

- $\mathcal{E}_{v,r} \triangleq \{(v_i, v_j)|e \in \mathcal{E}, v_i, v_j \in \mathcal{V}_{v,r}\}$.
- $\mathbf{Z}_{v,r}$ *consists of feature vectors in* $\mathbf{Z}$ *corresponding to* $v \in \mathcal{V}_{v,r}$.
- $\mathbf{A}_{v,r}$ *is the adjacency matrix corresponding to* $\mathcal{E}_{v,r}$.

Definition 3 (Local Explanation Methods). *Consider a graph classification task* $(\mathcal{G}, \mathcal{Y}, P_{G,Y})$*, a classification function* $f : \mathcal{G} \to \mathcal{Y}$*, a parameter* $r \in \mathbb{N}$*, and an explanation function* $\Psi : \mathcal{G} \to \mathcal{G}$*, where* $\mathcal{G}$ *is the set of all possible input graphs, and* $\mathcal{Y}$ *is the set of output labels. Let* $G' = \Psi(G) = (\mathcal{V}', \mathcal{E}'; \mathbf{Z}', \mathbf{A}')$*. The explanation function* $\Psi(\cdot)$ *is called an r-local explanation function if:*

1. *The Markov chain* $\mathbb{1}(v \in \mathcal{V}') \leftrightarrow G_{v,r} \leftrightarrow G$ *holds for all* $v \in \mathcal{V}$*, where* $\mathbb{1}(\cdot)$ *is the indicator function.*
2. *The edge* (v, v') *is in* $\mathcal{E}'$ *if and only if* $v, v' \in \mathcal{V}'$ *and* $e \in \mathcal{E}$*.*

The first condition in Definition 3 requires that the presence of each vertex v in the explanation G' only depends on its neighboring vertices in G which are within its r local neighborhood. The second condition requires that G' be a subgraph of G. It is straightforward to show that various explanation methods such as PGExplainer are local explanation methods due to the boundedness of their corresponding computation graphs. This is formalized in the following proposition.

Proposition 1 (Locality of PGExplainer). *Consider a graph classification task* $(\mathcal{G}, \mathcal{Y}, P_{G,Y})$ *and an* ℓ *layer GNN classifier* $f(\cdot)$*, for some* $\ell \in \mathbb{N}$*. Then, any explanation* $\Psi(\cdot)$ *for* $f(\cdot)$ *produced using the PGExplainer is an* ℓ*-local explanation function.*

Next, we argue that local explanation methods cannot be optimal in the modified GIB sense for various classification tasks. Furthermore, we argue that this issue may be mitigated by the addition of a hyperparameter k as described in subsequent sections in the context of the K-FactExplainer.

To provide concrete analytical arguments, we focus on a specific graph classification task, where the class labels are binary, the input graph has binary-valued edges, and the output label is a function of a set of indicator motifs. To elaborate, we assume that the label to be predicted is $Y = \max\{E_1, E_2, \cdots, E_s\}$, where $E_i, i \in [s]$ are Bernoulli variables, and if $E_i = 1$, then $g_i \subseteq G$ for some fixed subgraphs $g_i, i \in [s]$. In the explainability literature, each of the subgraphs $g_i, i \in [s]$ is called a motif for label $Y = 1$. Let us define $G_e = \bigcup_{i\in[s]} g_i \mathbb{1}(E_i = 1)$. So that G_e is the union of all the edges in the motifs that are present in G, and it is empty if $Y = 0$. Formally, the classification task under consideration is characterized by the following joint distribution:

$$\begin{aligned} &P_{G,Y}(g, y) \\ &= \sum_{e^s, g_0} P_{E^s}(e^s) P_{G_0}(g_0) \mathbb{1}(y = \max_{i\in[s]} e_i, g = g_0 \cup g_e), \end{aligned} \quad (4)$$

where $e^s \in \{0, 1\}^s$, $g_e \triangleq \bigcup_{i\in[s]} g_i \mathbb{1}(e_i = 1)$, and G_0 is the "irrelevant" edges in G with respect to the label Y.

