Interpreting Graph Neural Networks with In-Distributed Proxies

Zhuomin Chen
Florida International University
Miami, Florida, USA
zchen051@fiu.edu

Jiaxing Zhang
New Jersey Institute of Technology
Newark, New Jersey, USA
jz48@njit.edu

Jingchao Ni
AWS AI Labs
Seattle, Washington, USA
nijingchao@gmail.com

Xiaoting Li
Visa Research
Foster City, California, USA
xiaotili@visa.com

Yuchen Bian
Amazon Search A9
Palo Alto, California, USA
yuchbian@amazon.com

Md Mezbahul Islam
Florida International University
Miami, Florida, USA
misla093@fiu.edu

Ananda Mohan Mondal
Florida International University
Miami, Florida, USA
amondal@fiu.edu

Hua Wei
Arizona State University
Tempe, Arizona, USA
hwei27@asu.edu

Dongsheng Luo
Florida International University
Miami, Florida, USA
dluo@fiu.edu

ABSTRACT

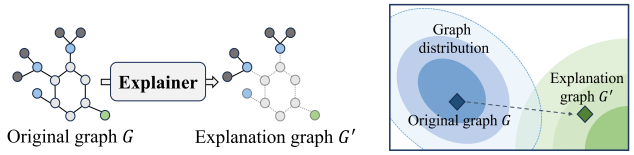
Graph Neural Networks (GNNs) have become a building block in graph data processing, with wide applications in critical domains. The growing needs to deploy GNNs in high-stakes applications necessitate explainability for users in the decision-making processes. A popular paradigm for the explainability of GNNs is to identify explainable subgraphs by comparing their labels with the ones of original graphs. This task is challenging due to the substantial distributional shift from the original graphs in the training set to the set of explainable subgraphs, which prevents accurate prediction of labels with the subgraphs. To address it, in this paper, we propose a novel method that generates proxy graphs for explainable subgraphs that are in the distribution of training data. We introduce a parametric method that employs graph generators to produce proxy graphs. A new training objective based on information theory is designed to ensure that proxy graphs not only adhere to the distribution of training data but also preserve essential explanatory factors. Such generated proxy graphs can be reliably used for approximating the predictions of the true labels of explainable subgraphs. Empirical evaluations across various datasets demonstrate our method achieves more accurate explanations for GNNs.

KEYWORDS

Graph Neural Networks, Trustworthy Learning, Explainable AI

1 INTRODUCTION

Graph Neural Networks (GNNs) have emerged as a pivotal technology for handling graph-structured data, demonstrating remarkable performance in various applications including node classification and link prediction [10, 15, 25, 28]. Their growing use in critical sectors such as healthcare and fraud detection has escalated the need for explainability in their decision-making processes [17, 31, 35]. To meet this demand, a variety of explanation methods have been recently developed to interpret the behavior of GNN models. These methods primarily concentrate on identifying a subgraph that significantly impacts the model's prediction for a particular instance [18, 33].



(a) Explanation process **(b) OOD problem**

Figure 1: Examples of the explanation process and out-of-distribution problem. (a) is the explanation process for a graph learning model. The original graph G undergoes an explanation process, resulting in an explanation graph G' that highlights the most significant features and relationships; (b) shows the explanation graph G' is out of distribution where the GNN is well-trained.

A prominent approach to explain GNNs involves the Graph Information Bottleneck (GIB) principle [32]. This principle focuses on extracting a compact yet informative subgraph from the input graph, ensuring that this subgraph retains sufficient information for the model to maintain its original prediction. A key aspect of the GIB approach is evaluating the predictive capability of such a subgraph. Typically, this is accomplished by feeding the subgraph into the GNN model and comparing its prediction against that of the complete input graph.

Although it is intuitively correct, the underlying assumption of the aforementioned approach – GNN model can make accurate predictions on explanation subgraphs – may not always hold. Prior research indicates that explanation subgraphs can significantly deviate from the distribution of original graphs, leading to an Out-Of-Distribution (OOD) issue [7, 9, 37]. For instance, in the MUTAG dataset [4], each graph represents a molecule, with nodes symbolizing atoms and edges indicating chemical bonds. The molecular graphs in this dataset usually contain hundreds of edges. In contrast, the NO_2 functional group, identified as a key subgraph influencing positive mutagenicity in a molecule, comprises merely 2 edges. This stark contrast in structural properties leads to a significant

difference in the distributions of explanation subgraphs and original graphs. Since the model was trained with original graphs, the reliability of predictions on subgraphs is undermined due to the distribution shifting problem.

Several pioneering studies have attempted to address this distributional challenge [9, 36, 37]. For example, CGE regards the GNN model as a transparent, fully accessible system. It considers the GNN model as a teacher network and employs an additional "student" network to predict the labels of explanation subgraphs [9]. As another example, MixupExplainer [37] generates a mixed graph for the explanation by blending it with a non-explanatory subgraph from a different input graph. This method posits that the mixup graph aligns with the distribution of the original input graphs. However, this claim is predicated on a rather simplistic assumption that the explanation and non-explanatory subgraphs are independently drawn. However, in real-world applications, these methods often face practical limitations. The dependence on a "white box" model in CGE and the oversimplified assumptions in MixupExplainer are not universally applicable. Instead, GNN models are usually given as "black boxes", and the graphs in real-life applications do not conform to strict independence constraints, highlighting the need for more versatile and realistic approaches to the OOD problem.

In response to these challenges, in this work, we introduce an innovative concept of proxy graphs to the realm of explainable GNNs. These proxy graphs are designed to be both explanation-preserving and distributionally aligned with the training data. By this means, the predictive capabilities of explanation subgraphs can be reliably inferred from the corresponding proxy graphs. We begin with a thorough investigation into the feasible conditions necessary for generating such in-distributed proxy graphs. Leveraging our findings, we further propose a novel architecture that incorporates variational graph auto-encoders to produce proxy graphs. Specifically, we utilize a graph auto-encoder to reconstruct the explainable subgraph and another variational auto-encoder to generate a non-explanatory subgraph. A proxy graph is then obtained by combining two output subgraphs of auto-encoders. We delineate our main contributions as follows:

- We systematically analyze and address the challenge of the out-of-distribution issue in explainable GNNs, which is pivotal for enhancing the reliability and interpretability of GNNs in real-world applications.
- We introduce an innovative parametric method that incorporates graph auto-encoders to produce in-distributed proxy graphs that are both situated in the original data distribution and preserve essential explanation information. This facilitates more precise and interpretable explanations in GNN applications.
- Through comprehensive experimentation and rigorous evaluations on various real-world datasets, we substantiate the effectiveness of our proposed approach, showcasing its practical utility and superiority in producing explanations.

2 NOTATIONS AND PRELIMINARY

2.1 Notations and Problem Formulation.

We denote a graph G from an alphabet $\mathcal{G}$ by a triple $(\mathcal{V}, \mathcal{E}; \boldsymbol{X})$, where $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ is the node feature matrix, where d is the feature dimension and the i-th row is the feature vector associated with node v_i. The adjacency matrix of G is denoted by $\boldsymbol{A} \in \{0, 1\}^{n \times n}$, which is determined by the edge set $\mathcal{E}$ that $A_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, $A_{ij} = 0$, otherwise. In this paper, we focus on the graph classification task, as the node classification can be converted to computation graph classification problem [18, 33]. Specifically, for the graph classification task, each graph G is associated with a label $Y \in \boldsymbol{C}$. The to-be-explained GNN model $f(\cdot)$ has been well-trained to classify G into its class, i.e., $f : \mathcal{G} \mapsto \{1, 2, \cdots, |C|\}$.

Following existing works [18, 33, 34], the explanation methods under consideration in this paper are model/task agnostic and treat GNN models as black boxes — i.e., the so-called *post-hoc, instance-level* explanation methods. Formally, our research problem is described as follows [11]:

Problem 1 (Post-hoc Instance-level GNN Explanation). *Given a to-be-explained GNN model $f(\cdot)$ and a set of graphs $\mathcal{G}$, the goal of post-hoc instance-level explanation is to learn a parametric function that for an arbitrary graph $G \in \mathcal{G}$, it finds a compact subgraph $G^* \subseteq G$ that can "explain" the prediction $f(G)$. The parametric mapping $\Psi_\psi : \mathcal{G} \mapsto \mathcal{G}^*$ is called an explanation function, where $\mathcal{G}^*$ is the alphabet of G^*, and ψ is the parameter of the explanation function.*

2.2 Graph Information Bottleneck as the Objective Function.

The Information Bottleneck (IB) principle, foundational in learning dense representations, suggests that optimal representations should balance minimal and sufficient information for predictions [27]. This concept has been adapted for GNNs through the Graph Information Bottleneck (GIB) approach, consolidating various post-hoc GNN explanation methods like GNNExplainer [33] and PG-Explainer [18]. The GIB framework aim to find a subgraph G^* to explain the GNN model's prediction on a graph G as [37]:

$$G^* = \arg\min_{G'} I(G, G') - \alpha I(Y, G'), \tag{1}$$

where G' is a candidate explanatory subgraph, Y is the label, and α is a balance parameter. The first term is the mutual information between the original graph G and the explanatory subgraph G' and the second term is negative mutual information of Y and the explanation subgraph G'. At a high level, the first term encourages detecting a small and dense subgraph for explanation, and the second term requires that the explanation subgraph is label preserving.

Due to the intractability of mutual information between Y and G' in equation 1, existing works [20, 32] derive a parameterized variational lower bound of $I(Y, G')$:

$$I(Y, G') \geq \mathbb{E}_{G',Y}[\log P(Y|G')] + H(Y), \tag{2}$$

where the first term measures how well the subgraphs predict the labels. A higher value indicates that the subgraphs are, on average, more predictive of the correct labels. The second term $H(Y)$ quantifies the amount of inherent unpredictability or variability in the labels. By introducing equation 2 to equation 1, a tractable upper bound is used as the objective:

$$G^* = \arg\min_{G'} I(G, G') - \alpha \mathbb{E}_{G',Y}[\log P(Y|G')], \tag{3}$$

where $H(Y)$ is omitted due to its independence to G'.

