Robust Surrogate Modeling for Explainable Graph Neural Networks

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

Jingchao Ni
University of Houston
Houston, TX, USA
jni7@uh.edu

Hojat Allah Salehi
Florida International University
Miami, FL, USA
hsalehi@fiu.edu

Xu Zheng
Florida International University
Miami, FL, USA
xzhen019@fiu.edu

Dongsheng Luo*
Singapore Management University
Singapore
luodongsheng01@gmail.com

Abstract

Graph Neural Networks (GNNs) have achieved remarkable performance in a wide range of graph-related learning tasks. However, explaining their predictions remains a challenging problem, especially due to the mismatch between the graphs used during training and those encountered during explanation. Most existing methods optimize soft edge masks on weighted graphs to highlight important substructures, but these graphs differ from the unweighted graphs on which GNNs are trained. This distributional shift leads to unreliable gradients and degraded explanation quality, especially when generating small, sparse subgraphs. To address this issue, we propose explaining a robust surrogate whose explanations provide a faithful proxy for those of the original model. Specifically, we train this surrogate model through an iterative re-weighting strategy that alternates between identifying explanation subgraphs and adapting the model to the resulting weighted graph distribution, supporting stable explanation refinement while preserving the validity of the soft-mask relaxation. We evaluate our method on multiple benchmark datasets using different GNN backbones and explanation methods. Experimental results show that our method consistently improves explanation quality and can be flexibly integrated with different architectures.

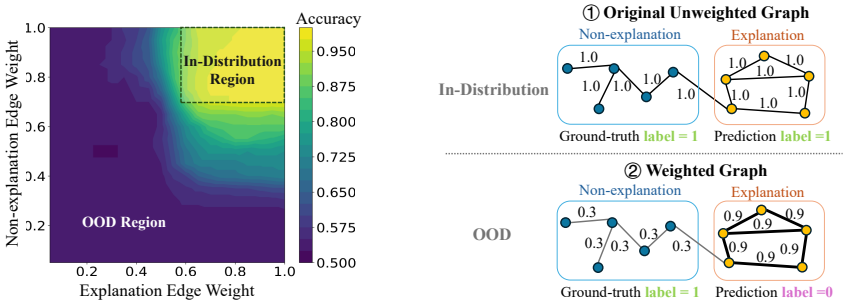
1 Introduction

Graph Neural Networks (GNNs) have become a powerful tool for learning from graph-structured data, with applications in molecular property prediction [28], social networks [5], and knowledge graphs [25]. However, they often behave as black-box models, posing challenges for explainability in high-stakes domains such as healthcare [4] and finance [21], where transparency is essential.

These concerns have led to the emergence of a number of post-hoc explanation methods to explain predictions of GNNs. The goal of these methods is to identify a small, label-preserving subgraph that retains the original model prediction $f(G)$ [31, 18, 32, 27]. Due to the combinatorial search space, most existing approaches rely on continuous relaxation. They introduce a learnable mask M, applied to the input graph's adjacency matrix, and optimize M by minimizing the discrepancy between the original prediction $f(G)$ and the masked prediction $f(M \odot G)$. The resulting soft masks are then binarized to yield the final discrete explanation subgraph [33, 24].

*Corresponding author. This work was primarily conducted while Dongsheng Luo was at Florida International University.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) GCN Accuracy Heat Map

(b) GCN Prediction Examples

Figure 1: (a) Classification accuracy of a GCN trained on unweighted graphs, evaluated on weighted graphs from the BA-2motif dataset with varying edge weights for motif and BA-graph edges. High accuracy is maintained only in the "in-distribution" region (both edge types near weight 1.0), but degrades significantly under distributional shift. (b) The original unweighted graph (example ①) is correctly classified, but reweighting motif edges to 0.9 and BA-graph edges to 0.3 (example ②) causes misclassification.

A key assumption underlying this post-hoc explanation framework is that the GNN model f, trained exclusively on unweighted graphs, will still produce meaningful predictions and reliable gradients when fed the continuously weighted graphs $M \odot G$ during mask optimization. This assumption is problematic as it introduces a significant distributional shift: the model is forced to process graphs with continuous edge weights, which deviate substantially from the binary-weighted graphs in its training distribution. This training scenario reflects the reality of many domains, such as molecular property prediction and social network analysis, where graph connections are often treated as binary (present or absent) due to data constraints or model simplification.

This distributional shift is particularly severe due to the sparsity objective necessary for obtaining faithful explanations. Meaningful explanations typically correspond to small, concise subgraphs, which requires the optimization to push most mask values in M toward zero to satisfy size constraints. This process results in highly sparse, weighted graphs that bear little resemblance to the model's training data. Operating in this Out-Of-Distribution (OOD) regime can cause unstable model behavior and yield unreliable gradient signals, ultimately culminating in suboptimal or unfaithful explanations. To the best of our knowledge, this explanation-induced distribution shift has not been explicitly identified in prior work on graph explainability.

To empirically validate the impact of distributional shifts, we conduct an experiment on the BA-2motifs dataset, which contains graphs with either house or cycle motifs attached to Barabási–Albert (BA) base graphs [2]. As shown in Figure 1(a), the classification accuracy of a GCN trained on unweighted graphs degrades significantly when evaluated on weighted graphs where edge weights deviate from the training distribution. High accuracy is maintained only in the upper-right "in-distribution" region where both motif and base edge weights are near 1.0. Figure 1(b) illustrates a concrete failure case: the original unweighted graph is correctly classified (Example ①), but reweighting edges to reflect a typical explanation configuration (Example ②) leads to misclassification. This outcome demonstrates that model predictions fail to remain consistent under shifted conditions. If the importance of explanatory edges cannot dominate predictions in this OOD regime, the robustness and trustworthiness of explanations generated by existing methods become highly questionable.

To address this critical limitation, inspired by surrogate explainable AI literature [22, 11], we propose explaining a **robust surrogate model** trained to handle continuous edge weights, whose explanations serve as a faithful proxy for those of the original model. We introduce a novel iterative framework called graph explanation with stochastic re-weighting (STORE) to build and explain this model. Specifically, our method proceeds through alternating phases of *subgraph identification* and *model adaptation*. Initially, we extract a relatively large explanation subgraph where soft mask optimization remains reliable. We then assign different weights to edges within and outside this subgraph and construct surrogate models by adding these weighted graphs to the training datasets. This step enables surrogate models to maintain predictive consistency under the specific weight distribution that resembles soft-masked graphs, ensuring the validity of subsequent mask optimization. The process iterates with progressively smaller subgraphs, creating a "ladder" in which each step maintains the

validity of the soft relaxation assumption. This framework offers several key advantages. First, it directly mitigates the distributional shift problem by ensuring that surrogate models can process weighted graphs at each stage of explanation refinement. Second, the progressive refinement from larger to smaller subgraphs provides a stable optimization trajectory, avoiding the severe distributional shifts that occur when directly searching for minimal explanations. Third, it naturally produces explanations at multiple levels of granularity, offering practitioners flexibility in balancing explanation size with fidelity. We summarize our contributions as follows:

- We identify and formally characterize a critical flaw in post-hoc GNN explainability: the distributional shift between unweighted training graphs and the weighted graphs used during mask optimization, which undermines explanation reliability.
- We propose a robust surrogate framework designed to resolve this OOD problem. By constructing a surrogate model capable of processing continuous edge weights, our framework ensures reliable gradient estimation and preserves the validity of soft mask optimization.
- We instantiate this framework through a novel iterative stochastic re-weighting strategy. This approach progressively aligns the surrogate model's training distribution with the explanation setting by assigning importance-aware weights to explanatory and non-explanatory edges.
- Extensive experiments on five benchmark datasets demonstrate that our method consistently improves explanation quality with different GNNs and explanation methods, while incurring only modest computational overhead.

2 Preliminaries and the OOD Problem

We represent a graph as $G = (\mathcal{V}, \mathcal{E}, \boldsymbol{X}, \boldsymbol{A})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes with $n = |\mathcal{V}|$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ is the node feature matrix where the i-th row $\boldsymbol{X}_i$ represents the d-dimensional feature of node v_i, and $\boldsymbol{A} \in \{0,1\}^{n \times n}$ is the binary adjacency matrix such that $A_{i,j} = 1$ if $(v_i, v_j) \in \mathcal{E}$. Each graph is associated with a label $y \in \mathcal{Y}$.