Remark 4. *The graph classification task on the MUTAG dataset is an instance of the above classification scenario, where there are two motifs, corresponding to the existence of* NH_2 *and* NO_2 *chemical groups, respectively (Ying et al. 2018). Similarly, the BA-4Motif classification task considered in the Appendix can be posed in the form of equation 4.*

In graph classification tasks characterized by equation 4, if the label of G is one, then at least one of the motifs is present in G. Note that the reverse may not be true as the motifs may randomly appear in the 'irrelevant' graph G_0 due to its probabilistic nature. A natural choice for the explanation function $\Psi(\cdot)$ of a classifier $f(\cdot)$ for this task is one which outputs one of the motifs present in G if $f(G) = 1$. For instance, in the MUTAG classification task, an explainer should output NH_2 or NO_2 subgraphs if the output label is equal to one. In the following, we argue that, in classification tasks involving more than one motif local explanation methods cannot produce the motifs accurately. Hence their output does not align with the natural explanation outlined above and is not optimal in the modified GIB sense. To make the result concrete, we further make the following simplifying assumptions:
i) The graph G_0 is Erdös-Rényi with parameter $p \in (0, \frac{1}{2})$:

$$P_{G_0}(g_0) = p^{|g_0|}(1-p)^{\frac{n(n-1)}{2} - |g_0|}.$$

ii) There exists $r, r' > 0$ such that the geodisc radius and geodisc diameter of g_i are less than or equal to r and r', respectively, for all $i \in [s]$.
iii) The geodisc distance between g_i and g_j is greater than r for all $i \neq j$.
iv) $E_i, i \in [s]$ are jointly independent Bernoulli variables with parameter p_i, where $P_{G_0}(g_i) \leq p_i$.

Theorem 2 (Suboptimality of Local Explanation Functions). *Let* $r, r' \in \mathbb{N}$*. For the graph classification task described in equation 4, the following hold:*
a) The optimal Bayes classification rule $f^*(g)$ *is equal to* $\mathbb{1}(\exists i \in [s] : g_i \subseteq g)$*.*
b) For any r-local explanation function, there exists $\alpha' > 0$ *such that the explanation is suboptimal for* f^* *in the modified GIB sense for all* $\alpha > \alpha'$ *and* γ *equal to maximum number of edges of* $g_i, i \in [s]$*.*
c) There exists an integer $k \leq s$*, a parameter* $\alpha' > 0$*, a collection of* r'*-local explanation functions* $\Psi_i(\cdot), i \in [k]$*, and an explanation function* Ψ^**, such that for all inputs g, we have* $\Psi(g) \in \{\Psi_1(g), \Psi_2(g), \cdots, \Psi_k(g)\}$ *and* Ψ^* *is optimal in the modified GIB sense for all* $\alpha > \alpha'$ *and* γ *equal to maximum number of edges of* $g_i, i \in [s]$*.*

The proof of Theorem 2 is provided in the Appendix.

Theorem 2 can be interpreted as follows: for graph classification tasks with more than one motif, although local explanation methods are not optimal in general, one can "patch" together several local explanation methods $\Psi_1(\cdot), \Psi_2(\cdot), \cdots, \Psi_k(\cdot)$ into an explanation method $\Psi^*(\cdot)$, such that i) for any given input g, the output of $\Psi^*(g)$ is equal to the output of one of the explanation functions $\Psi_1(g), \Psi_2(g), \cdots, \Psi_k(g)$, and ii) $\Psi^*(\cdot)$ is optimal in the modified GIB sense. This insight motivates the K-FactExplainer method introduced in the following section.

Remark 5. *Theorem 2 implies that local explanation methods are not optimal in multi-motif classification tasks. It*