The OOD Problem in GIB. In existing research, the estimation of $P(Y|G')$ is typically achieved by applying the to-be-explained model f, to the input graph G' [20, 33]. A critical assumption in existing approaches involves the GNN model's ability to accurately predict candidate explanation subgraphs G'. This assumption, however, often overlooks the OOD problem, where the distribution of explanation subgraphs significantly deviates from that of the original training graphs [6, 37]. Formally, let $P_{\mathcal{G}}$ denote the distribution of training graphs, and $P_{\mathcal{G}'}$ represents the distribution of explanation subgraphs. The core issue in the GIB objective function is caused by $P_{\mathcal{G}} \neq P_{\mathcal{G}'}$. This distributional disparity undermines the predictive reliability of the model $f(\cdot)$, trained on $P_{\mathcal{G}}$ when applied to subgraphs from $P_{\mathcal{G}'}$. As a result, the predictive power of explanations provided by $f(G')$ is unreliable approximation of $P(Y|G')$ in equation 3.

3 GRAPH INFORMATION BOTTLENECK WITH PROXY GRAPHS

In this section, we propose the concept of a *proxy graph* to mitigate the above-mentioned OOD issue. This proxy graph not only retains the label information present in G' but also conforms to the distribution of the original graph dataset. Specifically, we assume that proxy graphs $\tilde{G}$ are drawn from a distribution $P_{\mathcal{G}}$ and reformulate the estimation of $P(Y|G')$ by marginalizing over the distribution of proxy graphs as follows.

$$P(Y|G') = \mathbb{E}_{\tilde{G}\sim P_{\mathcal{G}}}[P(Y|\tilde{G}) \cdot P(\tilde{G}|G')] \tag{4}$$

In equation 4, we address the OOD challenge by predicting Y using a proxy graph $\tilde{G}$ instead of directly using G'. This approach is particularly effective when the conditional probability $P(Y|\cdot)$ is approximated by the model $f(\cdot)$. To facilitate the maximization of the likelihood as outlined in equation 4, we further approximate $P(\tilde{G}|G')$ with a parameterized function, denoted as $Q_{\boldsymbol{\phi}}(\tilde{G}|G')$. The formal representation is thus given by

$$P(Y|G') = \mathbb{E}_{\tilde{G}\sim P_{\mathcal{G}}}[P(Y|\tilde{G}) \cdot Q_{\boldsymbol{\phi}}(\tilde{G}|G')], \tag{5}$$

where $\boldsymbol{\phi}$ denotes model parameters.

Given the combinatorial complexity inherent in graph structures, it is computationally infeasible to enumerate all potential proxy graphs $\tilde{G}$ from the unknown distribution $P_{\mathcal{G}}$, which is necessary for calculating equation 5. To overcome this challenge, we propose approximating $P_{\mathcal{G}}$ with the parameterized function $Q_{\boldsymbol{\phi}}(\tilde{G}|G')$. Consequently, we estimate equation 5 by sampling proxy graphs from $Q_{\boldsymbol{\phi}}(\tilde{G}|G')$, leading to the following formulation:

$$P(Y|G') = \mathbb{E}_{\tilde{G}\sim Q_{\boldsymbol{\phi}}(\tilde{G}|G')}[P(Y|\tilde{G})], \tag{6}$$

with constraints

$$Q_{\boldsymbol{\phi}}(\tilde{G}|G') \approx P_{\mathcal{G}}, \quad H(Y|\tilde{G}) \approx H(Y|G'), \tag{7}$$

The first constraint ensures that $\tilde{G}$ is sampled from a distribution that approximates $P_{\mathcal{G}}$, effectively addressing the OOD challenge. The second constraint guarantees that the label information preserved in $\tilde{G}$ is similar to that in G'. Therefore, our proxy graph-induced objective function becomes:

$$\begin{aligned} &\arg\min_{G'} I(G,G') - \alpha \mathbb{E}_{G',Y}[\log \mathbb{E}_{\tilde{G}\sim Q_{\boldsymbol{\phi}}(\tilde{G}|G')}[P(Y|\tilde{G})]] \\ &\text{s.t. } Q_{\boldsymbol{\phi}}(\tilde{G}|G') \approx P_{\mathcal{G}},\ H(Y|\tilde{G}) \approx H(Y|G') \end{aligned} \tag{8}$$

Bi-level optimization In equation 8, we formulate a joint optimization loss function that aims to identify the optimal explanation alongside its corresponding proxy graphs. Building upon this, we refine the framework by replacing the first constraint with a distributional distance measure, specifically the Kullback-Leibler (KL) divergence, between $Q_{\boldsymbol{\phi}}(\tilde{G}|G')$ and $P_{\mathcal{G}}$. This leads to the development of a bi-level optimization model. Formally, the model is expressed as follows:

$$\begin{aligned} &\arg\min_{G'} I(G,G') - \alpha \mathbb{E}_{G',Y}[\log \mathbb{E}_{\tilde{G}\sim Q_{\boldsymbol{\phi}^*}(\tilde{G}|G')}[P(Y|\tilde{G}^*)]] \\ &\text{where } \boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathrm{KL}(Q_{\boldsymbol{\phi}}(\tilde{G}|G'), P_{\mathcal{G}}), \\ &\qquad \text{s. t. } H(Y|\tilde{G}) \approx H(Y|G'). \end{aligned} \tag{9}$$

3.1 Derivation of Outer Optimization

For the outer optimization objective, we elaborate on operationalizing $I(G,G')$. Akin to the original graph, We denote the explanation subgraph G' by $(\mathcal{V}, \mathcal{E}', \boldsymbol{X})$, whose adjacency matrix is $\boldsymbol{A'}$. We follow [20] to include a variational approximation distribution $R(G')$ for the distribution $P(G')$. Then, we obtain its upper bounds as follows.

$$I(G,G') \leq \mathbb{E}_G[\mathrm{KL}(P(G'|G)||R(G'))] \tag{10}$$

We follow the Erdős–Rényi model [5] and assume that each edge in G' has a probability of being present or absent, independently of the other edges. Specifically, we assume that the existence of an edge (u,v) in G' is determined by a Bernoulli variable $A'_{u,v} \sim \mathrm{Bern}(\pi_{uv})$. Thus, $P(G'|G)$ can be decompose with $P(G'|G) = \prod_{(u,v)\in\mathcal{E}} P(A'_{uv}|\pi_{uv})$. The bound is always true for any prior distribution $R(G')$. We follow an existing work and assume that in prior distribution, the existence of an edge (u,v) in G' is determined by another Bernoulli variable $A''_{u,v} \sim \mathrm{Bern}(r)$, where $r \in [0,1]$ is a hyper-parameter [20], independently to the graph G. We have $R(G') = P(|\mathcal{E}|)\prod_{(u,v)\in\mathcal{E}} P(A''_{uv})$. Thus, the KL divergence between $P(G'|G)$ and the above marginal distribution $R(G')$ becomes:

$$\begin{aligned} &\mathrm{KL}(P(G'|G)||R(G')) \\ &= \sum_{(u,v)\in\mathcal{E}} \pi_{uv}\log\frac{\pi_{uv}}{r} + (1-\pi_{uv})\log\frac{1-\pi_{uv}}{1-r} + \mathrm{Const.} \end{aligned} \tag{11}$$

By replacing $I(G'G')$ with the above tractable upper bound, we obtain the loss function, denoted by $\mathcal{L}_{\text{exp}}$, to train the explainer $\Psi(\cdot)$.

3.2 Derivation of Inner Optimization

For the inner optimization objective, we implement the first constraint by minimizing the distribution distance between $Q_{\boldsymbol{\phi}}(\tilde{G}|G')$ and $P_{\mathcal{G}}$, under the Erdős–Rényi assumption, the distribution loss is equivalent to the cross-entropy loss between $\tilde{G}$ given G' and G over the full adjacency matrix [3]. Considering that G is usually sparse compared to a fully connected graph, in practice, we adopt

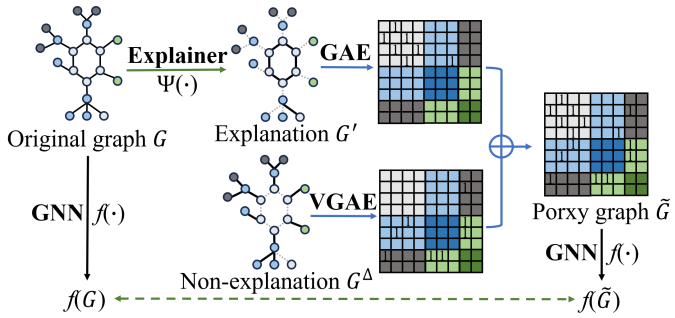


Figure 2: ProxyExplainer consists of two components: an Explainer and a Proxy Graph Generator. The Explainer takes a graph G as input and produces an explainable subgraph G', The proxy generator creates an in-distribution proxy graphs that preserve the label information in G'. The proxy generator consists of a graph auto-encoder (GAE) and a variational graph auto-encoder (VGAE).

a weighted version to emphasize more connected node pairs [29]. Formally, we have the following distribution loss.

$$\mathcal{L}_{\text{dist}} = \frac{\beta}{|\mathcal{E}|} \sum_{(u,v)\in\mathcal{E}} \log(\tilde{p}_{uv}) + \frac{1}{|\bar{\mathcal{E}}|} \sum_{(u,v)\in\bar{\mathcal{E}}} \log(1-\tilde{p}_{uv}), \quad (12)$$

where $\bar{\mathcal{E}}$ is the set of node pairs that are unconnected in G, $\tilde{p}_{uv}$ is the probability of node pair (u,v) in $\tilde{G}$, and β is a hyper-parameter to get a trade-off between connected and unconnected node pairs.

