SSDGExplainer: Structure-Semantic Dual-Guided Explainer for Graph Neural Networks

Zhiqiang Wang[1], **Chenchao Zhang**[1], **Jianqing Liang**[2]*,
Xingwang Zhao[1], **Jiye Liang**[1], **Chuangyin Dang**[3]
[1]Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, China
[2]School of Computer Science and Engineering, Southeast University, Nanjing, China
[3]Department of Systems Engineering, City University of Hong Kong, Hong Kong, China
wangzq@sxu.edu.cn, zhangchenchao@sxu.edu.cn, liangjq@sxu.edu.cn, zhaoxw@sxu.edu.cn, ljy@sxu.edu.cn, mecdang@cityu.edu.hk

Abstract

Post-hoc Graph Neural Networks explainers typically extract a compact subgraph to preserve the model's decision rationale. However, this extraction breaks topological integrity and induces a distribution shift, making predictions on subgraphs unreliable and consequently misleading explainer optimization under out-of-distribution (OOD) conditions. Existing attempts to build in-distribution proxy graphs often assume independence between explanation and background subgraphs and rely on hard splicing, which causes semantic mismatch and boundary discontinuities that degrade explanation reliability. We propose SSDGExplainer, a Structure–Semantic Dual-Guided explainer that formulates proxy-graph generation as conditional modeling under semantic consistency constraints to achieve deep semantic alignment between the explanation subgraph and the generated background, introduces a topology boundary optimization network to smooth structural fractures at the explanation–background interface, and enforces contrastive semantic constraints to prevent semantic drift. Extensive experiments on synthetic and real-world benchmarks demonstrate consistent improvements over state-of-the-art methods in both explanation quality and fidelity.

1 Introduction

Despite the superior performance of Graph Neural Networks (GNNs) in various tasks [1–3], the opacity of their decision-making processes hinders their deployment [4] in high-stakes domains such as molecular property prediction [5], social network analysis [6] and recommendation systems [7, 8]. Consequently, post-hoc explainability for GNNs has emerged as a focal point of current research. This task aims to identify a compact and discriminative subgraph for a given input graph and a pre-trained GNN model, thereby maximally preserving the model's predictive logic regarding the original graph.

Significant progress has been made in instance-level post-hoc explanation methods, which identify substructures within the input graph that most influence model predictions. For instance, GNNExplainer [9] evaluates node and edge importance by perturbing the graph structure, while PGExplainer [10] utilizes global information to select subgraphs that maximize mutual information with predictions. However, these perturbation-based approaches overlook a critical issue: distribution shift [11]. Since the extracted explanatory subgraphs are typically highly sparse and discrete, their data

*Corresponding author.

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

distribution deviates significantly from the original distribution used to train the GNN [12]. This mismatch between training and explanation data renders the pre-trained model's behavior on explanatory subgraphs unreliable. It provides misleading supervisory signals for the optimization of the explainer, severely compromising explanation fidelity.

Recent works have attempted to address this challenge. For example, CGE [13] employs an Expectation-Maximization (EM) framework to alternately update the explanatory subgraph and model parameters adapted to its distribution, though at the cost of increased computational complexity. MixupExplainer [14] adopts data augmentation strategies by mixing explanatory subgraphs with random background subgraphs to construct full samples; however, it assumes that explanations and backgrounds are drawn from independent distributions and can be combined arbitrarily. Proxy-Explainer [15] reconstructs the explanatory subgraph and perturbs the residual graph to generate a proxy graph. Our method augments this design with explicit semantic-anchor conditioning, topology refinement, and contrastive alignment. Although existing proxy graph methods have made preliminary progress in mitigating distribution shifts, they struggle to generate topologically continuous and semantically valid in-distribution samples due to defective generation mechanisms. For instance, in biochemical molecules, strict valence rules and spatial dependencies exist between functional groups (explanations) and carbon skeletons (backgrounds) [16, 17]. Ignoring such intrinsic structural dependencies results in chemically invalid proxy graphs and unnatural topological breaks at splicing interfaces.