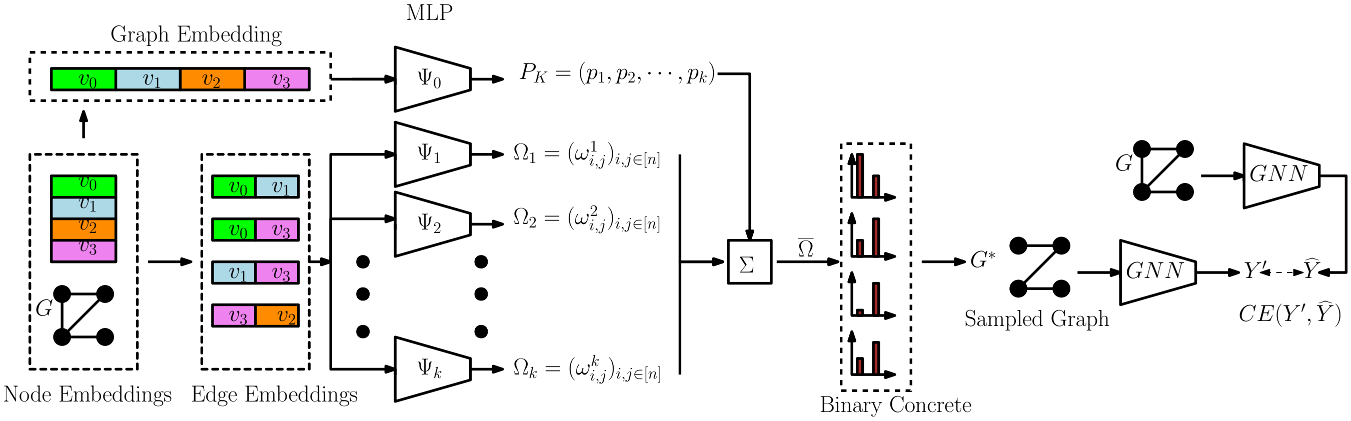

Figure 1: Illustration of K-FactExplainer method. Starting from the left, the node embeddings for graph G are produced using the original GNN. The edge embeddings and graph embedding are produced by concatenating the node embeddings. The MLP Ψ_0 assigns weight to the outputs of PGExplainer MLPs $\Psi_t, t \in [k]$. The resulting vector of edge probabilities $\overline{\Omega} \triangleq (\sum_{t=1}^{k} p_t \omega^t_{i,j})_{i,j\in[n]}$ is used to produce the sampled explanation graph G^*. The explanation is fed to the original GNN and the output label is compared with the original prediction. The training proceeds by minimizing the cross-entropy term $CE(Y', \widehat{Y})$, where $\widehat{Y}$ is GNN prediction for the original input graph G.

should be noted that even in single-motif tasks, post-hoc methods which rely on GNN generated node embeddings for explanations would perform suboptimally. The reason is that the aggregator function which is used to generate the embeddings is lossy (is not a one-to-one function) and potentially loses information during the GNN aggregation step. This can also be observed in the simulation results provided in the sequel, where we apply our proposed K-FactExplainer method and show gains compared to the state of the art in both multi-motif and single-motif scenarios.

K-FactExplainer and a Bootstrapping Algorithm

Motivated by the insights gained by the analytical results in the previous section, we propose a new graph explanation method. An overview of the proposed method is shown in Figure 1. To describe the method, let $f(\cdot)$ be the GNN which we wish to explain. Let $\mathbf{Z}_i, i \in [n]$ denote the node embedding for node $v_i, i \in [n]$ produced by $f(\cdot)$. We construct the edge embeddings $\mathbf{Z}_{i,j} = (\mathbf{Z}_i, \mathbf{Z}_j), i, j \in [n]$ and graph embedding $\mathbf{Z} = (\mathbf{Z}_i, i \in [n])$ by concatenating the edge embeddings. Let $k \in \mathbb{N}$ be the upper-bound on the number of necessary local explainers from Theorem 2. We consider k multi-layer neural networks (MLPs) denoted by $\Psi_t, t \in [k]$. Each MLP Ψ_t individually operates in a similar fashion as the MLP used in the PGExplainer method. That is, Ψ_t operates on each edge embedding $(\mathbf{Z}_i, \mathbf{Z}_j)$ individually, and outputs a Bernoulli parameter $\omega^t_{i,j} \in [0, 1]$. The parameter $\omega^t_{i,j} \in [0, 1]$ can be viewed as the probability that the edge (v_i, v_j) is in the sampled explanation graph. Based on the insights provided by Theorem 2, we wish to patch together the outputs of $\Psi_t, t \in [k]$ to overcome the locality issue in explaining GNNs in multi-motif classification tasks. This is achieved by including the additional MLP Ψ_0 which takes the graph embedding $\mathbf{Z}$ as input and outputs the probability distribution P_K on the alphabet $[k]$. At a high level, the MLP Ψ_0 provides a global view of the input graph, whereas each of the $\Psi_t, t \in [k]$ MLPs provide a local perspective of the input graph. The outputs $(\omega^t_{i,j})_{i,j\in[n]}$ of Ψ_t are linearly combined with weights associated with $P_K(t), t \in [k]$ and the resulting vector of Bernoulli probabilities $\overline{\Omega} = (\sum_{t=1}^{k} P_K(t)\omega^t_{i,j})_{i,j\in[n]}$ is used to sample the edges of the input graph G and produce the explanation graph G^*. In the training phase, G^* is fed to $f(\cdot)$ to produce the prediction Y'. Training is performed by minimizing the cross-entropy term $CE(Y', \widehat{Y})$, where $\widehat{Y} = f(G)$ is the label prediction of the GNN given input G. The next proposition provides an algorithm to bound the value of k, which determines the number of MLPs which need to be trained.