The second constraint requires the mutual information of Y and $\tilde{G}$ is the same as that of Y and G'. Due to the OOD problem, it is non-trivial to directly compute $P(Y|G')$ or $H(Y|G')$. Instead, we implement this constraint with a novel graph generator that $\tilde{G}$ is obtained by combining G' and a non-explanatory subgraph [37]. In practice, we implement the non-explanatory part by perturbing the remaining subgraph $G - G'$. The intuition is that if an explanation comprises information, it is hard to change the prediction by manipulating the remaining part that is non-explanatory, which is widely adopted in the literature [6, 37, 39].

4 THE PROXYEXPLAINER

Inspired by our novel GIB with proxy graphs, in this section, we introduce a straightforward yet theoretically robust instantiation, named ProxyExplainer. As shown in Figure 2, the architecture of our model comprises two key modules: the explainer and the proxy graph generator. The explainer takes the G as input and outputs a subgraph G' as an explanation, which is optimized through the outer objective in equation 9. The proxy graph generator generates an in-distributed proxy graph, which is optimized with the inner objective.

4.1 The Explainer

For the sake of optimization, we follow existing works to relax the element in $\boldsymbol{A}$ from binary values to continuous values in the range $[0, 1]$ [18, 20, 33]. We adopt a generative explainer due to its effectiveness and efficiency [18]. Specifically, we decompose the to-be-explained model $f(\cdot)$ into two functions that $f_{\text{enc}}(\cdot)$ learns node representations and $f_{\text{cls}}(\cdot)$ predicts graph labels based on node embeddings. Formally, we have $\boldsymbol{Z} = f_{\text{enc}}(\boldsymbol{X}, \boldsymbol{A})$ and $Y = f_{\text{cls}}(\boldsymbol{Z})$, where $\boldsymbol{Z} \in \mathbb{R}^{N\times F}$ is the node representation matrix and F is the number of feature dimenions. Routinely, $f_{\text{cls}}(\cdot)$ consists of a pooling layer followed by a classification layer and $f_{\text{enc}}(\cdot)$ consists of the other layers. Following the independence assumption in Section 3.1, we approximate Bernoulli distributions with binary concrete distributions [12, 19]. Specifically, the probability of sampling an edge (i, j) is computed by an MLP parameterized by ψ, denoted by $g_\psi(\cdot)$. Formally, we have

$$\begin{aligned} \omega_{ij} &= g_\psi([\boldsymbol{z}_i; \boldsymbol{z}_j]), \qquad \epsilon \sim \text{Uniform}(0,1) \\ A'_{ij} &= \sigma((\log\epsilon - \log(1-\epsilon) + \omega_{ij})/\tau), \end{aligned} \quad (13)$$

where $[\boldsymbol{z}_i; \boldsymbol{z}_j]$ is the concatenation of node representations $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, ϵ is a independent variable, $\sigma(\cdot)$ is the Sigmoid function, and τ is a temperature hyper-parameter for approximation.

4.2 The Proxy Graph Generator

As shown in our analysis in Section 3.2, we demonstrate the synthesis of a proxy graph through the amalgamation of the explanation subgraph G' and the perturbation of its non-explanatory subgraph, represented as $G^{\Delta} = (\mathcal{V}, \mathcal{E}^{\Delta}, \boldsymbol{X})$. We define the edge set of $G^{\Delta}, \mathcal{E}^{\Delta}$, as the differential set $\mathcal{E} - \mathcal{E}'$. Correspondingly, the adjacency matrix $\boldsymbol{A}^{\Delta}$ is derived through $\boldsymbol{A} - \boldsymbol{A}'$. Building upon this foundation, and as illustrated in Figure 2, our proposed framework introduces a dual-structured mechanism comprising two distinct graph auto-encoders(GAE) [1]. To be more specific, we first include an encoder to learn a latent matrix $\boldsymbol{Z}'$ according to $\boldsymbol{A}'$ and $\boldsymbol{X}'$, and a decoder network recovers $\boldsymbol{A}'$ based on $\boldsymbol{Z}'$. Formally, we have:

$$\boldsymbol{Z}' = \text{ENC}_1(\boldsymbol{A}', \boldsymbol{X}), \quad \tilde{\boldsymbol{A}}' = \text{DEC}(\boldsymbol{Z}'), \quad (14)$$

where ENC_1 and DEC are the encoder network and decoder network. Our framework is flexible to the choices of these two networks. $\tilde{\boldsymbol{A}}' \in \mathbb{R}^{n\times n}$ is the reconstructed adjacency matrix.

To introduce perturbations into the non-explanatory segment, our approach employs a Variational Graph Auto-Encoder (VGAE), a generative model adept at creating varied yet structurally coherent graph data. This capability is pivotal in generating nuanced variations of the non-explanatory subgraph. The VGAE operates by first encoding the non-explanatory subgraph G^{Δ} into a latent probabilistic space, characterized by Gaussian distributions. This process is articulated as:

$$\boldsymbol{\mu}^{\Delta} = \text{ENC}_1(\boldsymbol{A}^{\Delta}, \boldsymbol{X}), \quad \boldsymbol{\sigma}^{\Delta} = \text{ENC}_2(\boldsymbol{A}^{\Delta}, \boldsymbol{X}), \quad (15)$$

where ENC_1 and ENC_2 are encoder networks that learn the mean $\boldsymbol{\mu}^{\Delta}$ and variance $\boldsymbol{\sigma}^{\Delta}$ of the Gaussian distributions, respectively. Following this, the latent representations $\boldsymbol{Z}^{\Delta}$ are sampled from these distributions, ensuring that each generated instance is a unique variation of the original. The decoder network then reconstructs the perturbed non-explanatory subgraph from these sampled latent representations:

$$\boldsymbol{Z}^{\Delta} \sim \mathcal{N}(\boldsymbol{\mu}^{\Delta}, \text{diag}(\boldsymbol{\sigma}^{\Delta})^2), \quad \tilde{\boldsymbol{A}}^{\Delta} = \text{DEC}(\boldsymbol{Z}^{\Delta}), \quad (16)$$

where $\tilde{\boldsymbol{A}}^{\Delta} \in \mathbb{R}^{n\times n}$ represents the adjacency matrix of the perturbed non-explanatory subgraph. This novel use of VGAE facilitates the generation of diverse yet representative perturbations, crucial for enhancing the interpretability of explainers in our proxy graph

framework. The adjacency matrix of a proxy graph is then obtained by

$$\tilde{A} = \tilde{A}' + \tilde{A}^{\Delta}. \tag{17}$$

Loss function. To train the proxy graph generator, we introduce a standard Gaussian distribution as the prior for the latent space representations in the VGAE, specifically $Z \sim \mathcal{N}(0, \mathcal{I})$, where $\mathcal{I}$ represents the identity matrix. Then, the loss function is as follows.

$$\mathcal{L}_{\text{proxy}} = \mathcal{L}_{\text{dist}} + \lambda \mathcal{L}_{\text{KL}}, \tag{18}$$

where $\mathcal{L}_{\text{dist}}$ is the distribution distance between $\tilde{G}$ and G. $\mathcal{L}_{\text{KL}}$ represents the Kullback-Leibler (KL) divergence between the distribution of the latent representations Z^{Δ} and the assumed Gaussian prior. This term is crucial for regulating the variational aspect of the VGAE, ensuring that the generated perturbations are meaningful and controlled. λ is the hyper-parameter.

Alternative Training. To train explainer and proxy graph generator networks, we follow existing works [39] to use an alternating training schedule that trains the proxy graph generator network M times and then trains the explainer network one time. M is a hyper-parameter determined by grid search. The detailed algorithm description of our model is shown in Appendix B.

5 RELATED WORK

GNN Explanation. The goal of explainability in GNNs is to ensure transparency in graph-based tasks. Recent works have been directed towards elucidating the rationale behind GNN predictions. These explanation methods can be broadly classified into two categories: instance-level and model-level approaches [34]. In this study, we focus on instance-level explanations, which aim to clarify the specific reasoning behind individual predictions made by GNNs. These methods are critical for understanding the decision-making process on a case-by-case basis to enhance the explainability of GNNs in graph classification tasks. For example, GNNExplainer [33] excludes certain edges and node features to observe the changes in classification. However, its single-instance focus limits its applicability to provide a global understanding of the to-be-explained model. PGExplainer [18] introduces a parametric neural network to learn edge weights. Thus, once training is complete, it can explain new graphs without retraining. ReFine [30] integrates a pre-training phase that focuses on class comparisons and is fine-tuned to refine context-specific explanations. GStarX [38] assigns an importance score to each node by caculating the Hamiache and Navarro values of the structure to obtain explanatory subgraphs. GFlowExplainer [16] uses a generator to construct a TD-like flow matching condition to learn a policy for generating explanations by adding nodes sequentially.

Distribution Shifting in Explanations. The distribution shifting problem in post-hoc explanations has been increasingly recognized in explainable AI fields [2, 22]. For example, FIDO [2] works on enhancing image classifier explanations, focusing on relevant contextual details that agree with the training data's distribution. A recent study tackles the distribution shifting problem in image explanations by introducing a module that assesses the similarity between altered data and the original dataset distribution [22]. In the graph domain, an ad-hoc strategy to mitigate distribution shifting is to initially reduce the size constraint coefficient during the explanation process [8]. MixupExplainer [37] proposes a non-parametric solution by mixing up the explanation subgraph with a non-explainable part from another graph. However, this method operates under the assumption that the explanatory and non-explanatory subgraphs in mixed graphs are independent, which may not hold in many real-life graphs.