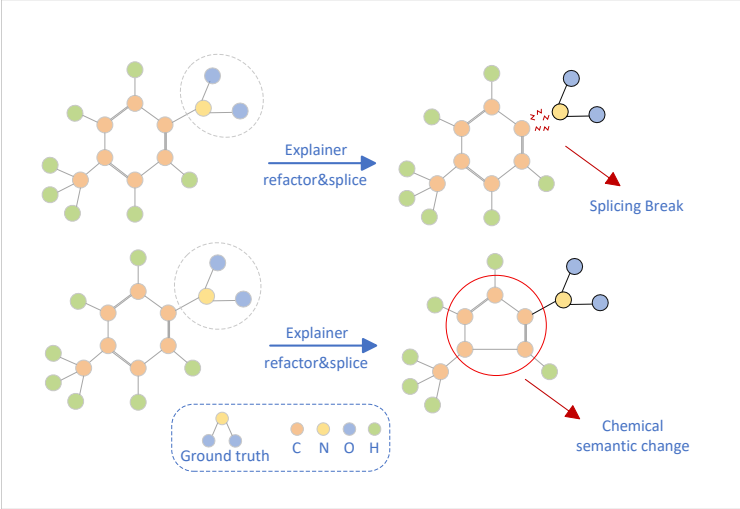


Figure 1: An example illustrating issues arising during the reconstruction of the explanation subgraph and the original graph. Boundary discontinuities occur when the nitro group is spliced with the background subgraph, and the benzene ring in the background is reconstructed as a five-membered carbon ring, leading to altered molecular chemical semantics.

To tackle these challenges, we propose a Structure-Semantic Dual-Guided GNN Explainer (SSDG-Explainer). First, addressing the failure of the independence assumption, we design a semantically constrained conditional proxy graph generation mechanism to capture structural dependencies and ensure semantic similarity between the proxy and original graphs. Second, to mitigate structural fractures caused by mechanical splicing, we introduce a topological boundary optimization network to repair splicing boundaries. Finally, to circumvent the risk of semantic drift during refinement, we utilize contrastive learning to anchor the proxy graph to the original graph's semantics within the feature space. Our contributions are as follows:

- We identify the limitations of existing proxy-graph generation methods, particularly the lack of explicit semantic anchoring and local topology refinement, and introduce a structure-semantic dual-guided generation framework.
- We propose the Structure-Semantic Dual-Guided GNN Explainer. Through the synergy of conditional proxy generation, topological boundary optimization, and contrastive semantic constraints, our method effectively circumvents the risks of structural fracture and semantic drift.

- Experimental results on multiple synthetic and real-world datasets demonstrate that our method outperforms baseline approaches in terms of explanation quality and fidelity, providing more trustworthy explanations.

2 Related Work

2.1 Post-hoc Explanation Methods for GNNs

Post-hoc explanation methods aim to reveal the decision logic of a trained GNN without modifying or intervening in the model. Existing approaches are broadly categorized into instance-level and model-level explanation methods.

Instance-level methods identify substructures within the input graph that most influence the prediction, providing a local explanation for each input graph. These methods fall into four categories: Gradient-based methods estimate feature importance by computing the gradients of the model output with respect to input features (SA [18], GradCAM [19], GNES [20]). However, they only reflect input–output sensitivity rather than true feature relevance. Perturbation-based methods search for explanatory components by perturbing the input graph and observing changes in model predictions (GNNExplainer [9], PGExplainer [10], SubgraphX [21], GSCExplainer [22], MSExplainer [23]). TAGE [24] and EGIB [25] extend this idea to task-agnostic explanations for self-supervised GNNs. Nevertheless, these methods often incur high computational costs and are prone to local optima. Surrogate-based methods train a simple, interpretable model as a proxy for the GNN to provide approximate explanations (PGM-Explainer [26], GraphLIME [27], RelEx [28]). However, discrepancies between the surrogate and the target GNN can reduce the fidelity of the resulting explanations. Decomposition-based methods propagate relevance through the network to identify influential input components, as in GNN-LRP [29] and Excitation BP [19]. Their attribution rules depend on the network architecture and the chosen propagation scheme.