Definition 4 (Minimal r-Cover). *Given a random graph $G = (\mathcal{V}, \mathcal{E}; \mathbf{Z}, \mathbf{A})$ and non-negative integer r, the collection $\mathsf{P} = (\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_{|\mathsf{P}|}), \mathcal{P}_j \subseteq \mathcal{V}$ is called an r-cover of G if*
i) Each partition element $\mathcal{P}_j$ has geodisc diameter at most equal to r, and
ii) $P(\mathcal{V}^* \subseteq \cup_{j\in[|\mathsf{P}|]}\mathcal{P}_j) = 1$, where $\mathcal{V}^*$ denotes the set of vertices of G which are not isolated.
The cover is called minimal if r is the smallest integer for which an r-cover of G exists.

Proposition 2 (Bounding the Number of MLPs). *Consider the setup in Theorem 2. The parameter k, the number of r'-local explainers needed to achieve optimal GIB performance, can be upper-bounded by k^* if G_e has a minimal r'-cover with k^* elements.*
Particularly, if the classifier to be explained, $f(\cdot)$, is a GNN with ℓ layers, and ℓ is greater than or equal to the largest geodisc diameter of the motifs $g_i, i \in [s]$, then k can be upper-bounded by s.

The proof of Proposition 2 is provided in the Appendix.

Proposition 2 provides a method to find an upper-bound on k; however, it requires that the motifs be known beforehand,

	BA-Shapes	BA-Community	Tree-Circles	Tree-Grid	BA-2motifs	MUTAG
GRAD	0.882	0.750	0.905	0.667	0.717	0.783
ATT	0.815	0.739	0.824	0.612	0.674	0.765
RGExp.	$0.985_{\pm 0.013}$	$0.919_{\pm 0.017}$	$0.787_{\pm 0.099}$	$\mathbf{0.927}_{\pm 0.032}$	$0.657_{\pm 0.107}$	$0.873_{\pm 0.028}$
DEGREE	$0.993_{\pm 0.005}$	$0.957_{\pm 0.010}$	$\mathbf{0.902}_{\pm 0.022}$	$0.925_{\pm 0.040}$	$0.755_{\pm 0.135}$	$0.773_{\pm 0.029}$
GNNExp.	$0.742_{\pm 0.006}$	$0.708_{\pm 0.004}$	$0.540_{\pm 0.017}$	$0.714_{\pm 0.002}$	$0.499_{\pm 0.004}$	$0.606_{\pm 0.003}$
PGExp.	$0.999_{\pm 0.000}$	$0.825_{\pm 0.040}$	$0.760_{\pm 0.014}$	$0.679_{\pm 0.008}$	$0.566_{\pm 0.004}$	$0.843_{\pm 0.162}$
K-FactExplainer	$\mathbf{1.000}_{\pm 0.000}$	$\mathbf{0.974}_{\pm 0.004}$	$0.779_{\pm 0.004}$	$0.770_{\pm 0.004}$	$\mathbf{0.821}_{\pm 0.005}$	$\mathbf{0.915}_{\pm 0.010}$