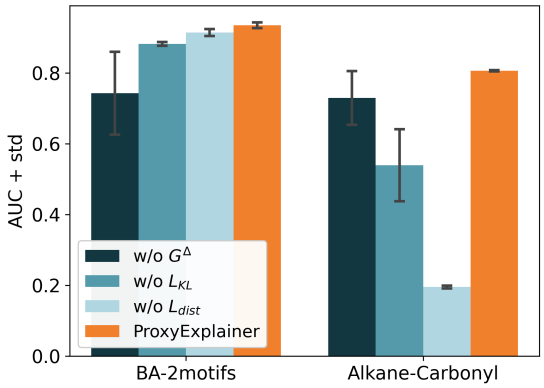


Figure 3: Ablation Studies on BA-2motifs and Alkane-Carbonyl.

6 EXPERIMENTS

We present empirical results that illustrate the effectiveness of our proposed method. These experiments are mainly designed to explore the following research questions:

- **RQ1:** Can the proposed framework outperform other baselines in identifying explanations for GNNs?
- **RQ2:** Is the distribution shifting severe in explanation subgraphs? Can the proposed approach alleviate that?
- **RQ3:** How does each component of ProxyExplainer impact the overall performance in generating explanations?

6.1 Experimental Settings

To evaluate the performance of ProxyExplainer, we use six benchmark datasets with ground-truth explanations. These include two synthetic datasets: BA-2motifs [18] and BA-3motifs [3], along with four real-world datasets: Alkane-Carbonyl [23], Benzene [23], Fluoride-Carbonyl [23], and MUTAG [13]. We take GradCAM [21], GNN-Explainer[33], PGExplainer[18], ReFine [30], and MixupExplainer [37] for comparison. We follow the experimental setting in previous works [18, 23, 33] to train a Graph Convolutional Network (GCN) model with three layers. We use the Adam optimizer [14] with the inclusion of a weight decay $5e-4$. To evaluate the quality of explanations, we approach the explanation task as a binary classification of edges. Edges that are part of ground truth subgraphs are labeled as positive, while all others are deemed negative. The importance weights given by the explanation methods are interpreted as prediction scores. An effective explanation technique is one that assigns higher weights to edges within the ground truth subgraphs

Table 1: Explanation accuracy in terms of AUC-ROC on edges under six datasets based on GCN.

	BA-2motifs	BA-3motifs	Alkane-Carbonyl	Benzene	Fluoride-Carbonyl	MUTAG
GradCAM	$0.714_{\pm 0.000}$	$0.709_{\pm 0.000}$	$0.496_{\pm 0.000}$	$0.662_{\pm 0.000}$	$0.604_{\pm 0.000}$	$0.573_{\pm 0.000}$
GNNExplainer	$0.644_{\pm 0.007}$	$0.511_{\pm 0.002}$	$0.532_{\pm 0.008}$	$0.499_{\pm 0.004}$	$0.540_{\pm 0.002}$	$0.682_{\pm 0.009}$
PGExplainer	$0.734_{\pm 0.117}$	$0.796_{\pm 0.010}$	$0.611_{\pm 0.071}$	$0.698_{\pm 0.099}$	$0.591_{\pm 0.041}$	$0.832_{\pm 0.032}$
ReFine	$0.698_{\pm 0.001}$	$0.629_{\pm 0.005}$	$0.630_{\pm 0.007}$	$0.533_{\pm 0.011}$	$0.507_{\pm 0.003}$	$0.612_{\pm 0.004}$
MixupExplainer	$0.906_{\pm 0.059}$	$0.859_{\pm 0.019}$	$0.642_{\pm 0.033}$	$0.520_{\pm 0.043}$	$0.534_{\pm 0.012}$	$0.883_{\pm 0.103}$
ProxyExplainer	$\mathbf{0.935}_{\pm 0.008}$	$\mathbf{0.960}_{\pm 0.008}$	$\mathbf{0.806}_{\pm 0.003}$	$\mathbf{0.745}_{\pm 0.012}$	$\mathbf{0.641}_{\pm 0.070}$	$\mathbf{0.977}_{\pm 0.009}$

Table 2: MMD results between the ground truth explanations and original graphs (GT); PGExplainer explanations and original graphs (PGE); proxy graphs in our methods and original graphs (Proxy).

Metric	BA-2motifs			BA-3motifs			Alkane-Carbonyl		
	GT	PGE	Proxy	GT	PGE	Proxy	GT	PGE	Proxy
Deg.	0.846	0.785	**0.032**	0.646	0.209	**0.135**	0.663	0.525	**0.225**
Clus.	0.601	0.684	**0.319**	0.834	0.741	**0.625**	**0.051**	**0.051**	**0.051**
Spec.	0.278	0.289	**0.023**	0.259	0.108	**0.103**	0.635	0.469	**0.049**
Sum.	1.725	1.759	**0.374**	1.740	1.059	**0.864**	1.349	1.046	**0.325**
Metric	Benzene			Fluoride-Carbonyl			MUTAG		
	GT	PGE	Proxy	GT	PGE	Proxy	GT	PGE	Proxy
Deg.	0.412	0.500	**0.279**	0.546	0.478	**0.275**	0.598	0.526	**0.235**
Clus.	0.059	**0.001**	0.052	**0.049**	**0.049**	**0.049**	**0.029**	**0.029**	0.071
Spec.	0.278	0.198	**0.096**	0.385	0.349	**0.075**	0.442	0.421	**0.141**
Sum.	0.749	0.698	**0.427**	0.980	0.876	**0.399**	1.070	0.976	**0.448**

compared to those outside of them. For quantitative assessment, we utilize the AUC-ROC metric. Detailed information regarding datasets and baselines is delineated in the Appendix C.

6.2 Quantitative Evaluation (RQ1)

To answer RQ1, we compare the proposed method, ProxyExplainer, to other baselines. Each experiment was conducted 10 times using random seeds, and the average AUC scores as well as standard deviations are presented in Table 1.

The results demonstrate that ProxyExplainer provides the most accurate explanations across all datasets. Specifically, it improves the AUC scores by an average of 7.5% on synthetic datasets and 12.3% on real-world datasets. Comparisons with baseline methods highlight the advantages of our proposed explanation framework. Besides, ProxyExplainer captures underlying explanatory factors consistently across diverse datasets. For instance, MixupExplainer exhibits proficiency on the synthetic BA-2motifs dataset but performs poorly on the real-world Benzene dataset. The reason is that MixupExplainer relies on the independence assumption of explanation and non-explanation subgraphs, which may not hold in real-world datasets. In contrast, ProxyExplainer consistently demonstrates high performance across different datasets, showcasing its robustness and adaptability.

6.3 Alleviating Distribution Shifts (RQ2)

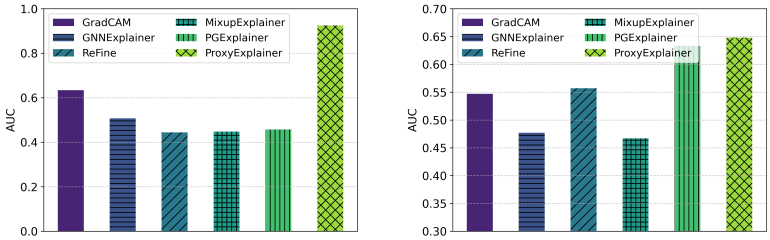
In this section, we assess ProxyExplainer's ability to generate in-distribution proxy graphs. Due to the intractable of direct computation, we follow the previous work [3] to compare distributions of multiple graph statistics, including degree distributions, clustering coefficients, and spectrum distributions, between the generated proxy graphs and original graphs. Specifically, we utilize Gaussian Earth Mover's Distance kernel when computing MMDs. Smaller MMD values indicate similar graph distributions. For comparison, we also include the Ground truth explanations and the ones generated by PGExplainer.

The results are shown in Table 2. "GT" denotes the MMDs between the ground truth explanations and original graphs. "PGE" represents the MMDs between explanations generated by PGExplainer and original graphs. "Proxy" denotes the MMDs between proxy graphs in our method and original graphs. We have the following observations, first, the MMDs between Ground truth and original graphs are usually large, verifying our motivation that a model trained on original graphs may not have correct predictions on the OOD explanation subgraphs. Second, the explanations generated by a representative work, PGExplainer, are often OOD from original graphs, indicating that the original GIB-based objective function may be sub-optimal. Third, in most cases, proxy graphs generated by our method are with smaller MMDs, demonstrating their in-distribution property.

6.4 Ablation Studies (RQ3)

In this section, we conduct ablation studies to investigate the roles of different components. Specifically, we consider the following variants of ProxyExplainer: (1) w/o G^{Δ}: in this variant, we remove the non-explanatory subgraph generator (VGAE), which is the bottom half as shown in Figure 2; (2) w/o $\mathcal{L}_{\text{KL}}$: in this variant, we remove the KL divergence from the training loss in ProxyExplainer; (3) w/o $\mathcal{L}_{\text{dist}}$: in this variant, we remove the distribution loss from ProxyExplainer. The results of the ablation study on BA-2motifs and Alkane-Carbonyl are reported in Figure 3.

Figure 3 illustrates a notable performance drop for all variants, indicating that each component contributes positively to the effectiveness of ProxyExplainer. Especially, in the real-life dataset, Alkane-Carbonyl, without the in-distribution constraint, w/o $\mathcal{L}_{\text{dist}}$ is much worse than ProxyExplainer, indicating the vital role of in-distributed proxy graphs in our framework. Extensive ablation studies on other datasets can be found in the Appendix D.2.



(a) BA-2motifs (b) Alkane-Carbonyl

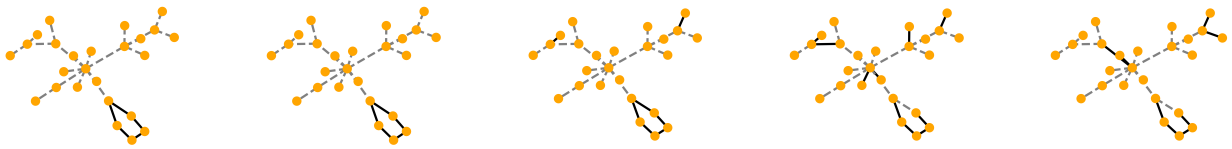
Figure 4: Explanation accuracy in terms of AUC-ROC on edges based on GIN.