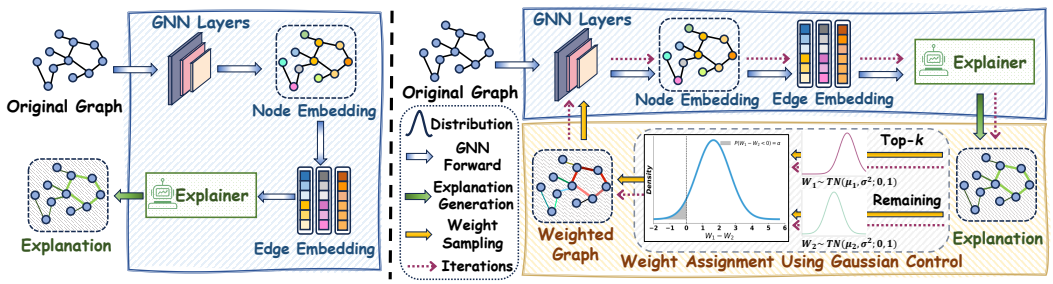
Let $f : \mathcal{G} \mapsto \{1, 2, \cdots, |\mathcal{Y}|\}$ denote a to-be-explained GNN model that maps an input graph $G \in \mathcal{G}$ to its predicted class. This work focuses on *post-hoc, instance-level* explanation methods, which treat the GNN as a black box and aim to provide task-agnostic explanations for its predictions [31, 18, 34, 12]. Given a GNN model f and the input graph G, the purpose of the explainer is to find a subgraph $G_{exp} \subseteq G$ containing relevant information for the model's prediction while excluding unnecessary structure. We evaluate an explanation based on two properties. Sufficiency measures how closely the predictive distribution conditioned on G_{exp} agrees with that conditioned on the full graph G, expressed as $d_{TV}(P_{\widehat{Y}|G}, P_{\widehat{Y}|G_{exp} \subseteq G}) \approx 0$. Minimality encourages a compact explanation by constraining the number of selected edges. Following the sufficiency–minimality formulation introduced in [9], our objective is to derive an explainer $\Psi(\cdot)$, with $G_{exp} = \Psi(G)$, to minimize the discrepancy of the two predictive distributions based on the expectation of an edge budget s:

$$\Psi^* = \arg\min_{\Psi} \ \mathbb{E}_G\Big[d_{TV}\Big(f(G), P_{\widehat{Y}|G_{exp} \subseteq G}\Big)\Big], \quad \text{s.t.} \quad \mathbb{E}_G[|\mathcal{E}_{exp}|] \leq s. \tag{1}$$

Here, $f(G)$ represents the predictive distribution $P_{\widehat{Y}|G}$. The remaining conditional distribution, $P_{\widehat{Y}|G_{exp} \subseteq G}$, is generally difficult to evaluate. This makes direct optimization impractical and motivates the approximations used in existing explanation methods.

To practically solve the optimization problem in equation 1, existing methods learn a parameterized explainer function, denoted as Ψ_ψ, where ψ represents the learnable parameters. For a given graph G, this function generates an intermediate, weighted subgraph, $\Psi_\psi(G)$, where edge weights signify their importance to the model's prediction. The final discrete explanation, G_{exp}, is then typically derived by binarizing this weighted subgraph.

The standard approach in the literature is to optimize the parameters ψ by computing the sufficiency loss on this intermediate weighted subgraph. The formal objective from equation 1 is thus approximated by an objective over ψ. The sufficiency term is replaced by a proxy like the Cross-Entropy (CE) loss, and the minimality constraint is handled by a regularization term that encourages sparsity in the generated subgraph, such as the norm of its weighted adjacency matrix, $||\Psi_\psi(G)||$. This leads



(a) Vanilla Explainer (b) Our Explainer Framework

Figure 2: Overview of the vanilla explainer and STORE. (a) Vanilla explainer. (b) STORE first inputs the original graph into a GNN to obtain node and edge embeddings. Then we use an explainer to generate a weighted explanation subgraph. Based on this explanation, we apply a Gaussian-based sampling strategy to assign edge weights and construct a weighted graph. This graph is used to retrain a surrogate GNN to improve explanation accuracy.

to the following standard optimization objective:

$$\psi^* = \arg\min_{\psi} \left(\sum_{(G,y)\in\mathcal{G}} \mathrm{CE}\big(f(G), f(\Psi_\psi(G))\big) + \lambda \|\Psi_\psi(G)\| \right). \tag{2}$$

However, we claim this standard formulation contains a critical flaw. The GNN model f is typically trained on graphs with unweighted edges (i.e., binary adjacency matrices). The intermediate subgraph, $\Psi_\psi(G)$, is inherently weighted. Consequently, evaluating the original model f on this subgraph constitutes an OOD problem. The gradients produced in this regime, which are used to update the explainer's parameters ψ, may be unstable and unreliable.

3 Robust Surrogate Framework

In this section, we present our graph explanation with stochastic re-weighting (STORE) framework. An overview is shown in Figure 2(b). Our approach addresses the distributional shift problem by iteratively building a **robust surrogate model** through a process that alternates between *subgraph identification* and *model adaptation*. Given a pre-trained GNN model, we first extract a relatively large explanation subgraph where soft mask optimization remains reliable. We then construct weighted training graphs by assigning different weights to edges within and outside the identified subgraph, and construct a surrogate GNN on these weighted graphs to ensure predictive consistency under the corresponding weight distribution. This process iterates for L rounds, progressively refining explanations from larger to smaller subgraphs. At each iteration, the explainer identifies important substructures, while the surrogate GNN maintains consistent predictions on weighted graphs, thereby preserving the validity of continuous relaxation throughout the explanation refinement process.

3.1 Explanation-Guided Stochastic Re-Weighting

Our iterative process begins by generating a foundational set of explanations to guide model adaptation. To accomplish this, we initialize $f^{(0)}$ with f, where f is the pre-trained GNN model to be explained, and train an initial explainer, $\Psi_\psi^{(0)}$ on $f^{(0)}$ using the standard objective from equation 2. This process yields a continuous score for each edge, representing its learned importance. From these scores, we derive an initial explanation subgraph by selecting the top-k edges. Following a common strategy, we use a relatively large 'k' in this first step to create an inclusive set of potentially important edges.

This initial subgraph provides the necessary input for our re-weighting strategy. However, since the explanation is generated under the previously discussed OOD condition, the core of our method is to address this distributional shift directly. We achieve this by training a surrogate GNN model on weighted graphs constructed through a process we call **stochastic re-weighting**. This adaptation step supervises the surrogate model to maintain consistent predictions when processing graphs with varying edge weights, which is a critical requirement for generating reliable explanations.

As illustrated in Figure 2(b), given explanation subgraphs from the initial explainer, we construct weighted training graphs by stochastically assigning different weights to explanatory and non-explanatory edges indicated by the explanation subgraphs. Specifically, we use two well-separated truncated Gaussian distributions: explanatory edges receive weights sampled from a higher-mean distribution, while non-explanatory edges are sampled from a lower-mean distribution. This stochastic re-weighting strategy ensures that the surrogate model learns to respect the relative importance of explanatory edges, and the controlled randomness helps it generalize to the continuous weight variations encountered during soft mask optimization.

Gaussian Parameter Selection. Our stochastic re-weighting strategy samples edge weights to construct new weighted training graphs. Specifically, we use two truncated Gaussian distributions: explanatory edges receive weights sampled from $W_1 \sim \mathcal{TN}(\mu_1, \sigma^2; 0, 1)$, while non-explanatory edges receive weights from $W_2 \sim \mathcal{TN}(\mu_2, \sigma^2; 0, 1)$. We use the $[0, 1]$ interval because this range directly mimics the continuous, weighted graphs generated by the soft mask optimization in standard GNN explainers, which is essential for bridging the distributional shift.

To parameterize these distributions, we first fix two hyperparameters: the desired mean separation $\Delta\mu = \mu_1 - \mu_2 > 0$ and the overlap probability $\alpha = \mathbb{P}(\tilde{W}_1 < \tilde{W}_2)$, where $\tilde{W}_1 \sim \mathcal{N}(\mu_1, \sigma^2)$ and $\tilde{W}_2 \sim \mathcal{N}(\mu_2, \sigma^2)$ are the untruncated counterparts of W_1 and W_2. We set $\alpha = 0.001$ to ensure a clear separation between the two edge types. The overlap probability is given by: $\mathbb{P}(\tilde{W}_1 < \tilde{W}_2) = \mathbb{P}(\tilde{W}_1 - \tilde{W}_2 < 0) = \Phi\left(\frac{-\Delta\mu}{\sqrt{2\sigma^2}}\right)$, where $\Phi(\cdot)$ is the standard normal cumulative distribution function: $\Phi(w) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{w} e^{-t^2/2}\, dt$. By fixing $\Delta\mu$ and α, the variance σ^2 is computed as: $\sigma^2 = \frac{(\Delta\mu)^2}{2[\Phi^{-1}(\alpha)]^2}$. Once σ is determined, we select the means. We sample μ_2 from Uniform$(2\sigma, 1 - \Delta\mu - 2\sigma)$, which in turn determines $\mu_1 = \mu_2 + \Delta\mu$. This sampling strategy ensures that both μ_1 and μ_2 are positioned at least 2σ from the 0 and 1 boundaries. Consequently, the probability of the untruncated Gaussians $\tilde{W}_1$ and $\tilde{W}_2$ falling outside $[0, 1]$ is negligible. This alignment ensures that the truncated distributions $\mathcal{TN}$ used in practice retain means and variances almost identical to their untruncated $\mathcal{N}$ counterparts, preserving the desired separation properties.

Constructing Weighted Training Graphs. For each graph G in the training set, we use the current explainer Ψ to identify the set of explanatory edges $\mathcal{E}^{\text{exp}}$. We then construct a weighted graph G' by assigning edge weights sampled from $\mathcal{TN}(\mu_1, \sigma^2; 0, 1)$ to explanatory edges and from $\mathcal{TN}(\mu_2, \sigma^2; 0, 1)$ to non-explanatory edges. This stochastic re-weighting creates a training distribution that bridges the gap between unweighted graphs and the weighted graphs encountered during mask optimization. The full process is detailed in Appendix B Algorithm 2.