Table 1: Explanation faithfulness in terms of AUC-ROC on edges under six datasets. The higher, the better. Our approach achieves consistent improvements over GIB-based explanation methods.

so that G_e is known and the size of its minimal cover can be computed. In practice, we do not know the motifs before the start of the explanation process, since the explanation task would be trivial otherwise. We provide an approximate solution, where instead of finding the minimal cover for G_e, we use a bootstrapping method in which we find the minimal cover for the explanation graphs produced by another pre-trained explainer, e.g., a PGExplainer. To elaborate, It takes the GNN model to be explained f, a set of training input graphs $\mathbb{G}$, and a post-hoc explainer Ψ as input. In our simulations, we adopt PGExplainer as the post-hoc explainer Ψ. Other explanation methods such as GNNExplainer (Ying et al. 2019) can also be used in this step. For each graph $G \in \mathbb{G}$, we first apply the explainer Ψ on G to get the initial explanation graph, whose nodes are listed in $\mathbb{V}_e$ and edge mask matrix is denoted by $\mathbf{M}$. This is used as an estimate for G_e. To find its minimal cover, we rank the nodes in $\mathbb{V}_e$ based on their degrees and initialize $k' = 0$. For each step, we select a node v from $\mathbb{V}_e$ and extract its k^*-hop neighborhood graph, $G_v^{(l)}$. Then, we remove all nodes in $G_v^{(l)}$ from $\mathbb{V}_e$. After that, we add a count to k' and select the next node in $\mathbb{V}_e$ until $|\mathbb{V}_e| = 0$. We iterate all graphs in $\mathbb{G}$ and report the maximum value of k' as $\hat{k}$, the estimate of k. A detailed algorithm can be found in Appendix.

Related Work

Graph neural networks (GNNs) have gained increasing attention in recent years due to the need for analyzing graph data structures (Kipf and Welling 2017; Veličković et al. 2018; Xu et al. 2019; Feng et al. 2020; Satorras, Hoogeboom, and Welling 2021; Bouritsas et al. 2022). In general, GNNs model messages from node representations and then propagate messages with message-passing mechanisms to update representations. GNNs have been successfully applied in various graph mining tasks, such as node classification (Kipf and Welling 2017), link prediction (Zhang and Chen 2018), and graph classification (Xu et al. 2019). Despite their popularity, akin to other deep learning methodologies, GNNs operate as black box models, which means their functioning can be hard to comprehend, even when the message passing techniques and parameters used are known. Furthermore, GNNs stand apart from conventional deep neural networks that assume instances are identically and independently distributed. GNNs instead integrate node features with graph topology, which complicates the interpretability issue.

Recent studies have aimed to interpret GNN models and offer explanations for their predictions (Ying et al. 2019; Luo et al. 2020; Yuan et al. 2020, 2022, 2021; Lin, Lan, and Li 2021; Wang and Shen 2023; Miao et al. 2023; Fang et al. 2023; Ma et al. 2022; Zhang, Luo, and Wei 2023). These methods generally fall into two categories based on granularity: i) instance-level explanation (Ying et al. 2019; Zhang et al. 2022), which explains predictions for each instance by identifying significant substructures; and ii) model-level explanation (Yuan et al. 2020; Wang and Shen 2023; Azzolin et al. 2023), designed to understand global decision rules incorporated by the target GNN. Among these methods, Post-hoc explanation ones (Ying et al. 2019; Luo et al. 2020; Yuan et al. 2021), which employ another model or approach to explain a target GNN. Post-hoc explanations have the advantage of being model-agnostic, meaning they can be applied to a variety of GNNs. Therefore, our work focuses on post-hoc instance-level explanations (Ying et al. 2019), that is, identifying crucial instance-wise substructures for each input to explain the prediction using a trained GNN model. A detailed survey can be found in (Yuan et al. 2022).

Experimental Study

In this section, we empirically verify the effectiveness and efficiency of the proposed K-FactExplainer by explaining both node and graph classifications. We also conduct extensive studies to verify our theoretical claims. Due to the space limitation, detailed experimental setups, full experimental results, and extensive experiments are presented in Appendix.