(a) BA-2motifs (b) Alkane-Carbonyl

Figure 5: Parameter analysis of λ on BA-2motifs and Alkane-Carbonyl. The left side of each graph shows the explanation performance. The right side shows the Distance Analysis between h and $\tilde{h}$.

(a) BA-2motifs (b) Alkane-Carbonyl

Figure 6: Parameter analysis of the dimension of node latent embedding. The left side of each graph shows the explanation performance and the right side displays the Cosine score and Euclidean distance between h and $\tilde{h}$.

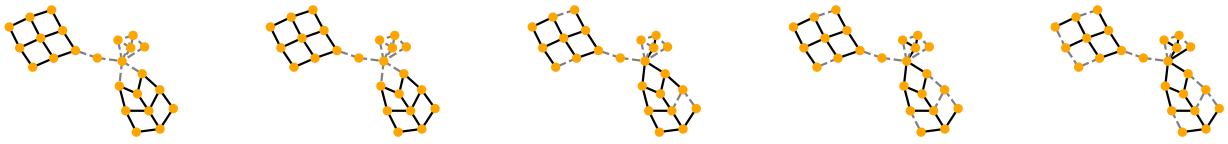


(a) Ground Truth (b) ProxyExplainer (c) MixupExplainer (d) PGExplainer (e) GNNExplainer

Figure 7: Visualization of explanation results from different explanation models on BA-2motifs. The generated explanations are highlighted by bold black edges.

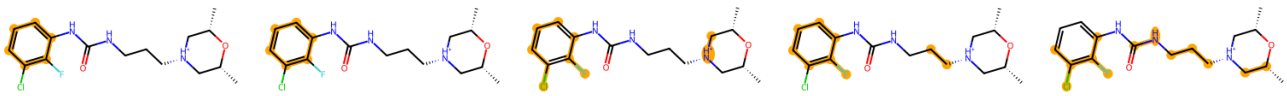
6.5 Experiment with GIN

In order to show the robustness of our ProxyExplainer in explaining different GNN models, we replace the GCN with the Graph Isomorphism Network (GIN). We use both baseline methods in the previous experiment and our method ProxyExplainer to provide explanations for the pre-trained GIN model. As seen in Figure 4, it



(a) Ground Truth (b) ProxyExplainer (c) MixupExplainer (d) PGExplainer (e) GNNExplainer

Figure 8: Visualization of explanation results from different explanation models on BA-3motifs. The generated explanations are highlighted by bold black edges.



(a) Ground Truth (b) ProxyExplainer (c) MixupExplainer (d) PGExplainer (e) GNNExplainer

Figure 9: Visualization of explanation results from different explanation models on Benzene. The generated explanations are highlighted by bold orange edges.

is noticeable that ProxyExplainer achieves the best performance on these datasets. Specifically, it improves the AUC scores of 45.9% on BA-2motifs, and 2.4% on Alkane-Carbonyl. From the analysis, we can see that our ProxyExplainer can identify accurate explanations across different datasets. The results of GIN and the extensive explanation experiment on other datasets are displayed in Appendix D.3.

6.6 Parameter Sensitive Analysis

In this section, we analyze the influence of parameters including λ, which controls the KL divergence during the proxy graph generation process, and the dimension D of node latent embedding. We vary λ from 0.01 to 1.0. For D, we vary it among {32, 64, 128, 256, 512, 1024}. We only change the value of a specific parameter while fixing the remaining parameters to their respective optimal values.

Figure 5 shows the performance of our ProxyExplainer with respect to λ on synthetic dataset BA-2motifs and one real-world dataset Alkane-Carbonyl. From Figure 5, we can see that ProxyExplainer consistently outperforms the best baseline, MixupExplainer, for $\lambda \in [0.25, 1.0]$. This indicates that ProxyExplainer is stable. Figure 6 presents how the dimension of node latent embedding affects performance in ProxyExplainer, the results show that ProxyExplainer can reach the best performance at $D = 512$. Extensive parameter sensitive analysis on other datasets can be found in the Appendix D.4.

6.7 Case Studies

In this part, we conduct case studies to qualitatively compare the performances of ProxyExplainer against others. We adopt examples from BA-2motifs, BA-3motifs and Benzene in this part. We show visualization results in Figure 7, Figure 8, and Figure 9. Explanations are highlighted with bold black in Figure 7 and Figure 8, and bold orange edges in Figure 9. From the results, our ProxyExplainer stands out by generating more compelling explanations compared to baselines. Specifically, ProxyExplainer maintains clarity without introducing irrelevant edges and exhibits more concise results compared to alternative methods. The visualized performance underscores ProxyExplainer's ability to provide meaningful and focused subgraph explanations. More visualizations on these three datasets can be found in the Appendix D.5.

7 CONCLUSION

In this paper, we systematically investigate the OOD problem in the de facto framework, GIB, for learning explanations in GNNs, which is highly overlooked in the literature. To address this issue, we extend the GIB by innovatively including in-distributed proxy graphs. On top of that, we derive a tractable objective function for practical implementations. We further present a new explanation framework that utilizes two graph auto-encoders to generate proxy graphs. We conduct comprehensive empirical studies on both benchmark synthetic datasets and real-life datasets to demonstrate the effectiveness of our method in alleviating the OOD problem and achieving high-quality explanations. There are several topics we will investigate in the future. First, the OOD problem also exists in obtaining model-level explanations and counterfactual explanations. We will extend our method to these research problems. Second, we will also analyze explainable learning methods with proxies in other data structures, such as image, language, and time series.

REFERENCES

[1] Dor Bank, Noam Koenigstein, and Raja Giryes. 2023. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook* (2023), 353–374.

[2] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2019. Explaining Image Classifiers by Counterfactual Generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=B1MXz20cYQ

[3] Jialin Chen, Shirley Wu, Abhijit Gupta, and Zhitao Ying. 2023. D4Explainer: In-distribution Explanations of Graph Neural Network via Discrete Denoising Diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=GJtP1ZEzua

[4] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry* 34, 2 (1991), 786–797.

[5] Paul Erdős, Alfréd Rényi, et al. 1960. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci* 5, 1 (1960), 17–60.

[6] Junfeng Fang, Wei Liu, Yuan Gao, Zemin Liu, An Zhang, Xiang Wang, and Xiangnan He. 2023. Evaluating Post-hoc Explanations for Graph Neural Networks via Robustness Analysis. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[7] Junfeng Fang, Wei Liu, An Zhang, Xiang Wang, Xiangnan He, Kun Wang, and Tat-Seng Chua. 2023. On Regularization for Explaining Graph Neural Networks: An Information Theory Perspective. https://openreview.net/forum?id=5rX7M4wa2R_

[8] Junfeng Fang, Wei Liu, An Zhang, Xiang Wang, Xiangnan He, Kun Wang, and Tat-Seng Chua. 2023. On Regularization for Explaining Graph Neural Networks: An Information Theory Perspective. https://openreview.net/forum?id=5rX7M4wa2R_

[9] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. 2023. Cooperative Explanations of Graph Neural Networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 616–624.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[11] Rundong Huang, Farhad Shirani, and Dongsheng Luo. 2024. Factorized Explainer for Graph Neural Networks. In *Proceedings of the AAAI conference on artificial intelligence*.

[12] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rkE3y85ee

[13] Jeroen Kazius, Ross McGuire, and Roberta Bursi. 2005. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry* 48, 1 (2005), 312–320.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJU4ayYgl

[16] Wenqian Li, Yinchuan Li, Zhigang Li, Jianye Hao, and Yan Pang. 2023. DAG Matters! GFlowNets Enhanced Explainer for Graph Neural Networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

[17] Yiqiao Li, Jianlong Zhou, Sunny Verma, and Fang Chen. 2022. A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint arXiv:2207.12599* (2022).

[18] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems* 33 (2020), 19620–19631.

[19] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *International Conference on Learning Representations*. https://openreview.net/forum?id=S1jE5L5gl

[20] Siqi Miao, Mia Liu, and Pan Li. 2022. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International Conference on Machine Learning*. PMLR, 15524–15543.

[21] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. 2019. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10772–10781.

[22] Luyu Qiu, Yi Yang, Caleb Chen Cao, Yueyuan Zheng, Hilary Ngai, Janet Hsiao, and Lei Chen. 2022. Generating perturbation-based explanations with robustness to out-of-distribution data. In *Proceedings of the ACM Web Conference 2022*. 3594–3605.

[23] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. 2020. Evaluating attribution for graph neural networks. *Advances in neural information processing systems* 33 (2020), 5898–5910.

[24] Benjamín Sánchez-Lengeling, Jennifer N. Wei, Brian K. Lee, Emily Reif, Peter Wang, Wesley Wei Qian, Kevin McCloskey, Lucy J. Colwell, and Alexander B. Wiltschko. 2020. Evaluating Attribution for Graph Neural Networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[25] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[26] Teague Sterling and John J. Irwin. 2015. ZINC 15 - Ligand Discovery for Everyone. *J. Chem. Inf. Model.* 55, 11 (2015), 2324–2337.

[27] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJXMpikCZ

[29] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1225–1234.

[30] Xiang Wang, Ying-Xin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. 2021. Towards Multi-Grained Explainability for Graph Neural Networks. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. 18446–18458.

[31] Bingzhe Wu, Jintang Li, Junchi Yu, Yatao Bian, Hengtong Zhang, CHaochao Chen, Chengbin Hou, Guoji Fu, Liang Chen, Tingyang Xu, et al. 2022. A survey of trustworthy graph learning: Reliability, explainability, and privacy protection. *arXiv preprint arXiv:2205.10014* (2022).