The surrogate GNN is then trained on both the original graphs $\mathcal{G}$ and the weighted graphs $\mathcal{G}'$:

$$\arg\min_{f^{(l)}} \left(\sum_{(G,y)\in\mathcal{G}\cup\mathcal{G}'} -y \log(f^{(l)}(G)) \right), \tag{3}$$

This joint training ensures that the surrogate model maintains accurate predictions on both unweighted graphs (preserving original performance) and weighted graphs (enabling reliable mask optimization).

3.2 Iterative Refinement Process

Our framework alternates between training the surrogate GNN on weighted graphs and updating the explainer, progressively refining explanations over L iterations. After updating the surrogate at each iteration, we update the explainer using the corresponding surrogate model $f^{(l)}$. This iterative process, summarized in Appendix B Algorithm 1, progressively refines from larger to smaller subgraphs, which alleviates distributional shifts that occur when searching for minimal explanations.

Remark. Despite involving iterative surrogate refinement, the computational overhead remains modest. In practice, we find that the framework requires only a small number of iterations (≤ 3) , and significant improvements are often observable after just a single adaptation step. As empirically demonstrated in Section 6.3 Figure 4, the cost per iteration is low, confirming that the iterative refinement process is computationally efficient.

4 Theoretical Analysis

Our framework is designed to address the critical issue of distributional shift, where a GNN trained on unweighted graphs yields unreliable explanations when evaluated on the weighted graphs inherent to the explanation process. The iterative refinement process, which produces a final robust model, $f^{(L)}$, is empirically effective. Here, we provide the theoretical foundation that justifies when explanations from our final robust model can be considered faithful to the original, non-robust model, $f^{(0)}$.

Following Zheng et al. [38], a task is (s, κ)-explainable if there exists an explanation function Ψ such that the label is nearly conditionally independent of the full input given $\Psi(G)$, with expected explanation size at most s. A classifier is (s, ζ)-explainable under the same condition with predictions replacing labels. We also adopt the stability assumption from Zheng et al. [38], which requires that if $\Psi(g) \subseteq g'$, then $\Psi(g') = \Psi(g)$. This assumption ensures that explanations remain consistent when the input graph is extended to a supergraph. Under this uniqueness condition, we can formally connect the explanations of the original model $f^{(0)}$ and the robust model $f^{(L)}$. The following proposition establishes that, when both models achieve errors close to the Bayes optimal, they share the same explanation function almost surely.

Proposition 4.1 (Shared explanation for $f^{(0)}$ and $f^{(L)}$)**.** *Consider a binary task with Bayes error $\epsilon^\star$. If the task is uniquely (s, κ)-explainable by $\Psi^\star$ under the stability assumption, and if $f^{(0)}$ and $f^{(L)}$ achieve errors at most $\epsilon^\star + \delta_0$ and $\epsilon^\star + \delta_L$, then both classifiers are (s, η_i)-explainable with respect to $\Psi^\star$ (with $\eta_i \to 0$ as $\delta_i, \epsilon^\star, \kappa \to 0$) and share the same explanation almost surely.*

This result shows that, under the stability and uniqueness assumptions, the robust model $f^{(L)}$ and the original model $f^{(0)}$ share the same underlying explanation almost surely. In other words, explanations from the robust model can be viewed as faithful surrogates for those of the original model, with the approximation error shrinking as both models approach Bayes optimality. The assumptions themselves are natural in many settings: stability reflects the idea that once a subgraph is sufficient to explain a prediction, adding irrelevant structure should not change the explanation; uniqueness means that for each instance, there is a well-defined explanatory subgraph. These principles echo the intuition behind widely used explanation methods such as LIME [22], which rely on the existence of a consistent local explanation. Together, the assumptions highlight why our theoretical result is relevant in practice. This is further supported by the empirical analysis in Appendix D.2, where we show that the robust surrogate preserves both the predictive behavior and explanatory structures of the original GNN.

5 Related Work

GNN Explanation Methods. Existing research on GNN explainability primarily focuses on instance-level explanations, where the goal is to identify influential subgraphs for specific predictions. The dominant paradigm relies on perturbation, where methods optimize soft or binary masks to isolate subgraphs that preserve the original prediction. This category includes approaches that learn soft edge masks via optimization [31, 18, 19, 26], employ search-based policies [33, 15], or leverage game-theoretic values [37]. Recent studies have further advanced GNN explainability through factorized explanation models [12], graph output attribution [17], efficient edge-induced subgraph search [16], and structure-aware interaction indices [6]. Other work has extended graph interpretability to specialized settings, including 3D molecular graphs [20] and multi-granular graph representations [30]. Despite improving fidelity, these methods inherently force the GNN to process modified graph structures that deviate from the discrete graphs seen during training, introducing artificial evidence.

Distribution Shift in GNN Explanations. To mitigate distribution shifts in GNN explanations, existing studies have explored various strategies for generating in-distribution explanations. MixupExplainer [35] uses graph augmentation to reduce distribution shifts, while ProxyExplainer [8] constructs in-distribution proxy graphs to facilitate explanation generation. RegExplainer [36] further adopts a mixup-based strategy to address distribution shifts in graph regression explanations. Other approaches, such as D4Explainer [7] and Gen-GraphEx [23], leverage generative models to produce in-distribution graph explanations, primarily for counterfactual or model-level interpretability. However, these approaches do not address the specific mismatch identified in our work, where the target GNN is trained on unweighted graphs but explained using continuous weighted masks. Unlike these

Table 1: Performance comparison of different explanation methods using GCN as the backbone

Explainer Method	BA-2motifs	MUTAG	Benzene	Alkane-Car.	Fluoride-Car.
GNNExplainer	$0.515_{\pm 0.107}$	$0.707_{\pm 0.086}$	$\mathbf{0.604}_{\pm 0.029}$	$0.650_{\pm 0.042}$	$0.640_{\pm 0.018}$
+ STORE	$\mathbf{0.528}_{\pm 0.108}$	$\mathbf{0.720}_{\pm 0.057}$	$0.595_{\pm 0.026}$	$\mathbf{0.686}_{\pm 0.026}$	$\mathbf{0.647}_{\pm 0.019}$
PGExplainer	$0.672_{\pm 0.326}$	$0.533_{\pm 0.272}$	$0.541_{\pm 0.204}$	$0.778_{\pm 0.237}$	$0.784_{\pm 0.031}$
+ STORE	$\mathbf{0.758}_{\pm 0.232}$	$\mathbf{0.774}_{\pm 0.189}$	$\mathbf{0.662}_{\pm 0.152}$	$\mathbf{0.801}_{\pm 0.227}$	$\mathbf{0.802}_{\pm 0.020}$
Refine	$0.522_{\pm 0.078}$	$0.502_{\pm 0.193}$	$0.723_{\pm 0.166}$	$0.623_{\pm 0.268}$	$0.633_{\pm 0.051}$
+ STORE	$\mathbf{0.546}_{\pm 0.095}$	$\mathbf{0.529}_{\pm 0.197}$	$\mathbf{0.747}_{\pm 0.178}$	$\mathbf{0.752}_{\pm 0.153}$	$\mathbf{0.648}_{\pm 0.043}$
MixupExplainer	$0.847_{\pm 0.121}$	$\mathbf{0.799}_{\pm 0.154}$	$0.555_{\pm 0.270}$	$0.444_{\pm 0.312}$	$0.664_{\pm 0.090}$
+ STORE	$\mathbf{0.871}_{\pm 0.110}$	$0.783_{\pm 0.160}$	$\mathbf{0.570}_{\pm 0.252}$	$\mathbf{0.493}_{\pm 0.299}$	$\mathbf{0.667}_{\pm 0.130}$
ProxyExplainer	$0.729_{\pm 0.226}$	$0.614_{\pm 0.232}$	$0.584_{\pm 0.214}$	$0.678_{\pm 0.198}$	$0.718_{\pm 0.036}$
+ STORE	$\mathbf{0.812}_{\pm 0.137}$	$\mathbf{0.615}_{\pm 0.233}$	$\mathbf{0.615}_{\pm 0.190}$	$\mathbf{0.766}_{\pm 0.128}$	$\mathbf{0.742}_{\pm 0.029}$

Table 2: Fidelity evaluation

		BA-2motifs	MUTAG	Fluoride-Car.	D&D	Graph-SST2
$Fid_{\alpha_1,+} \uparrow$	PGExplainer	$0.038_{\pm 0.029}$	$0.042_{\pm 0.008}$	$0.055_{\pm 0.012}$	$0.010_{\pm 0.011}$	$0.008_{\pm 0.002}$
	+ STORE	$\mathbf{0.074}_{\pm 0.023}$	$\mathbf{0.060}_{\pm 0.002}$	$\mathbf{0.064}_{\pm 0.010}$	$\mathbf{0.022}_{\pm 0.008}$	$\mathbf{0.011}_{\pm 0.001}$
$Fid_{\alpha_2,-} \downarrow$	PGExplainer	$0.037_{\pm 0.023}$	$0.063_{\pm 0.044}$	$0.018_{\pm 0.007}$	$0.003_{\pm 0.005}$	$0.005_{\pm 0.002}$
	+ STORE	$\mathbf{0.035}_{\pm 0.022}$	$\mathbf{0.041}_{\pm 0.022}$	$\mathbf{0.016}_{\pm 0.008}$	$\mathbf{0.001}_{\pm 0.004}$	$\mathbf{0.003}_{\pm 0.002}$

methods, our STORE framework explicitly bridges this gap by training a robust surrogate model capable of handling the continuous edge weights inherent to the explanation process.