Experimental Setup. We compare our method with representative GIB-based explanation methods, GNNExplainer (Ying et al. 2019) and PGExplainer (Luo et al. 2020), classic explanation methods, GRAD (Ying et al. 2019) and ATT (Veličković et al. 2018), and SOTA methods, RGExplainer (Shan et al. 2021) and DEGREE (Feng et al. 2022). We follow the routinely adopted framework to set up our experiments (Ying et al. 2019; Luo et al. 2020). Six benchmark datasets with ground truth explanations are used for evaluation, with BA-Shapes, BA-Community, Tree-Circles, and Tree-Grid (Ying et al. 2019) for the node classification task, and BA-2motifs (Luo et al. 2020) and MUTAG (Debnath et al. 1991) for the graph classification task. For each dataset, we train a graph neural network model to perform the node or graph classification task. Each model is a three-layer GNN with a hidden size of 20, followed by an MLP that maps these embeddings to the number of classes. After training the model, we apply the K-FactExplainer and the baseline

	BA-Shapes	BA-Community	Tree-Circles	Tree-Grid	BA-2motifs	MUTAG
PGExp.	$0.999_{\pm 0.000}$	$0.825_{\pm 0.040}$	$0.760_{\pm 0.014}$	$0.679_{\pm 0.008}$	$0.566_{\pm 0.004}$	$0.843_{\pm 0.162}$
$k = 1$	$\mathbf{1.000}_{\pm 0.000}$	$0.850_{\pm 0.047}$	$0.758_{\pm 0.023}$	$0.711_{\pm 0.011}$	$\underline{0.580}_{\pm 0.041}$	$0.769_{\pm 0.119}$
$k = 2$	$\underline{\mathbf{1.000}}_{\pm 0.000}$	$0.880_{\pm 0.023}$	$\mathbf{0.779}_{\pm 0.018}$	$\underline{0.707}_{\pm 0.570}$	$0.581_{\pm 0.039}$	$0.801_{\pm 0.105}$
$k = 3$	$\mathbf{1.000}_{\pm 0.000}$	$\underline{0.902}_{\pm 0.022}$	$\underline{0.772}_{\pm 0.012}$	$0.710_{\pm 0.005}$	$0.586_{\pm 0.034}$	$0.895_{\pm 0.034}$
$k = 5$	$\mathbf{1.000}_{\pm 0.000}$	$0.899_{\pm 0.011}$	$0.768_{\pm 0.013}$	$0.709_{\pm 0.006}$	$0.573_{\pm 0.044}$	$0.892_{\pm 0.030}$
$k = 10$	$\mathbf{1.000}_{\pm 0.000}$	$0.926_{\pm 0.012}$	$0.774_{\pm 0.006}$	$0.706_{\pm 0.004}$	$0.578_{\pm 0.039}$	$\mathbf{0.915}_{\pm 0.021}$
$k = 20$	$\mathbf{1.000}_{\pm 0.000}$	$0.938_{\pm 0.013}$	$0.778_{\pm 0.006}$	$0.704_{\pm 0.002}$	$0.586_{\pm 0.032}$	$0.911_{\pm 0.014}$
$k = 60$	$\mathbf{1.000}_{\pm 0.000}$	$\mathbf{0.952}_{\pm 0.011}$	$0.778_{\pm 0.004}$	$\mathbf{0.770}_{\pm 0.004}$	$\mathbf{0.588}_{\pm 0.030}$	$\underline{\mathbf{0.915}}_{\pm 0.010}$

Table 2: Explanation performances w.r.t. k. We use underlines to denote k selected by the proposed method.

	$k=1$	$k=2$	$k=3$	$k=5$	$k=10$	$k=20$	$k=60$
BA-Community(20)	0.850	0.880	0.902	0.899	0.926	0.938	0.952
BA-Community(80)	0.893	0.899	0.895	0.894	0.895	0.895	0.897
Tree-Circles(20)	0.758	0.779	0.772	0.768	0.774	0.778	0.778
Tree-Circles(80)	0.871	0.871	0.871	0.870	0.871	0.871	0.870

Table 3: Effects of lossy aggregation on GNNs with different hidden layer sizes

methods to generate explanations for both node and graph instances. For each experiment, we conduct 10 times with random parameter initialization and report the average results as well as the standard deviation. Detailed experimental setups are put in the appendix.