[32] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. 2020. Graph information bottleneck. *Advances in Neural Information Processing Systems* 33 (2020), 20437–20448.

[33] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems* 32 (2019).

[34] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[35] He Zhang, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei. 2022. Trustworthy graph neural networks: Aspects, methods and trends. *arXiv preprint arXiv:2205.07424* (2022).

[36] Jiaxing Zhang, Zhuomin Chen, Dongsheng Luo, Hua Wei, et al. 2023. RegExplainer: Generating Explanations for Graph Neural Networks in Regression Tasks. In *The Second Learning on Graphs Conference*.

[37] Jiaxing Zhang, Dongsheng Luo, and Hua Wei. 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.

[38] Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. 2022. GStarX: Explaining Graph Neural Networks with Structure-Aware Cooperative Games. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

[39] Xu Zheng, Farhad Shirani, Tianchun Wang, Wei Cheng, Zhuomin Chen, Haifeng Chen, Hua Wei, and Dongsheng Luo. 2023. Towards Robust Fidelity for Evaluating Explainability of Graph Neural Networks. *arXiv preprint arXiv:2310.01820* (2023).

A NOTATIONS

In Table 3, we summarized the main notations we used and their descriptions in this paper.

Table 3: Symbols and their descriptions.

Symbols	Descriptions
$\mathcal{G}$	A set of graphs
$G, \mathcal{V}, \mathcal{E}$	Graph instance, node set, edge set
v_i	The i-th node
$\boldsymbol{X}$	Node feature matrix
$\boldsymbol{A}$	Adjacency matrix
$\boldsymbol{Z}$	Node representation matrix
Y	Label of graph G
$\boldsymbol{C}$	A set of labels
G^*	Optimal explanatory subgraph
$\mathcal{G}^*$	A set of G^*
G'	Candidate explanatory subgraph
G^Δ	Non-explanatory graph
$\tilde{G}$	Proxy graph of G' with a fixed distribution
$\boldsymbol{h}$, $\boldsymbol{h}'$, $\tilde{\boldsymbol{h}}$	Graph embedding
d	Dimension of node feature
D	Dimension of node latent embedding
$f(\cdot)$	To-be-explained GNN model
Ψ_ψ	Explanation function
ψ	Parameter of the explanation function
$t_\psi(\cdot)$	Perturbation function
$P_{\mathcal{G}}$	Distribution of original training graphs
$P_{\mathcal{G}'}$	Distribution of explanation subgraphs
$Q_{\boldsymbol{\phi}}$	Parameterized function of $P(\tilde{G}\|G')$
$\boldsymbol{\phi}$	Model parameters of $Q_{\boldsymbol{\phi}}$
$\boldsymbol{\phi}^*$	Optimal $\boldsymbol{\phi}$
α	Balance parameter between $I(G, G')$ and $I(Y, G')$
$\bar{\mathcal{E}}$	The set of node pairs that are unconnected in G
$\tilde{p}_{uv}$	Probability of node pair (u, v) in $\tilde{G}$
β	A hyper-parameter to get a trade-off between connected and unconnected node pairs
$f_{\text{enc}}(\cdot)$	The front part of GNNs that learns node representations
$f_{\text{cls}}(\cdot)$	The back part of GNNs that predicts graph labels based on node embeddings
$\sigma(\cdot)$	Sigmoid function
τ	Temperature hyper-parameter for approximation
λ	A hyper-parameter in Proxy loss function
$\mathcal{L}_{\text{dist}}$	Distribution loss between $\tilde{G}$ and G
$\mathcal{L}_{\text{KL}}$	KL divergence between distribution of $\boldsymbol{Z}^\Delta$ and its prior
$\mathcal{L}_{\text{proxy}}$	Proxy loss
$\mathcal{L}_{\text{exp}}$	Explainer loss

B ALGORITHM

We take as input a set of graphs $\mathcal{G} = \{G_i\}_{i=0}^N$. For each graph G_i, we use an explainer model to identify a subgraph G_i' as the explanation and then compute the non-explanatory graph G_i^Δ, which is obtained by removing edges in G_i that exist in G_i'. We use GAE to reconstruct a subgraph, denoted by $\tilde{G}_i'$ from the subgraph G_i' and another VGAE to generate a new subgraph, denoted by $\tilde{G}_i^\Delta$, from the non-explainable subgraph G_i^Δ. The proxy graph, $\tilde{G}_i$, is obtained by combining them, whose adjacency matrix can be denoted by $\tilde{\boldsymbol{A}}_i = \tilde{\boldsymbol{A}}_i' + \tilde{\boldsymbol{A}}_i^\Delta$. Here $\tilde{\boldsymbol{A}}_i'$ and $\tilde{\boldsymbol{A}}_i^\Delta$ are adjacency matrices of $\tilde{G}_i'$ and $\tilde{G}_i^\Delta$, respectively. We alternatively train the explainer model and proxy graph generator as shown in Algorithm 1.

Algorithm 1 Algorithm of ProxyExplainer

Input: A set of graphs $\mathcal{G} = \{G_i\}_{i=0}^{N}$, with each $G_i = (\mathcal{V}_i, \mathcal{E}_i, \boldsymbol{X}_i)$,a pretrain to-be-explained model $f(\cdot)$, hyper parameters α, λ, M, epochs E.
Initialize an explainer function $\Psi_\psi(\cdot)$
epoch $\leftarrow 0$
while epoch $< E$ **do**
  for $G_i \in \mathcal{G}$ **do**
    $G_i' \leftarrow \Psi_\psi(G_i)$
    $G_i^\Delta \leftarrow G_i - G_i'$
    Compute $\tilde{\boldsymbol{A}}_i'$ with equation 14
    Compute $\boldsymbol{Z}^\Delta$ and $\tilde{\boldsymbol{A}}_i^\Delta$ with equation 16
    compute proxy loss $\mathcal{L}_{\text{proxy}}$ with equation 18
    Update parameters in proxy graph generator with backpropagation
    if epoch$\%M == 0$ **then**
      compute explainer loss $\mathcal{L}_{\text{exp}}$
      Update parameters in the explainer with backpropagation.
    end if
  end for
  epoch $\leftarrow$ epoch + 1
end while

C FULL EXPERIMENTAL SETUPS

C.1 Datasets

BA-2motifs [18]. The BA-2motifs dataset consists 1,000 synthetic graphs, each derived from a basic Barabasi-Albert (BA) model. The dataset is divided into two categories: one part of the graphs add patterns that mimic the structure of a house, and the remaining integrate five-node cyclic patterns. The classification of these graphs depends on the specific patterns.
BA-3motifs [3]. BA-3motifs is an extended dataset inspired by the BA-2motifs and contains 3,000 synthetic graphs. Each base graph is accompanied by one of three different patterns: house, cycle, or grid.
Alkane-Carbonyl [24]. The Alkane-Carbonyl dataset consists of a total of 4,326 molecule graphs that are categorized into two different classes. Positive samples refer to molecules that have both alkane and carbonyl functional groups. The ground-truth explanation includes alkane and carbonyl functional groups in a given molecule.
Benzene [24]. Benzene consists 12,000 molecular graphs from the ZINC15[26] database, which can be classified into two classes. The main goal is to determine if a Benzene ring is exited in each molecule. In cases where multiple Benzene rings are present, each Benzene ring is treated as a distinct explanation.
Fluoride-Carbonyl [24]. The Fluoride-Carbonyl dataset has 8,671 molecular graphs. The ground-truth explanation is based on the particular combination of fluoride atoms and carbonyl functional groups present in each molecule.
MUTAG [13]. MUTAG dataset includes 4,337 molecular graphs, each of them is classified into two groups based on its mutagenic effect on the Gram-negative bacterium S. Typhimurium. This classification comes from several specific virulence gland groups with mutagenicity in molecular mapping by Kazius et al.[13].

C.2 Baselines

To evaluate our model, we well-train the GCN model in to ensure that it takes good performance in graph classification tasks. The results are displayed in Table 5. We utilize a well-established explanation method as a benchmark baseline PGExplainer. Furthermore, for a comprehensive comparison, we incorporate various post-hoc explanation methods, including GradCAM, GNNExplainer, PGExplainer, ReFine, and MixupExplainer.

- **GradCAM [21].** This method utilizes gradients as a weighting mechanism to merge various feature maps. It operates on heuristic assumptions and cannot elucidate node classification models.
- **GNNExplainer [33].** It learns soft masks for edges and node features, and aims to elucidate predictions through mask optimization. GNNExplainer integrates these masks with the original graph through element-wise multiplications. The masks are optimized by enhancing the mutual information between the original graph and the modified graph prediction results.
- **PGExplainer [18].** This method extends the idea of GNNExplainer by assuming that the graph is a random Gilbert graph. PGExplainer generates each edge embedding by combining the embeddings of its constituent nodes, then uses these edge embeddings to determine

Table 4: Statistics of molecule datasets for graph classifcation with ground-truth explanations.