6 Experiments

6.1 Experimental Setup

Datasets. We evaluate our method on five benchmark datasets with ground-truth explanations. These include four real-world datasets: MUTAG [18], Benzene, Alkane-Carbonyl, and Fluoride-Carbonyl [1], as well as one synthetic dataset, BA-2motifs [18]. Detailed dataset statistics and descriptions are provided in Appendix C.1.

Baselines. We compare our method with five representative explainers: GNNExplainer [31], PG-Explainer [18], ReFine [26], MixupExplainer [35], and ProxyExplainer [8]. We adopt a three-layer Graph Convolutional Network (GCN) [14] as the backbone, using the experimental settings in [31, 18]. Experiments on another representative GNN, Graph Isomorphism Network (GIN) [29], are provided in Appendix C.2.

Settings. Unlike previous works [31, 18], which train a single GNN with a fixed seed and evaluate the explainer using 10 random seeds, we observe that the robustness of the GNN itself can vary substantially across different initializations and may exhibit different degrees of distributional shift. To obtain a more reliable and unbiased evaluation, we train both the GNN and the explainer under 10 random seeds. For each seed, a GNN is trained from scratch and an explainer is applied to that model, and we report the final results as the average across the 10 runs[2]. Additional implementation details are provided in Appendix C.3.

6.2 Quantitative Evaluation

We report the mean AUC-ROC over 10 random runs to quantitatively evaluate the quality of explanations. Table 1 shows the results for five explanation methods, each evaluated in both their vanilla form and when integrated with our iterative framework, using GCN as the backbone. From the results, we have several key observations: First, although different explanation methods exhibit varying levels of performance, all of them consistently benefit from our iterative framework, STORE, which improves AUC-ROC scores over their vanilla counterparts across most datasets.

[2]Our code is available at `https://github.com/realMoana/STORE`

For example, on BA-2motifs, we observe a 12.8% relative gain for PGExplainer and a 11.4% improvement for ProxyExplainer, confirming the generalizability of our method across both synthetic and real-world datasets. Finally, STORE enhances stability, particularly on OOD-prone datasets such as BA-2motifs. While vanilla explainers (e.g., MixupExplainer) experience instability in some datasets, our method mitigates this by preserving semantically important substructures during iterative training. This targeted perturbation reduces the risk of degrading important graph structures, resulting in more reliable explanations. Moreover, Appendix D.1 reports results under the GIN architecture, showing that STORE is compatible with different GNN architectures and explainers.

To further evaluate the faithfulness of explanations, we adopt fidelity-based metrics. While standard fidelity metrics [33] are widely used, they suffer from OOD problem [3], which compromise their reliability. To address this, we use the robust fidelity scores ($Fid_{\alpha_1,+}$, $Fid_{\alpha_2,-}$), following the formulation in [38], with the default parameters $\alpha_1 = 0.1, \alpha_2 = 0.9$. As shown in Table 2, STORE outperforms all baselines under both metrics, including the large-scale D&D [10] and Graph-SST2 datasets [34]. We also provide a visual comparison in Appendix D.4, including the ground-truth explanations, those produced by the vanilla PGExplainer, and those generated by our method.

6.3 Efficiency Analysis

Analysis of Iterations. To determine whether only a small number of iterations is sufficient for stable performance gains, we vary the number of refinement iterations and evaluate the explanation results. We report AUC-ROC results for iteration counts ranging from 0 to 5. Additionally, we implement a progressively shrinking top-k strategy during iterative surrogate model training: at each iteration, only the top-k most important edges are retained in the explanation graph, with k decreasing from 0.9 to 0.1 across iterations 1 through 5. The progressively shrinking strategy begins with broad subgraph exploration and incrementally focuses on discriminative structures. This process enforces finer attention to class-relevant structures while mitigating distractions from less informative edges.

As shown in Figure 3, on BA-2motifs, the AUC-ROC steadily increases during the first few iterations and plateaus after iteration 3, suggesting that iterative refinement and stronger substructure contrast effectively enhance explanation fidelity and model robustness. On MUTAG and Fluoride-Carbonyl, performance peaks at iteration 3 and slightly declines thereafter. This suggests that overly aggressive pruning of edges, resulting from smaller k values in later iterations, may remove informative substructures, thus impairing both explanation fidelity and model generalization. Notably, a substantial improvement already appears at iteration 1 across all datasets compared with the original model at iteration 0, demonstrating that STORE does not rely on many refinement steps and showing that even one or two iterations are sufficient to achieve stable performance gains.

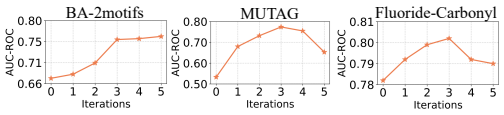


Figure 3: AUC-ROC performance with different numbers of iterations.

Moreover, the shrinking top-k strategy encourages the explainer to focus on the most important edges over time. The observed improvements suggest that this gradual refinement enhances explanation accuracy and improves the robustness of the GNN to noise from less important structures. We also find that the performance after iterative training consistently surpasses the original baseline across all iterations, which suggests the robustness of our method.

Running Time. Figure 4 illustrates the computational efficiency of our framework by reporting the average runtime calculated over 10 independent runs. We compare the standard approach, which includes GCN training and PGExplainer optimization, against one complete iteration of STORE that involves stochastic graph construction and surrogate model training. The results indicate that the additional overhead is modest across all datasets. Notably, for BA-2motifs and Fluoride-Carbonyl, the runtime for one iteration is nearly equivalent to the standard training baseline, while the increase on MUTAG remains within a highly practical range. Given the significant

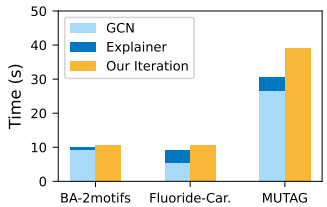


Figure 4: Average training time. GCN and Explainer jointly represent the standard pipeline.

improvements in explanation accuracy shown in Figure 3, these results confirm that our iterative framework achieves substantial performance gains with minimal computational cost.

6.4 Ablation Studies

Sampling Methods. To validate our Gaussian-based re-weighting, we compare it against two baselines: i) Random Perturbation: Edge weights are sampled uniformly from $(0, 1)$, ignoring explainer importance. ii) Uniform-Based Sampling: A structured uniform approach where explanatory edges are sampled from $W_1 \sim \text{Uniform}(u, 1)$, and non-explanatory edges are scaled by noise $Z \sim \text{Uniform}(0, 1)$ such that $W_2 = W_1 \cdot Z$. This concentrates non-explanatory weights in $(0, u)$, providing separation but lacking the smooth distributional properties of our Gaussian approach.

Table 3: Performance comparison with different sampling methods

	BA-2motifs	MUTAG	Fluoride-Car.
Random	$0.631_{\pm 0.282}$	$0.513_{\pm 0.264}$	$0.763_{\pm 0.035}$
Uniform	$0.678_{\pm 0.241}$	$0.757_{\pm 0.154}$	$0.795_{\pm 0.022}$
Gaussian	$0.758_{\pm 0.232}$	$0.774_{\pm 0.189}$	$0.802_{\pm 0.020}$

Table 3 reports the performance on BA-2motifs, MUTAG, and Fluoride-Carbonyl under different sampling strategies. We observe that the random perturbation yields the lowest AUC-ROC scores and the largest variances, indicating that random noise disrupts the learning process, leading to reduced stability. The Uniform-based strategy leads to notable performance gains compared to random perturbation, especially on MUTAG and Fluoride-Carbonyl, demonstrating that incorporating edge importance helps preserve informative structures. The Gaussian-based strategy used in STORE achieves the best performance on all three datasets. It outperforms the other two variants in average performance, suggesting that it provides a better balance between edge importance preservation and controlled perturbation.

Effects of $\Delta\mu$. We conduct a hyperparameter analysis to investigate how the difference $\Delta\mu$ between the truncated Gaussian distributions for explanatory and non-explanatory edges affects the explainer's performance. As shown in Figure 5, we vary $\Delta\mu$ from 0.1 to 0.7 and report the AUC-ROC scores on BA-2motifs, MUTAG, and Fluoride-Carbonyl. For the larger values $\Delta\mu = 0.6$ and $\Delta\mu = 0.7$, we reduce the boundary margin in the mean-sampling interval to keep it non-empty, while keeping the truncated Gaussian sampling on $[0, 1]$. We observe that the performance remains relatively stable across a wide range of $\Delta\mu$ values. On BA-2motifs and Fluoride-Carbonyl, the AUC-ROC improves steadily with increasing $\Delta\mu$ and reaches its peak around 0.5. On MUTAG, the performance improves initially and peaks at $\Delta\mu = 0.4$, then drops slightly beyond that. These findings indicate that a moderate separation between important and unimportant edges helps the model better distinguish informative substructures.