Quantitative Evaluation

Comparison to Baselines. We adopted the well-established experimental framework (Ying et al. 2019; Luo et al. 2020; Shan et al. 2021), where the explanation problem is framed as a binary classification of edges. Within this setup, edges situated inside motifs are regarded as positive edges, while all others are treated as negative. The importance weights offered by the explanation methods are treated as prediction scores. An effective explanation method, therefore, would assign higher weights to edges located within the ground truth motifs as opposed to those outside. To quantitatively evaluate the performance of these methods, we employed AUC as our metric. The average AUC scores and the associated standard deviations are reported in Table 1. We observe that with a manually selected value for k, K-FactExplainer consistently outperforms GNNExplainer and PGExplainer and competes with high-performing models like RGExplainer and DEGREE. The comparison demonstrates that our K-FactExplainer considers locality, providing more accurate, comprehensive explanations and mitigating common locality pitfalls seen in other models.

Model Analysis

Effectiveness of Bootstrapping Algorithm. To directly show the effects of k in K-FactExplainer . We change the value of k from 1 to 60 and show the resulting performance in Table 2. We observe that, in general, a higher value of k leads to improved performance. The reason is that large k in K-FactExplainer mitigates the locality and lossy aggregation losses in parametric explainers as discussed previously. We use an underline to indicate the upper-bound for the value of k suggested by the bootstrap algorithm in Section . It should be noted that this upper-bound is particularly relevant to multi-motif scenarios considered in Theorem 2. Restricting to values of k that are less than or equal to the suggested upper-bound achieves the best performance in the multi-motif MUTAG task, which is aligned with our theoretical analysis.

Effects of Lossy Aggregation. To evaluate the effects of lossy aggregation, we consider the BA-Community and Tree-Cycles in this part. As shown in Table 2, K-FactExplainer significantly outperforms PGExplainer. The reason is that the K-FactExplainer partially mitigates the aggregation loss in GNN explanation methods by combining the outputs of multiple MLPs, hence combining multiple 'weak' explainers into a stronger one. In addition, we observe that the performances of K-FactExplainer are positively related to k. Next, we increase the dimensionality of hidden representation in the GNN model from 20 to 80. This reduces the loss in aggregation as at each layer several low dimensional vectors are mapped to high dimensional vectors. The explanation performances are shown in Table 3. For these two datasets, the performance improves as k is increased when the dimension is 20, due to the mitigation of the aggregation loss, however, as expected, no improvement is observed when increasing k in explaining the GNN with dimension 80, since there is no significant aggregation loss to mitigate in that case.

Conclusion

In this work, we theoretically investigate the trivial solution problem in the popular objective function for explaining GNNs, which is largely overlooked by the existing post-hoc instance-level explanation approaches. We point out that the trivial solution is caused by the signal problem and propose a new GIB objective with a theoretical guarantee. On top of that, we further investigate the locality and lossy aggregation issues in existing parametric explainers and show that most of them can be unified within the local explanation Methods, which are weak at handling real-world graphs, where the mapping between labels and motifs is one-to-many. We propose a new factorization-based explanation model to address these issues. Comprehensive experiments are conducted to verify the effectiveness of the proposed method.

Acknowledgments

This project was partially supported by NSF grants IIS-2331908 and CCF-2241057. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Abid, A.; Balin, M. F.; and Zou, J. 2019. Concrete Autoencoders for Differentiable Feature Selection and Reconstruction. arXiv:1901.09346.

Azzolin, S.; Longa, A.; Barbiero, P.; Liò, P.; and Passerini, A. 2023. Global explainability of gnns via logic combination of learned concepts. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bouritsas, G.; Frasca, F.; Zafeiriou, S.; and Bronstein, M. M. 2022. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 657–668.

Chereda, H.; Bleckmann, A.; Kramer, F.; Leha, A.; and Beißbarth, T. 2019. Utilizing Molecular Network Information via Graph Convolutional Neural Networks to Predict Metastatic Event in Breast Cancer. *Studies in health technology and informatics*, 267: 181–186.