Model-level methods aim to uncover general patterns learned by the entire model, seeking discriminative subgraph structures that are independent of specific inputs. XGNN [30] formulates graph generation as a reinforcement learning problem, using the generated graphs as explanations and representative patterns for target predictions; GNNInterpreter [31] learns discriminative graph patterns by optimizing an objective function specifically designed for model-level explanations; MAGE [32] constructs explanations using motifs as basic units, ensuring chemical validity for human-understandable results in molecular tasks; GnnXemplar [33] iteratively refines explanation rules using large language model–guided representative nodes; GCNeuron [34] extracts high-level semantic subgraph concepts by analyzing internal neuron activation patterns. However, the explanatory graphs produced by these methods may be abstract and difficult for humans to interpret.

2.2 Distribution Alignment Strategies for GNN Explanations

GNNs are typically trained on full graphs, whereas explanations rely on extracted subgraphs [35]. This distribution shift causes unreliable subgraph predictions, misguiding explainer optimization. Existing methods address this issue from three perspectives: data augmentation, generative modeling, and co-optimization.

Data augmentation methods expand the training distribution via synthetic samples. For instance, MixupExplainer [14] mixes explanatory and background subgraphs from different inputs to create distribution-aligned mixed graphs, enhancing explanation credibility.

Generative model-based methods produce explanations within the true data distribution. D4Explainer [36] utilizes a pre-trained generator to learn structural distributions and searches for representative in-distribution subgraph patterns. Similarly, proxy graph methods [15] reconstruct and fuse explanatory and background subgraphs into complete, distribution-consistent proxy graphs to preserve fidelity and consistency.

Co-optimization approaches jointly refine explanations and models. CGE [13] employs an Expectation-Maximization framework [37] to alternately update the explanatory subgraph and distribution-adapted model parameters, avoiding cross-distribution evaluation issues.

Despite their effectiveness, current methods have limitations. Existing augmentation and generation strategies often assume sample independence, yielding statistically plausible but semantically drifted

samples. Moreover, they are prone to producing unnatural fractures at splicing boundaries in terms of topological structure, limiting the further improvement of explanations.

3 Problem Definition

Given an undirected graph $G = (V, E)$ with node features $X \in \mathbb{R}^{|V| \times d}$ and adjacency matrix $A \in \{0, 1\}^{|V| \times |V|}$, let $\hat{y} = f(G) = f(X, A)$ be the prediction of a pre-trained GNN f. Post-hoc instance-level explainability methods aim to identify an explanatory subgraph $G' = A \odot M_A$, where $M_A \in [0, 1]^{|V| \times |V|}$ is an edge importance mask and $\odot$ is element-wise multiplication.

From an information-theoretic perspective, prior works often formulate this within the Graph Information Bottleneck (GIB) framework. It seeks a minimal yet sufficient subgraph by optimizing:

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

where $I(\cdot, \cdot)$ denotes mutual information, G^* is a candidate subgraph, and α is a hyperparameter balancing structural compactness and predictive sufficiency.

4 SSDGExplainer

This section presents the proposed SSDGExplainer in detail. It constructs more reliable in-distribution proxy graphs via structure–semantic dual guidance, thereby providing stable and consistent supervision for the explainer and yielding more trustworthy explanations. Overall, SSDGExplainer consists of three key components: semantic-consistency-constrained conditional proxy-graph generation, a topology boundary optimization network, and contrastive semantic constraints.