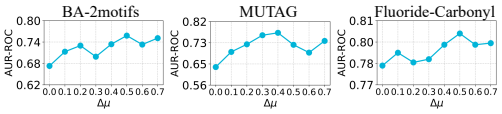


Figure 5: AUC-ROC performance with different $\Delta\mu$ for PGExplainer.

6.5 Alleviating Distribution Shifts

In this section, we visualize the classification accuracy under different edge weight configurations, where the x-axis denotes explanation edge weights and the y-axis denotes non-explanation edge weights. Figure 6 presents heat maps on BA-2motifs, MUTAG, and Fluoride-Carbonyl. For each dataset, we compare the original GCN model with our explanation-aware surrogate model.

In the original model, we observe that only the upper-right region achieves high accuracy, where both explanation and non-explanation edge weights are large. In contrast, accuracy drops significantly in the lower-right region, where the graph contains only explanation edges. These results indicate that the model heavily relies on the global graph structure and fails to generalize when presented with explanation subgraphs, revealing the OOD problem. On the MUTAG dataset, we further find that the model can still achieve high accuracy even when explanation edge weights are small and non-explanation edge weights are large. This suggests that the model may exploit spurious background structures to make predictions, thereby undermining the faithfulness of explanation subgraphs. After

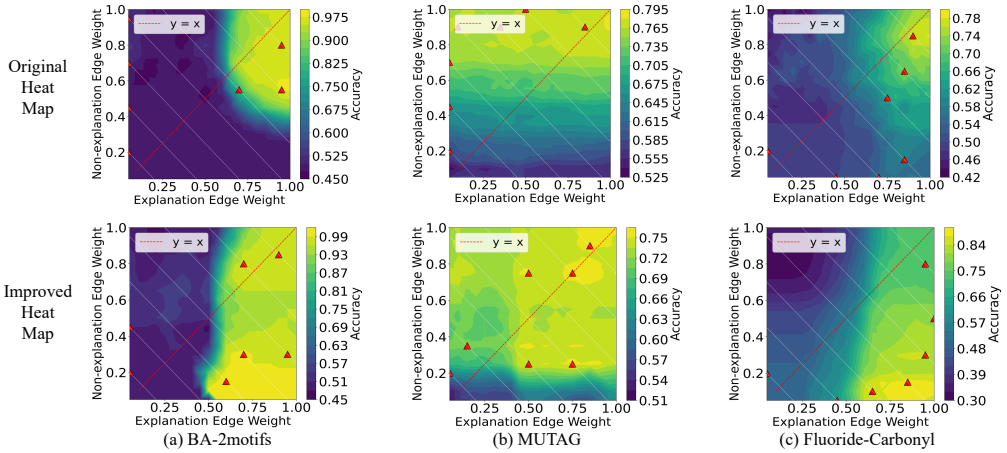


Figure 6: Heat maps of classification accuracy under different edge weights.

applying explanation-aware iteration training, the model achieves consistently higher accuracy in a broader range of configurations, especially in the lower-right region. This indicates that the model has learned to better focus on explanation structures and that our method mitigates the OOD problem. In Appendix D.2, we further report classification results and qualitative visualizations, confirming that the surrogate GCN preserves both performance and explanatory consistency.

7 Conclusion

In this paper, we identify a critical flaw in post-hoc GNN explainability: the distributional shift caused by applying continuous explanation masks to models trained on unweighted graphs. To overcome this, we propose STORE, an iterative framework that constructs a robust surrogate model tailored to handle the weighted graphs inherent to the explanation process. By alternating between explanation identification and stochastic re-weighting, we ensure this surrogate serves as a faithful proxy for the original model while maintaining gradient reliability during mask optimization. Extensive experiments demonstrate that our method effectively mitigates the distribution mismatch, providing a general and model-agnostic solution for enhancing GNN interpretability.

Limitations. This work focuses on static graph classification. Extending the framework to node-level prediction and larger-scale graph learning, as well as developing adaptive stopping criteria for surrogate refinement, are promising directions for future work.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Science Foundation under grants ECCS-2242700 and IIS-2529283, the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 Grant (26-SCIS-SMU-008), and the SMU Presidential Early Career Professorship (SMU-PEP) Start-Up Grant.

References

[1] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 10(1):144, 2023.

[2] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[3] Kenza Amara, Mennatallah El-Assady, and Rex Ying. Ginx-eval: Towards in-distribution evaluation of graph neural network explanations. *arXiv preprint arXiv:2309.16223*, 2023.

[4] Valentin Anklin, Pushpak Pati, Guillaume Jaume, Behzad Bozorgtabar, Antonio Foncubierta-Rodriguez, Jean-Philippe Thiran, Mathilde Sibony, Maria Gabrani, and Orcun Goksel. Learning

whole-slide segmentation from inexact and incomplete labels using tissue graphs. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part II 24*, pages 636–646. Springer, 2021.

[5] Fanchen Bu and Kijung Shin. On improving the cohesiveness of graphs by merging nodes: Formulation, analysis, and algorithms. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 117–129, 2023.

[6] Ngoc Bui, Hieu Trung Nguyen, Viet Anh Nguyen, and Rex Ying. Explaining graph neural networks via structure-aware interaction index. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4854–4883. PMLR, 2024.

[7] Jialin Chen, Shirley Wu, Abhijit Gupta, and Rex Ying. D4explainer: In-distribution explanations of graph neural network via discrete denoising diffusion. *Advances in Neural Information Processing Systems*, 36:78964–78986, 2023.

[8] Zhuomin Chen, Jiaxing Zhang, Jingchao Ni, Xiaoting Li, Yuchen Bian, Md Mezbahul Islam, Ananda Mondal, Hua Wei, and Dongsheng Luo. Generating in-distribution proxy graphs for explaining graph neural networks. In *Forty-first International Conference on Machine Learning*, 2024.

[9] Zhuomin Chen, Hojat Allah Salehi, Esteban Schafir, Xu Zheng, Jiaxing Zhang, Hua Wei, Jingchao Ni, Farhad Shirani, and Dongsheng Luo. Addressing structural distribution shift in explanations for graph neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 2026.

[10] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

[11] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[12] Rundong Huang, Farhad Shirani, and Dongsheng Luo. Factorized explainer for graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12626–12634, 2024.

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

[14] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[15] Wenqian Li, Yinchuan Li, Zhigang Li, Jianye HAO, and Yan Pang. DAG matters! GFlownets enhanced explainer for graph neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[16] Shengyao Lu, Bang Liu, Keith G. Mills, Jiao He, and Di Niu. Eig-search: Generating edge-induced subgraphs for GNN explanation in linear time. In *Forty-first International Conference on Machine Learning*, 2024.

[17] Shengyao Lu, Keith G. Mills, Jiao He, Bang Liu, and Di Niu. Goat: Explaining graph neural networks via graph output attribution. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

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

[19] Dongsheng Luo, Tianxiang Zhao, Wei Cheng, Dongkuan Xu, Feng Han, Wenchao Yu, Xiao Liu, Haifeng Chen, and Xiang Zhang. Towards inductive and efficient explanations for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5245–5259, 2024.

[20] Jingxiang Qu, Wenhan Gao, Jiaxing Zhang, Xufeng Liu, Hua Wei, Haibin Ling, and Yi Liu. RISE: Radius of influence based subgraph extraction for 3D molecular graph explanation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 50744–50761. PMLR, 2025.

[21] Arun Rai. Explainable ai: From black box to glass box. *Journal of the academy of marketing science*, 48:137–141, 2020.

[22] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[23] Sayan Saha, Monidipa Das, and Sanghamitra Bandyopadhyay. Gen-graphex: Generative in-distribution graph explanations for time-efficient model-level interpretability of gnns. *IEEE Transactions on Neural Networks and Learning Systems*, 36(11):19775–19789, 2025.

[24] Minh Vu and My T Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33:12225–12235, 2020.

[25] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 968–977, 2019.

[26] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, 34:18446–18458, 2021.

[27] Xiaoqi Wang and Han-Wei Shen. Gnninterpreter: A probabilistic generative model-level explanation for graph neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[28] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[29] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *ICLR*, 2019.

[30] Jie Yang, Yuwen Wang, Kaixuan Chen, Tongya Zheng, Yihe Zhou, Zhenbang Xiao, Ji Cao, Mingli Song, and Shunyu Liu. From GNNs to trees: Multi-granular interpretability for graph neural networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

[32] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 430–438, 2020.

[33] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*, pages 12241–12252. PMLR, 2021.

[34] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5782–5799, 2022.

[35] Jiaxing Zhang, Dongsheng Luo, and Hua Wei. Mixupexplainer: Generalizing explanations for graph neural networks with data augmentation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3286–3296, 2023.

[36] Jiaxing Zhang, Zhuomin Chen, Hao Mei, Longchao Da, Dongsheng Luo, and Hua Wei. Regexplainer: Generating explanations for graph neural networks in regression tasks. *Advances in Neural Information Processing Systems*, 37:79282–79306, 2024.

[37] Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. Gstarx: Explaining graph neural networks with structure-aware cooperative games. In *Advances in Neural Information Processing Systems*, volume 35, pages 19810–19823, 2022.