Debnath, A. K.; Lopez de Compadre, R. L.; Debnath, G.; Shusterman, A. J.; and Hansch, C. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2): 786–797.

El Gamal, A.; and Kim, Y.-H. 2010. Lecture notes on network information theory.

Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, Y.; Tang, J.; and Yin, D. 2019. Graph Neural Networks for Social Recommendation. *The World Wide Web Conference*.

Fang, J.; Wang, X.; Zhang, A.; Liu, Z.; He, X.; and Chua, T.-S. 2023. Cooperative Explanations of Graph Neural Networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 616–624.

Feng, Q.; Liu, N.; Yang, F.; Tang, R.; Du, M.; and Hu, X. 2022. DEGREE: Decomposition Based Explanation for Graph Neural Networks. In *International Conference on Learning Representations*.

Feng, W.; Zhang, J.; Dong, Y.; Han, Y.; Luan, H.; Xu, Q.; Yang, Q.; Kharlamov, E.; and Tang, J. 2020. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33: 22092–22103.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R. P.; Tang, J.; and Liu, H. 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6): 1–45.

Lin, W.; Lan, H.; and Li, B. 2021. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*, 6666–6679. PMLR.

Liu, Z.; Yang, L.; Fan, Z.; Peng, H.; and Yu, P. S. 2022. Federated social recommendation with graph neural network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4): 1–24.

Lu, Z.; Lv, W.; Xie, Z.; Du, B.; Xiong, G.; Sun, L.; and Wang, H. 2022. Graph Sequence Neural Network with an Attention Mechanism for Traffic Speed Prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(2): 1–24.

Luo, D.; Cheng, W.; Xu, D.; Yu, W.; Zong, B.; Chen, H.; and Zhang, X. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33: 19620–19631.

Ma, J.; Guo, R.; Mishra, S.; Zhang, A.; and Li, J. 2022. CLEAR: Generative Counterfactual Explanations on Graphs. In *Proceedings of Advances in neural information processing systems*.

Mansimov, E.; Mahmood, O.; Kang, S.; and Cho, K. 2019. Molecular Geometry Prediction using a Deep Generative Graph Neural Network. *Scientific Reports*, 9.

Miao, S.; Liu, M.; and Li, P. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*, 15524–15543. PMLR.

Miao, S.; Luo, Y.; Liu, M.; and Li, P. 2023. Interpretable Geometric Deep Learning via Learnable Randomness Injection. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Satorras, V. G.; Hoogeboom, E.; and Welling, M. 2021. E (n) equivariant graph neural networks. In *International conference on machine learning*, 9323–9332. PMLR.

Shan, C.; Shen, Y.; Zhang, Y.; Li, X.; and Li, D. 2021. Reinforcement Learning Enhanced Explainer for Graph Neural Networks. *Advances in Neural Information Processing Systems*, 34.

Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, 1–5. IEEE.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, X.; and Shen, H.-W. 2023. GNNInterpreter: A Probabilistic Generative Model-Level Explanation for Graph Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Wang, X.; Wu, Y.; Zhang, A.; He, X.; and Chua, T.-S. 2021. Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, 34: 18446–18458.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph information bottleneck. *Advances in Neural Information Processing Systems*, 33: 20437–20448.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*, 974–983.

Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, 9240–9251.

Yu, J.; Xu, T.; Rong, Y.; Bian, Y.; Huang, J.; and He, R. 2021. Graph Information Bottleneck for Subgraph Recognition. In *International Conference on Learning Representations*.

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

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

Yuan, H.; Yu, H.; Wang, J.; Li, K.; and Ji, S. 2021. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning*, 12241–12252. PMLR.

Zhang, J.; Luo, D.; and Wei, H. 2023. MixupExplainer: Generalizing Explanations for Graph Neural Networks with Data Augmentation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3286–3296.

Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31.

Zhang, S.; Liu, Y.; Shah, N.; and Sun, Y. 2022. Gstarx: Explaining graph neural networks with structure-aware cooperative games. *Advances in Neural Information Processing Systems*, 35: 19810–19823.