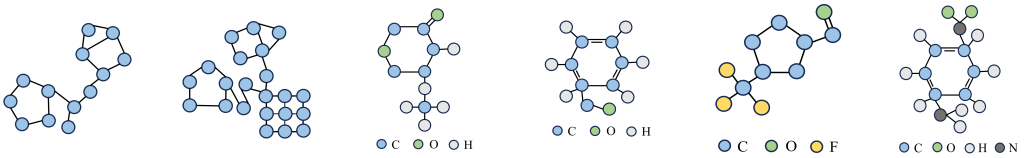
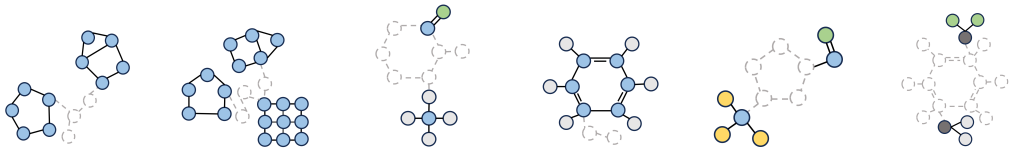
	BA-2motifs	BA-3motifs	Alkane-Carbonyl	Benzene	Fluoride-Carbonyl	MUTAG
Graphs	1,000	3,000	4,326	12,000	8,671	4,337
Average nodes	25.00	21.92	21.13	20.58	21.36	29.15
Average edges	25.48	29.51	44.95	43.65	45.37	60.83
Original graph						
Features	None	None	Atom	Atom	Atom	Atom
Ground truth explanation						
	House, cycle	House,cycle, grid	Alkane,C=O	Benzene Ring	F^-,C=O	NH_2, NO_2

Table 5: The graph-classification task performances of the GCN model

	BA-2motifs	BA-3motifs	Alkane-Carbonyl	Benzene	Fluoride-Carbonyl	MUTAG
Train Acc	0.999	0.997	0.979	0.930	0.951	0.850
Val Acc	1.0	0.997	0.986	0.923	0.956	0.834
Test Acc	1.0	0.977	0.975	0.915	0.951	0.804

a Bernoulli distribution to indicate whether to mask an edge or not, and utilizes a Gumbel-Softmax approach to model the Bernoulli distribution for end-to-end training.

- **ReFine [30].** ReFine identifies the edge probabilities for the entire category by maximizing the mutual information and contrastive loss between categories. In fine-tuning, it uses the edge probabilities from the previous stage to sample edges, and find explanations that maximize mutual information for specific instances.
- **MixupExplainer [37].** This method combines original explanatory subgraphs with randomly sampled, label-independent base graphs in a non-parametric way to mitigate the common OOD issue which found in previous methods.

D EXTENSIVE EXPERIMENTS

D.1 Extensive Distribution Analysis

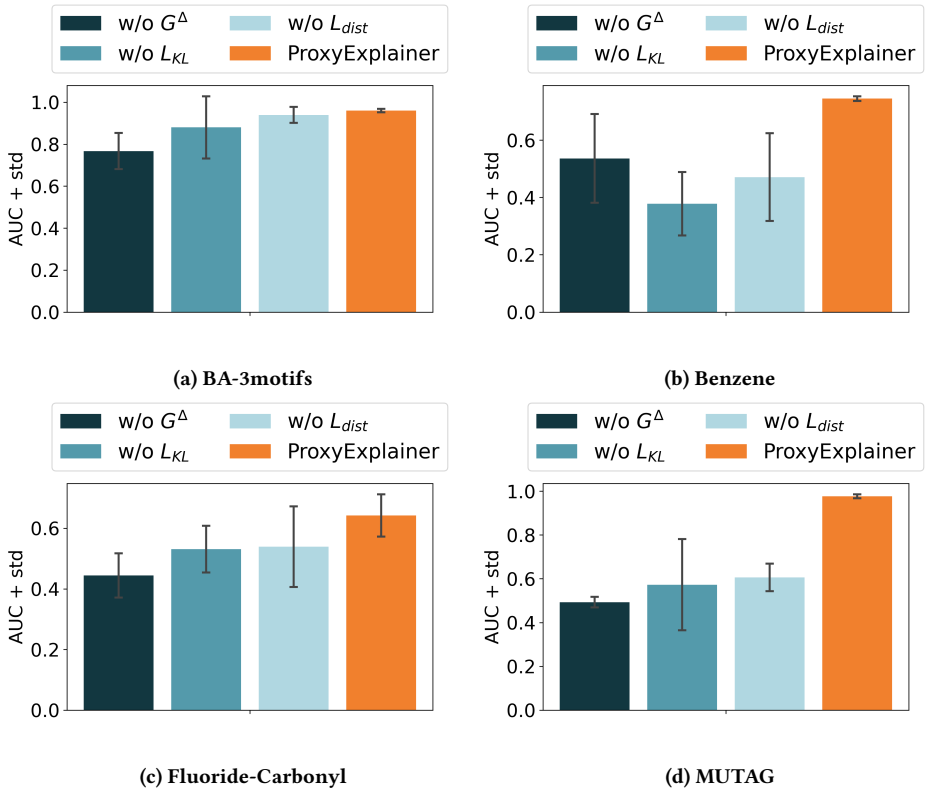
As shown in previous work [37], another way to approximate the distribution differences of graphs is to compare their vector embeddings. Here, we adopt the same setting and use Cosine similarity and Euclidean distance to intuitively approximate the similarities between graph distributions. Table 6 presents the computed Cosine similarity and Euclidean distance between the distribution embeddings of the original graph $\boldsymbol{h}$, the ground truth explanation subgraph $\boldsymbol{h}'$, and the generated proxy graph $\tilde{\boldsymbol{h}}$. Notably, our proxy graph embedding $\tilde{\boldsymbol{h}}$ exhibits higher Cosine similarity scores and lower Euclidean distance with the original graph embedding $\boldsymbol{h}$ compared to the ground truth explanation embedding $\boldsymbol{h}'$. We observe an average improvement of 17.9% in Cosine similarity and 33.6% in Euclidean distance. Particularly in the BA-2motifs dataset, there is a significant improvement of 60.4% in Cosine similarity and 51.8% in Euclidean distance. These findings underscore the effectiveness of our ProxyExplainer method in mitigating distribution shifts caused by induction bias in the to-be-explained GNN model $f(\cdot)$, thereby enhancing explanation performance.

Table 6: The Cosine score and Euclidean distance between the distribution embedding of the original graph h, explanation subgraph h', and our proxy graph $\tilde{h}$ on different datasets.

	BA-2motifs	BA-3motifs	Alkane-Carbonyl	Benzene	Fluoride-Carbonyl	MUTAG
Avg. Cosine(h, h') ↑	0.571	0.686	0.879	0.859	0.906	0.883
Avg. Cosine(h, $\tilde{h}$) ↑	**0.916**	**0.918**	**0.938**	**0.905**	**0.908**	**0.985**
Avg. Euclidean(h, h') ↓	1.210	1.199	0.912	0.893	0.805	0.975
Avg. Euclidean(h, $\tilde{h}$) ↓	**0.583**	**0.613**	**0.719**	**0.767**	**0.779**	**0.368**

D.2 Extensive Ablation Study

In Figure 10, we conduct a comprehensive ablation study across other four datasets(BA-3motifs, Benzene, Fluoride-Carbonyl, MUTAG) to examine the impact of different components. The findings illustrate the effectiveness of these components and their positive contribution to our ProxyExplainer.



(a) BA-3motifs (b) Benzene

(c) Fluoride-Carbonyl (d) MUTAG

Figure 10: Ablation study on other four datasets.

D.3 Extensive experiment with GIN

We show more experiment results with GIN, we first pre-train GIN on two synthetic datasets (BA-2motifs and BA-3motifs) and two real-world datasets (Alkane-Carbonyl and Fluoride-Carbonyl) to ensure that GIN has the ability to classify graphs accurately. The results are displayed in Figure 11. Figure 12 shows that our ProxyExplainer improves the AUC scores of 6.1% on BA-3motifs, and 12.5% on Fluoride-Carbonyl.

D.4 Extensive Parameter Sensitive Analysis

Figure 13 shows the performance of our ProxyExplainer with respect to λ on BA-3motifs and MUTAG.

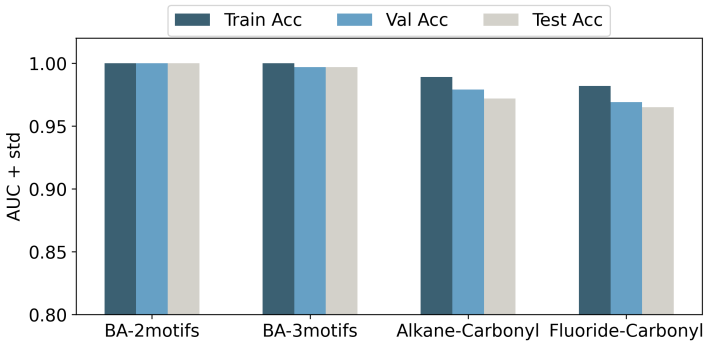
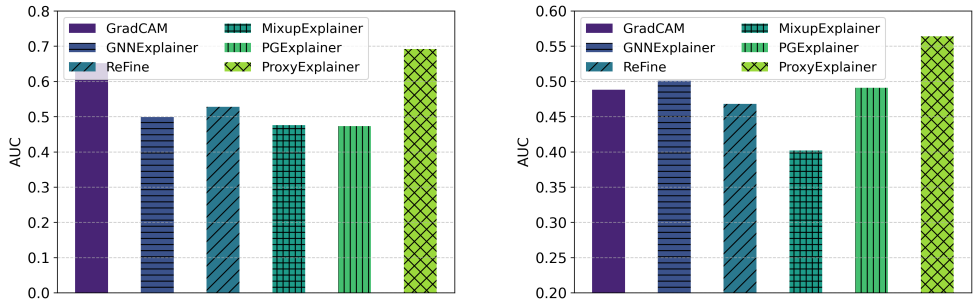
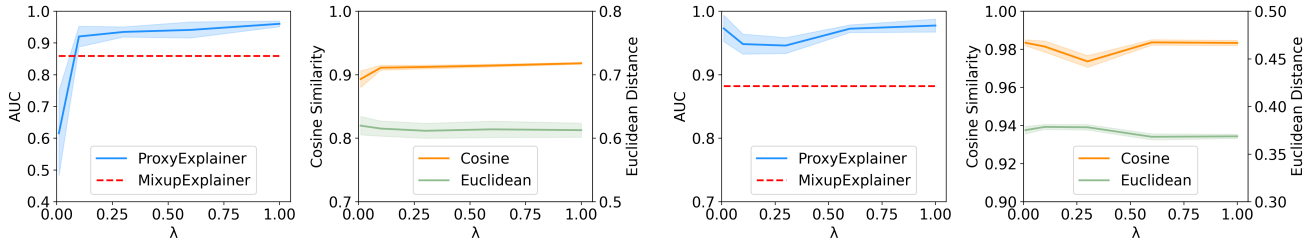


Figure 11: The graph-classification task performances of the GIN model.