[38] Xu Zheng, Farhad Shirani, Tianchun Wang, Wei Cheng, Zhuomin Chen, Haifeng Chen, Hua Wei, and Dongsheng Luo. Towards robust fidelity for evaluating explainability of graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

A Theoretical Analysis: Full Proofs

This appendix provides the detailed definitions and proofs omitted from the main text. We follow the formalism of Zheng et al. [38].

A.1 Preliminaries

Definition A.1 (Explainable Classification Task [38])**.** A task with distribution $P_{Y,\boldsymbol{X},\mathbf{A}}$ is (s,κ)-explainable if there exists an explanation function $\Psi : \mathcal{G} \to 2^{\mathcal{V}} \times 2^{\mathcal{E}}$ such that

$$I\big(Y; G \,\big|\, \mathbb{1}_{\Psi(G)}\big) \le \kappa \quad \text{and} \quad \mathbb{E}_G\big[|\mathcal{E}_{\text{exp}}|\big] \le s,$$

where $\Psi(G) = (\mathcal{V}_{\text{exp}}, \mathcal{E}_{\text{exp}})$.

Definition A.2 (Explainable Classifier [38])**.** For a classifier $f : \mathcal{G} \to \Delta_{\mathcal{Y}}$, let $\widehat{Y} \sim f(G)$. Then f is (s,ζ)-explainable if there exists Ψ with

$$I\Big(\widehat{Y}; G \,\Big|\, \mathbb{1}_{\Psi(G)}\Big) \le \zeta \quad \text{and} \quad \mathbb{E}_G\big[|\mathcal{E}_{\text{exp}}|\big] \le s.$$

Condition 1 (Stability under supergraphs). For all g, g', if $\Psi(g) \subseteq g'$, then $\Psi(g') = \Psi(g)$. This condition implies

$$I(\widehat{Y}; G \,|\, \mathbb{1}_{\Psi(G)}) = I(\widehat{Y}; G \,|\, \Psi(G)).$$

Definition A.3 (Uniqueness of Task Explanation)**.** A binary task is *uniquely* (s,κ)*-explainable* if there exists an explanation $\Psi^\star$ satisfying Definition A.1 and Condition 1, and for any other valid Ψ we have

$$\mathbb{P}\big(\Psi(G) = \Psi^\star(G)\big) = 1.$$

A.2 Proof of Proposition 4.1

Proof. The Theorem 1 in Zheng et al. [38] states that, if a task is (s,κ)-explainable with an explanation function satisfying Condition 1, then any classifier f with error at most $\epsilon^\star + \delta$ is (s,η)-explainable for some $\eta \triangleq h_b(\tau) + \tau$, where $h_b(\tau) \triangleq -\tau \log \tau - (1-\tau)\log 1 - \tau$, $\tau \in [0,1]$ is the binary entropy function,

$$\tau \triangleq \frac{\big(2\sqrt{2}\epsilon^* + \sqrt{\xi}\big)^2}{2}, \quad \xi \triangleq \delta + e_{max}(\epsilon^*, \kappa) - \epsilon^*, \quad \delta \in (0, 9\epsilon^* - e_{max}(\epsilon^*, \kappa)),$$

and $e_{\max}(\epsilon^\star, \kappa)$ is the error bound from the reverse-Fano inequality, defined in the Proposition 1 of Zheng et al. [38]. In particular, $\eta \to 0$ as $(\delta, \epsilon^\star, \kappa) \to 0$. The classifiers $f^{(0)}$ and $f^{(L)}$ defined in our paper satisfy the accuracy conditions with error at most $\epsilon^\star + \delta$. Thus, both $f^{(0)}$ and $f^{(L)}$ are (s,η_i)-explainable with respect to certain explanation functions.

Furthermore, the task admits a unique explanation $\Psi^\star$ by using the fact that the explanation $\Psi^\star$ satisfies Definition A.1 and Condition 1 in Definition A.3. Thus, any valid explanation Ψ that makes a classifier (s,η)-explainable must coincide with $\Psi^\star$ almost surely. Hence, both $f^{(0)}$ and $f^{(L)}$ are explained by the same $\Psi^\star$. This completes the proof. □

A.3 Discussion

The proof highlights two essential assumptions:

1. Condition 1, which allows conditioning on $\Psi(G)$ directly.
2. Uniqueness of the task explanation, which ensures both classifiers align with the same $\Psi^\star$.

These strengthen the original statement and remove the gap caused by the possibility of multiple valid explanations.

B Algorithm

Algorithm 1 Iterative Explanation Framework

Input: A set of original graphs $\mathcal{G}$, GNN model $f(\cdot)$, explainer $\Psi(\cdot)$, number of iterations L
Output: explainer $\Psi^{(L)}(\cdot)$

$f^{(0)} \leftarrow f$
$\Psi^{(0)} \leftarrow \text{TrainExplainer}(f^{(0)}, \mathcal{G})$
for $l = 1$ to L **do**
  $\mathcal{G}^{\text{exp}} \leftarrow \Psi^{(l-1)}(\mathcal{G})$ ▷ Generate explanation graphs
  $\mathcal{G}' \leftarrow \text{AugmentGraph}(\mathcal{G}, \mathcal{G}^{\text{exp}})$ ▷ Construct augmentation
  $f^{(l)} \leftarrow \text{TrainGNN}(\mathcal{G}')$ ▷ Train surrogate GNN on augmented data
  $\Psi^{(l)} \leftarrow \text{TrainExplainer}(f^{(l)}, \mathcal{G})$ ▷ Retrain explainer on original graphs
end for
return $\Psi^{(L)}(\cdot)$

Algorithm 2 Construction of Augmented Graphs

Input: A set of original graphs $\mathcal{G}$, corresponding explanation graphs $\mathcal{G}^{\text{exp}}$, mean shift $\Delta\mu$, top-k edges to select, separation probability α
Output: A set of augmented graphs $\mathcal{G}'$

Compute variance: $\sigma^2 = \frac{(\Delta\mu)^2}{2(\Phi^{-1}(1-\alpha))^2}$
Sample $\mu_2 \sim \mathcal{U}(2\sigma, 1 - \Delta\mu - 2\sigma)$, set $\mu_1 = \mu_2 + \Delta\mu$
Initialize augmented graph set: $\mathcal{G}' \leftarrow \emptyset$
for $i = 1$ to $|\mathcal{G}|$ **do**
  Copy original graph: $\mathcal{G}^{\text{aug},i} \leftarrow \mathcal{G}^i$
  Select top-k explanation edges: $\mathcal{E}^{\text{exp}} \leftarrow \text{top-}k(\mathcal{G}^{\text{exp},i})$
  for each edge $e \in \mathcal{E}^{\text{exp}}$ **do**
    Sample $W_{1,e} \sim \mathcal{TN}(\mu_1, \sigma^2; 0, 1)$
    Assign $w(e) \leftarrow W_{1,e}$ in $\mathcal{G}^{\text{aug},i}$
  end for
  for each edge $e \in \mathcal{G}^i \setminus \mathcal{E}^{\text{exp}}$ **do**
    Sample $W_{2,e} \sim \mathcal{TN}(\mu_2, \sigma^2; 0, 1)$
    Assign $w(e) \leftarrow W_{2,e}$ in $\mathcal{G}^{\text{aug},i}$
  end for
  Add to augmented set: $\mathcal{G}' \leftarrow \mathcal{G}' \cup \{\mathcal{G}^{\text{aug},i}\}$
end for
return $\mathcal{G}'$

C Detailed Experimental Setup

C.1 Datasets

- **BA-2motifs** [18]. This synthetic dataset contains 1,000 graphs generated from the Barabási–Albert (BA) model. Each graph belongs to one of two classes: one with house-shaped motifs and the other with five-node cycle structures.

- **MUTAG** [18]. MUTAG consists of 4,337 molecular graphs, each labeled by its mutagenic effect on the Gram-negative bacterium Salmonella typhimurium.

- **Benzene** [1]. Benzene consists of 12,000 molecular graphs, categorized into two groups—those containing benzene rings and those without.

- **Alkane-Carbonyl** [1]. This dataset contains 4,326 molecules. For positive labels, we consider molecules that have both alkane and carbonyl functional groups in one molecule.

- **Fluoride-Carbonyl** [1]. Fluoride-Carbonyl consists of 8,671 molecular graphs, whose ground-truth explanations are based on both the presence of fluoride atoms and carbonyl functional groups in the molecule.

- **D&D** [10]. D&D consists of protein graphs labeled as either enzymes or non-enzymes. Nodes correspond to amino acids, and an edge is added between two nodes whenever their spatial distance is no greater than 6 Angstroms.

- **Graph-SST2** [34]. This dataset is a sentiment-analysis benchmark in which each sentence is converted into a graph representation, and the resulting graphs are annotated with their corresponding sentiment labels.

A summary of dataset statistics is provided in Table 4.

Table 4: Statistics of datasets used for graph classification task

Dataset	Domain	#Graphs	Avg.#nodes	Avg.#edges	#Feature
BA-2motifs	Synthetic	1,000	25	51	10
MUTAG	Biochemical molecules	4,337	30.32	30.77	14
Benzene	Biochemical molecules	12,000	20.58	43.65	14
Alkane-Car.	Biochemical molecules	4,326	21.13	44.95	14
Fluoride-Car.	Biochemical molecules	8,671	21.36	45.37	14
D&D	Bioinformatics	1,178	284.32	715.66	89
Graph-SST2	Natural language	70,042	10.20	18.40	768

C.2 Baselines

- **GNNExplainer** [31]. GNNExplainer learns edge and node feature masks to identify graph components relevant to a prediction. It applies these masks element-wise to the input graph and optimizes them through a mutual-information objective.