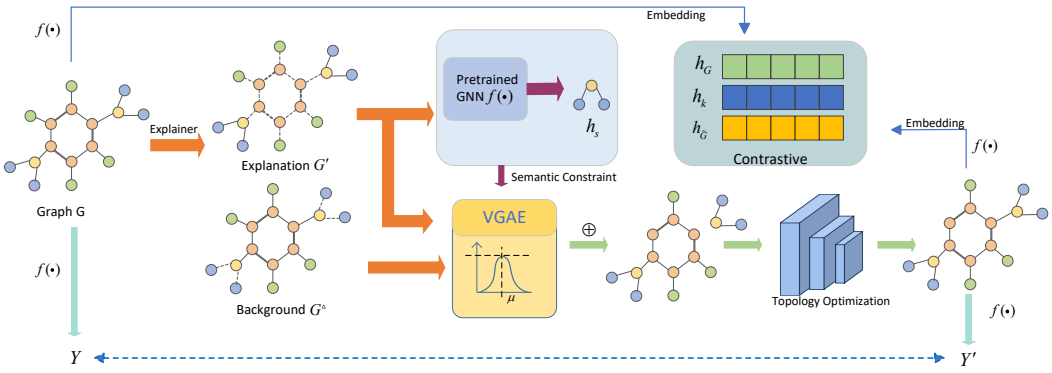


Figure 2: An overview of the SSDGExplainer. The explainer extracts an explanation subgraph and its background from the input graph, and a pretrained GNN provides a semantic representation as guidance. Under this constraint, a conditional generation module produces a semantically consistent proxy graph, which is further refined by a topology optimization module. Finally, a contrastive semantic constraint aligns the proxy graph with the original graph in the embedding space, enabling stable in-distribution supervision for explanation learning.

4.1 Semantic-Constrained Conditional Proxy Graph Generation

To address the issues associated with independent generation, we design a semantic-constrained conditional variational graph autoencoder. This module conditions proxy-graph reconstruction on a semantic representation of the explanatory subgraph, learning the reconstruction distribution of the full graph under the semantic constraints of the explanatory subgraph. First, utilizing the edge importance mask M generated by the explainer Ψ, we calculate the explanatory subgraph $G' = G \odot M$. Then, the frozen encoder of the target GNN is employed to extract node features $z_n = \mathrm{GNN}(G', X)$, and a global semantic representation of the explanatory subgraph is obtained through joint pooling:

$$h_s = [\mathrm{MaxPool}(z_n) \parallel \mathrm{MeanPool}(z_n)], \tag{2}$$

h_s is defined as the *semantic anchor*, serving as a constraint signal to guide the subsequent generation process.

To model the conditional distribution, we employ a dual-branch conditional variational graph autoencoder [38, 39], consisting of two distinct graph autoencoders that process the explanatory subgraph and the background subgraph respectively. For the explanatory subgraph part, we adopt a deterministic inference strategy. We directly treat the mean as the latent variable and obtain the graph-level representation through a joint pooling operation:

$$\begin{aligned} \mu_1, _ &= \mathrm{Encoder}(X, A \odot M), \\ Z_1 &= \mathrm{Concat}\big(\mathrm{MaxPool}(\mu_1), \mathrm{MeanPool}(\mu_1)\big). \end{aligned} \tag{3}$$

Using the mean as the latent representation makes the explanatory branch deterministic and avoids sampling noise during reconstruction.

For the background subgraph, to generate diverse backgrounds, we input the background subgraph into a Variational Graph Autoencoder, introducing Gaussian noise to extract the node feature distribution of the full graph. Subsequently, the graph-level latent variable is obtained using the same processing method:

$$\begin{aligned} \mu_2, \log\sigma^2 &= \text{Encoder}(X, A \odot (1 - M)), \\ z_2 &= \mu_2 + \epsilon \odot \exp\left(\frac{1}{2}\log\sigma^2\right), \quad \epsilon \sim \mathcal{N}(0, I), \\ Z_2 &= \text{Concat}(\text{MaxPool}(z_2), \text{MeanPool}(z_2)). \end{aligned} \tag{4}$$

Afterward, the semantic anchor is concatenated with the representations from both branches to obtain the conditional latent variables:

$$\begin{aligned} Z_{cond1} &= \text{Concat}(Z_1, h_s), \\ Z_{cond2} &= \text{Concat}(Z_2, h_s). \end{aligned} \tag{5}$$

This step forces the latent variable to be anchored around the semantics of the subgraph, compelling the model to consider the structural features of the subgraph during the subsequent generation process, thereby enabling the model to capture global structural dependencies.