(a) BA-3motifs **(b) Fluoride-Carbonyl**

Figure 12: Explanation accuracy in terms of AUC-ROC on edges based on GIN.

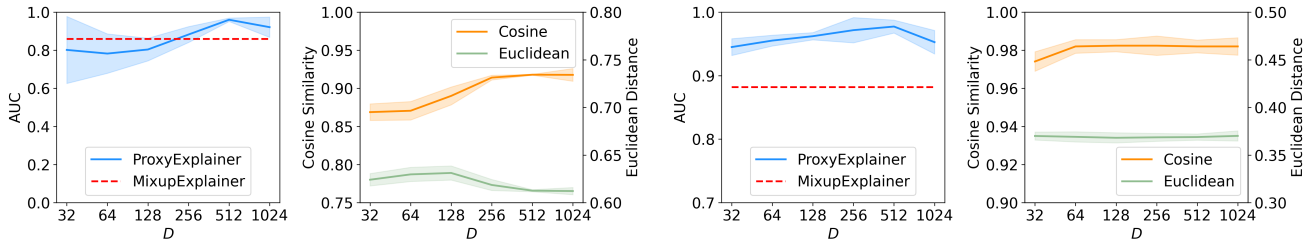


(a) BA-3motifs **(b) MUTAG**

Figure 13: Parameter analysis of λ on BA-3motifs) and MUTAG datasets. The left side of each graph shows the explanation performance. The right side shows the Distance Analysis between h and $\tilde{h}$.

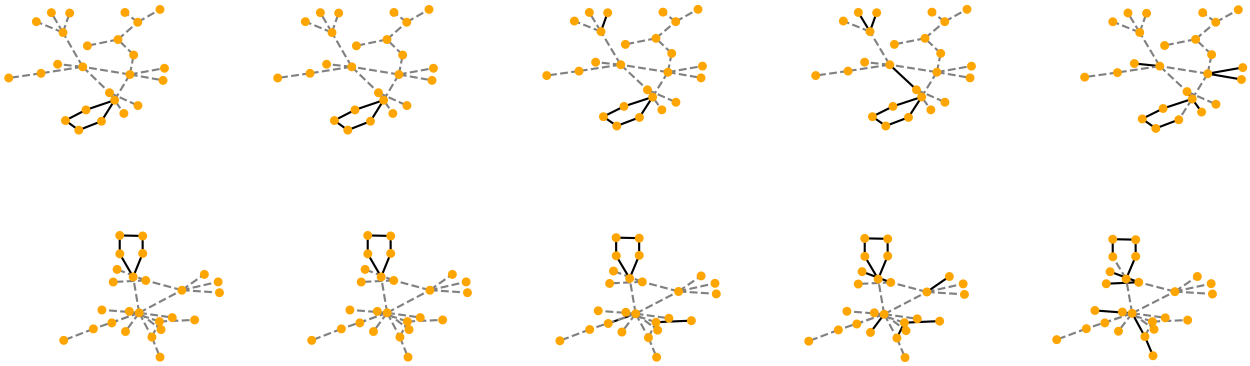
D.5 Extensive Case Study

We show more visualization examples of explanation graphs on BA-2motifs, BA-3motifs and Benzene in Figure 15, Figure 16, and Figure 17, respectively.



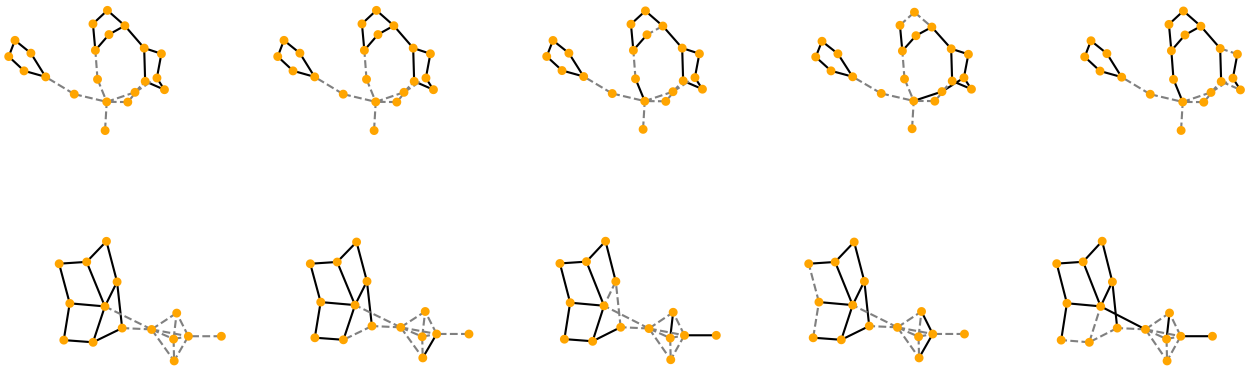
(a) BA-3motifs **(b) MUTAG**

Figure 14: Parameter analysis of the dimension of node latent embedding. The left side of each graph shows the explanation performance and the right side displays the Cosine score and Euclidean distance between h and $\tilde{h}$.



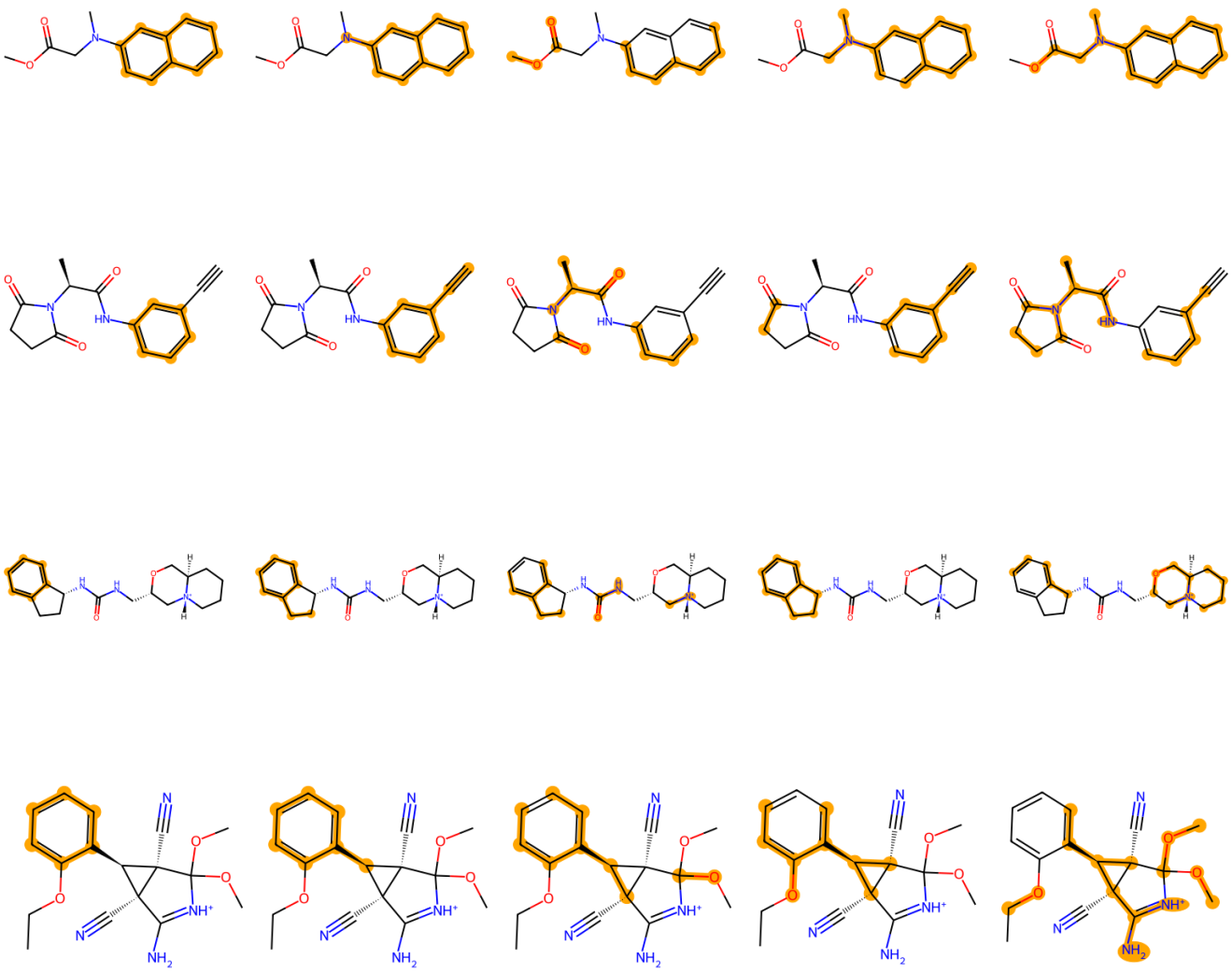
(a) Ground Truth **(b) ProxyExplainer** **(c) MixupExplainer** **(d) PGExplainer** **(e) GNNExplainer**

Figure 15: Visualization of explanation on BA-2motifs.



(a) Ground Truth **(b) ProxyExplainer** **(c) MixupExplainer** **(d) PGExplainer** **(e) GNNExplainer**

Figure 16: Visualization of explanation on BA-3motifs.



(a) Ground Truth (b) ProxyExplainer (c) MixupExplainer (d) PGExplainer (e) GNNExplainer

Figure 17: Visualization of explanation on Benzene.