- **PGExplainer** [18]. Building on GNNExplainer, PGExplainer models edge selection using Bernoulli distributions under a Gilbert-graph assumption. The inclusion probability of each edge is predicted from the concatenated embeddings of its endpoint nodes, and Gumbel-Softmax enables differentiable sampling during training.

- **ReFine** [26]. ReFine adopts a two-stage approach. It first estimates edge selection probabilities at the class level using mutual-information and contrastive objectives. These probabilities then guide

edge sampling during instance-level fine-tuning to obtain explanations that retain information relevant to the model's prediction.

- **MixupExplainer** [35]. MixupExplainer uses a non-parametric augmentation strategy to reduce distribution shifts. It combines the selected explanation with a background subgraph sampled from another graph without conditioning on its class label.

- **ProxyExplainer** [8]. ProxyExplainer mitigates the distributional shift in explanation by generating in-distribution proxy graphs via a parametric graph generator, ensuring both fidelity and alignment with the original graphs.

C.3 Implementation

We use the Adam optimizer [13] with a learning rate of 1×10^{-3} for training the GNN, and use the same optimizer with a weight decay of 5×10^{-4} for training the explainer. We use a linearly decreasing top-schedule, with k = 0.9,0.7,0.5 for iterations 1–3, respectively. At each iteration, we retain the top-k fraction of edges. The AUC-ROC metric is adopted for quantitative evaluation. All experiments are conducted on a Linux machine equipped with four NVIDIA A100-PCIE GPUs, each with 40 GB of memory. The CUDA version is 12.3, and the driver version is 545.23.08.

D Extra Experimental Study

D.1 Experiments with GIN Model

In Table 5, we report results using GIN as the backbone architecture. The results show that STORE consistently improves the performance of various explanation methods across most datasets. For example, PGExplainer achieves a 29.2% improvement on Benzene and a 7.7% gain on Fluoride-Carbonyl, while ReFine and ProxyExplainer also show improvements on BA-2motifs and Fluoride-Carbonyl, respectively. These results further demonstrate the model-agnostic nature of our framework: even when switching to a different GNN architecture, the iterative refinement process still improves explanation quality.

Table 5: Performance comparison of different explanation methods using GIN as the backbone.

Explainer Method	BA-2motifs	MUTAG	Benzene	Alkane-Car.	Fluoride-Car.
GNNExplainer	$0.511_{\pm 0.005}$	$0.553_{\pm 0.127}$	$0.469_{\pm 0.079}$	$0.502_{\pm 0.126}$	$0.501_{\pm 0.029}$
+ STORE	$\mathbf{0.544}_{\pm 0.053}$	$\mathbf{0.589}_{\pm 0.114}$	$\mathbf{0.514}_{\pm 0.018}$	$\mathbf{0.552}_{\pm 0.071}$	$\mathbf{0.513}_{\pm 0.005}$
PGExplainer	$0.756_{\pm 0.315}$	$0.489_{\pm 0.246}$	$0.617_{\pm 0.260}$	$0.495_{\pm 0.259}$	$0.678_{\pm 0.088}$
+ STORE	$\mathbf{0.791}_{\pm 0.176}$	$\mathbf{0.657}_{\pm 0.209}$	$\mathbf{0.797}_{\pm 0.108}$	$\mathbf{0.631}_{\pm 0.258}$	$\mathbf{0.730}_{\pm 0.118}$
Refine	$0.518_{\pm 0.066}$	$0.535_{\pm 0.225}$	$0.499_{\pm 0.127}$	$0.603_{\pm 0.190}$	$0.551_{\pm 0.074}$
+ STORE	$\mathbf{0.584}_{\pm 0.066}$	$\mathbf{0.537}_{\pm 0.150}$	$\mathbf{0.638}_{\pm 0.158}$	$\mathbf{0.650}_{\pm 0.117}$	$\mathbf{0.597}_{\pm 0.054}$
MixupExplainer	$0.738_{\pm 0.281}$	$\mathbf{0.480}_{\pm 0.178}$	$0.593_{\pm 0.227}$	$0.420_{\pm 0.260}$	$0.767_{\pm 0.088}$
+ STORE	$\mathbf{0.759}_{\pm 0.161}$	$0.475_{\pm 0.192}$	$\mathbf{0.649}_{\pm 0.213}$	$\mathbf{0.464}_{\pm 0.236}$	$\mathbf{0.785}_{\pm 0.065}$
ProxyExplainer	$0.621_{\pm 0.205}$	$0.567_{\pm 0.110}$	$0.537_{\pm 0.139}$	$0.465_{\pm 0.156}$	$\mathbf{0.605}_{\pm 0.055}$
+ STORE	$\mathbf{0.659}_{\pm 0.173}$	$\mathbf{0.644}_{\pm 0.131}$	$\mathbf{0.582}_{\pm 0.136}$	$\mathbf{0.582}_{\pm 0.129}$	$0.600_{\pm 0.067}$

D.2 Validity of Robust Surrogate Model

In Section 6.2, we have demonstrated the effectiveness of STORE in generating high-quality explanations. These results confirm the practical utility of addressing the distributional shift. To further validate that the robust surrogate remains a faithful proxy for the original model, we directly compare their behaviors on the original unweighted graphs. Specifically, we evaluate whether the surrogate preserves the original model's predictive performance, sample-wise predictions, and explanatory structures.

Predictive Consistency. First, we examine whether the robust surrogate preserves the predictive capability of the original model. Table 6 reports the graph classification performance of the original

GCN versus the final robust surrogate on the original unweighted test set. For MUTAG, we observe slight improvements in all metrics, while for BA-2motifs and Fluoride-Carbonyl, the performance remains highly stable. These findings indicate that while the surrogate model extends its domain to handle weighted graphs, it retains the predictive performance of the original model.

Since similar accuracy alone does not guarantee sample-wise consistency, we further evaluate the direct agreement between the two models in Table 7. Let $\hat{y}_i^{(0)}$ and $\hat{y}_i^{(L)}$ denote the predictions of the original and surrogate GCNs, respectively. We define prediction agreement as the fraction of test samples satisfying $\hat{y}_i^{(0)} = \hat{y}_i^{(L)}$. We also report the IoU between the correctly predicted sample sets, where $C^{(0)} = \{i : \hat{y}_i^{(0)} = y_i\}$ and $C^{(L)} = \{i : \hat{y}_i^{(L)} = y_i\}$. For reference, we include a simple independence baseline computed from the two models' marginal test accuracies. The consistently high agreement suggests that the surrogate largely preserves the decision behavior of the original GNN rather than merely matching its overall accuracy.

Table 6: Graph classification performance of the original GCN model and the final robust surrogate GCN model.

		BA-2motifs	MUTAG	Fluoride-Carbonyl
Train Acc	Original	$0.923_{\pm 0.061}$	$0.787_{\pm 0.026}$	$0.764_{\pm 0.036}$
	Surrogate	$0.915_{\pm 0.057}$	$0.802_{\pm 0.010}$	$0.757_{\pm 0.038}$
Val Acc	Original	$0.942_{\pm 0.050}$	$0.785_{\pm 0.025}$	$0.773_{\pm 0.044}$
	Surrogate	$0.933_{\pm 0.062}$	$0.791_{\pm 0.011}$	$0.766_{\pm 0.033}$
Test Acc	Original	$0.934_{\pm 0.049}$	$0.771_{\pm 0.026}$	$0.750_{\pm 0.053}$
	Surrogate	$0.926_{\pm 0.062}$	$0.786_{\pm 0.014}$	$0.752_{\pm 0.040}$

Table 7: Prediction and explanation agreement between the original GCN and the final robust surrogate GCN.

Metric	BA-2motifs	MUTAG	Fluoride-Carbonyl
Indep. baseline	87.0%	65.5%	62.6%
Pred. agreement	93.5%	80.3%	82.8%
Correct-set IoU	92.6%	78.1%	79.5%
Exp. overlap	89.7%	76.8%	84.4%
Exp. IoU	81.9%	72.6%	76.3%

Consistency of Explanation Structures. We further evaluate whether the surrogate model preserves the explanatory structures of the original model. For each test graph, we extract the top-k edges according to the explanation scores from the original and surrogate explainers, denoted by $E_i^{(0)}$ and $E_i^{(L)}$, respectively. Table 7 reports the average explanation overlap, $|E_i^{(0)} \cap E_i^{(L)}| / \min\{|E_i^{(0)}|, |E_i^{(L)}|\}$, and the average explanation IoU, $|E_i^{(0)} \cap E_i^{(L)}| / |E_i^{(0)} \cup E_i^{(L)}|$. The results show substantial explanation-level agreement across datasets, suggesting that the surrogate relies on graph substructures similar to those used by the original GNN.

We also visualize the edge attributions produced by both models. We apply Grad to both the original and robust GCNs using the same input graphs. As shown in Figure 7, the highlighted substructures are highly consistent across both models, with only minor variations in intensity. This consistency confirms that the surrogate model preserves the reasoning logic of the original model, ensuring that the explanations generated via the surrogate are valid for the original target.

D.3 Weight Distribution Visualization

To illustrate the edge weight sampling strategy used in our iterative framework, we visualize the distributions of sampled weights for explanatory and non-explanatory edges. Specifically, we draw samples from two truncated Gaussian distributions: $W_1 \sim \mathcal{TN}(\mu_1, \sigma^2; 0, 1)$ for important edges, and $W_2 \sim \mathcal{TN}(\mu_2, \sigma^2; 0, 1)$ for less important edges, where $\mu_1 > \mu_2$. Figure 8 presents the distributions of sampled edge weights for the BA-2motifs, MUTAG, and Fluoride-Carbonyl datasets. In each

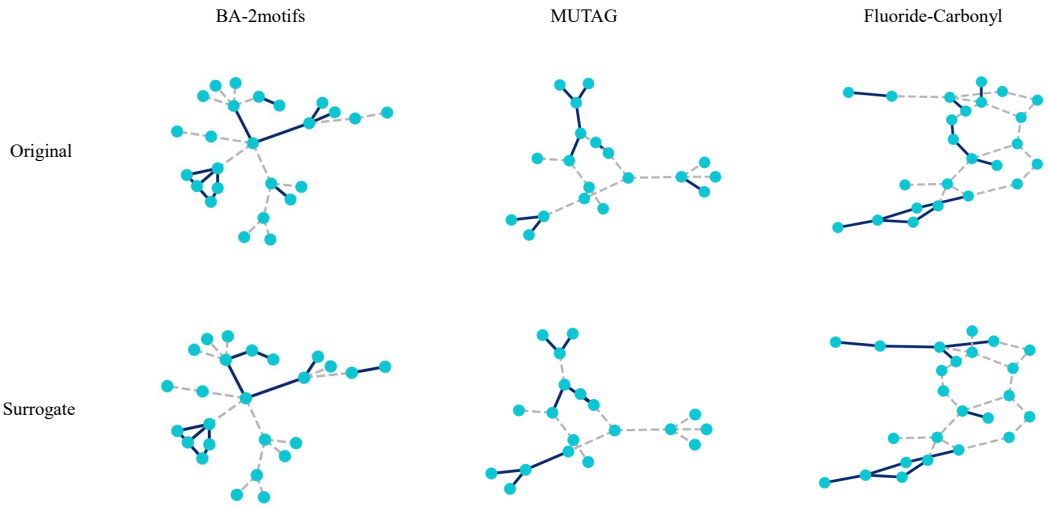
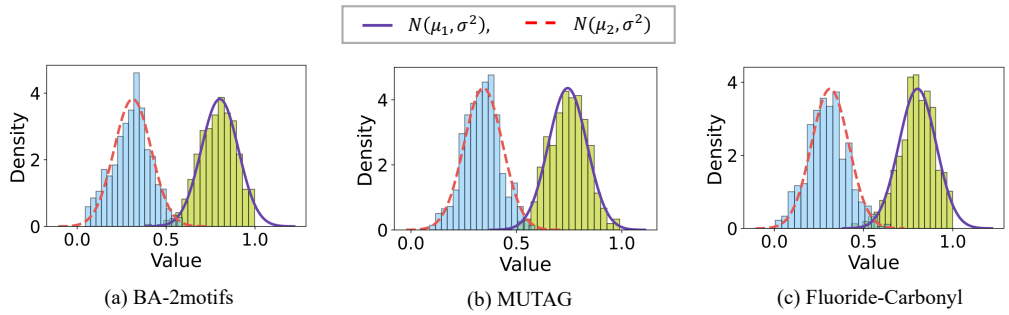


Figure 7: The explanation visualization comparison between original and the robust surrogate GCNs



(a) BA-2motifs (b) MUTAG (c) Fluoride-Carbonyl

Figure 8: Sampled edge weight distributions for explanatory and non-explanatory edges.

subfigure, the blue histogram represents the distribution of weights assigned to non-explanatory edges, and the green histogram corresponds to explanatory edges. We also overlay the histograms with kernel density estimation curves corresponding to the Gaussian distributions $\mathcal{N}(\mu_1, \sigma^2)$ and $\mathcal{N}(\mu_2, \sigma^2)$, shown as solid purple and dashed red curves, respectively.

As shown in the figure, while the actual sampling process is based on truncated Gaussian distributions, the curves demonstrate that the sampled histograms closely resemble the Gaussian distributions. In addition, the two distributions exhibit a clear separation. These results confirm that the Gaussian-based sampling strategy enhances structural signals through explanation-aware weighting, and constrains noise from irrelevant edges, enabling the GNN to learn more effectively.

D.4 Case Study

To qualitatively evaluate the effectiveness of STORE, we conduct case studies on representative samples. Figure 9 visualizes the ground-truth substructures, the explanations produced by the vanilla model, and those generated by our method. In each subfigure, important edges are highlighted in bold brown. For fair comparison, we select the same number of top-ranked edges from the explanation as the number of edges in the ground-truth. From the results, STORE produces explanations that more closely match the ground-truth substructures, whereas the vanilla baseline also selects edges outside these regions.

D.5 Declaration of LLM usage

We used LLMs only for sentence-level language refinement and grammar polishing. The scientific content, methodology, experiments, and conclusions were developed and verified by the authors.

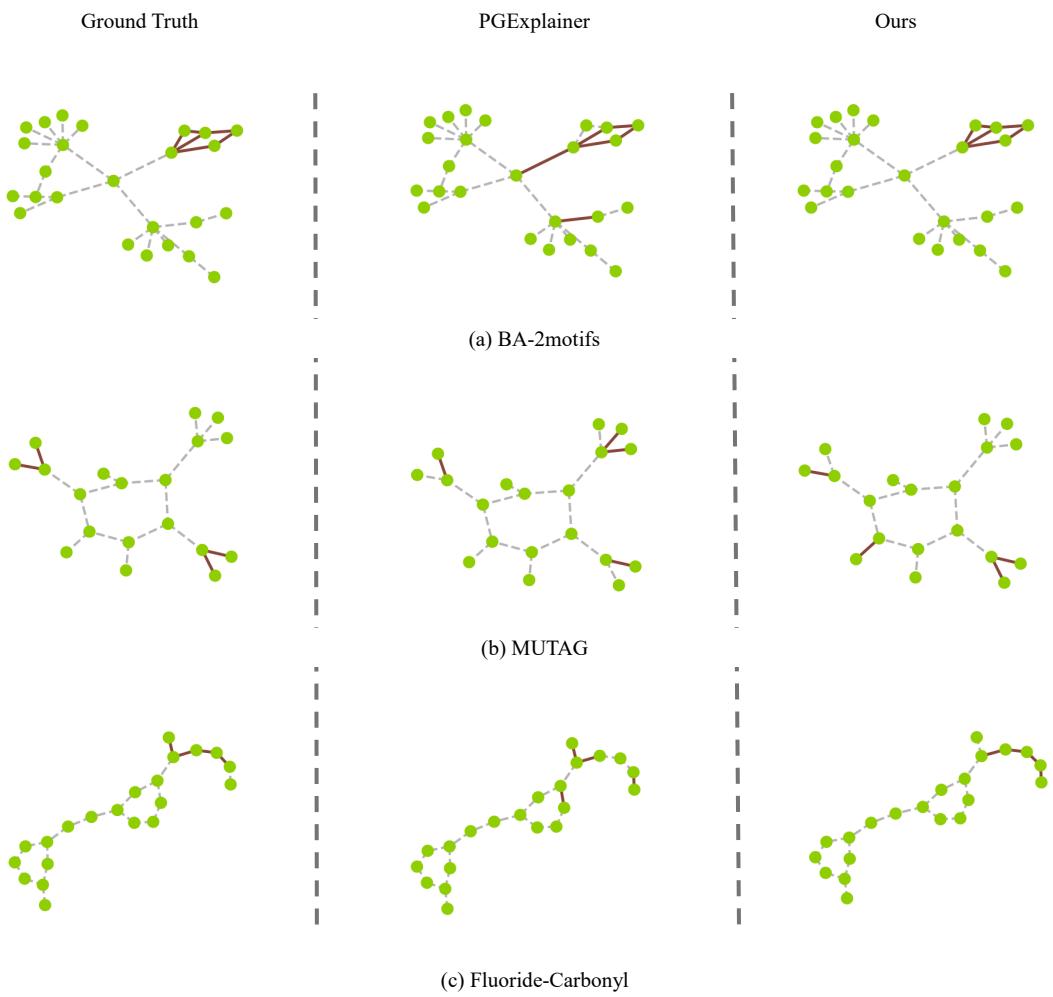


Figure 9: Visual comparison of explanation results.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are accurate and supported by our experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provided the discussion about the limitations at the end of conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the assumption and proof in the main body and appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detailed experimental settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The data and code will be available after this paper gets accepted.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is presented in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conduct our experiments in 10 random seeds and report the mean and std values in the table.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided the information of computing resources in the section of implementation details in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms with the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed, other than helping human beings better understand the black-box GNN models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The data and the baseline models we used are also cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not contain experiments with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not contain experiments with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We provide the LLM usage in the appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.