In the decoding stage, to flexibly reconstruct the graph structure, a Multi-Layer Perceptron (MLP) is adopted as the decoder to reconstruct the latent variables Z_{cond} into adjacency matrices, which are then subjected to element-wise maximization splicing to obtain the conditional proxy graph:

$$A_{recon} = \max(\text{Dec}(Z_{cond1}), \text{Dec}(Z_{cond2})), \tag{6}$$

where Dec denotes the decoder, and $\max(\cdot)$ denotes element-wise maximization splicing. Our final optimization objective is:

$$L_{recon} = L_{dist} + \beta L_{KL}, \tag{7}$$

where L_{dist} is the cross-entropy loss between G and $\tilde{G}$, L_{KL} is the KL divergence, and β is a hyperparameter controlling the degree of perturbation.

4.2 Topological Boundary Optimization Network

Although conditional generation can produce semantically compatible graph structures, the hard splicing structure between the explanatory subgraph and the background subgraph may introduce unnatural topological fractures at the boundaries, leading to distribution shifts. To address this, we design a parameterized adaptive topological boundary optimization network for local structural adjustment.

First, to accurately capture potential structural inconsistencies in the reconstructed graph, we utilize a two-layer GCN module to extract the local neighborhood features of each node in A_{recon}, allowing nodes to perceive their local topology within the spliced graph:

$$H^{(1)} = \text{ReLU}(\text{GCN}_{conv1}(A_{recon}, X)), \tag{8}$$

$$H^{(2)} = \text{GCN}_{conv2}(A_{recon}, H^{(1)}), \tag{9}$$

where ReLU is the activation function, and $H^{(2)} \in \mathbb{R}^{n \times d}$ represents the second-order local features of each node.

For any edge (u, v) in the graph, we concatenate the features of its endpoints in $H^{(2)}$ as the edge representation $e_{uv} = [h_u \parallel h_v]$. Subsequently, a lightweight edge predictor is used to calculate the adaptive weight adjustment for this edge:

$$\delta_{uv} = \text{Tanh}(\text{MLP}_{edge}(e_{uv})). \tag{10}$$

We then apply bounded correction to the reconstructed adjacency matrix obtained from hard splicing:

$$\bar{A}_{uv} = \text{Clamp}((A_{\text{recon}})_{uv} + \gamma\delta_{uv}, 0, 1), \tag{11}$$

where γ is a hyperparameter controlling the scaling degree. The weight adjustment is fine-tuned, and the Clamp$(\cdot)$ operation restricts edge weights to the $[0, 1]$ interval to prevent overflow.

4.3 Semantic Consistency Constraint

Although the aforementioned conditional reconstruction and structural refinement modules can generate proxy graphs that are topologically reasonable and compatible with the explanation, relying solely on reconstruction loss cannot fully guarantee that the generated proxy graph remains consistent with the original graph in the high-dimensional semantic space. Particularly in graph classification tasks, minor structural perturbations, such as the loss of key edges or the introduction of noisy edges, may lead to semantic drift, reducing explanation credibility. To eliminate this hazard, we introduce a global semantic alignment module.

To ensure that the proxy graph is similar to the original graph in the target model's discrimination, we directly use the encoding layers of the model-to-be-explained as a feature extractor to obtain graph-level embedding representations:

$$h_G = \text{Pool}(f_{enc}(G)), \quad h_{\bar{G}} = \text{Pool}(f_{enc}(\bar{G})), \tag{12}$$

where f_{enc} denotes the pre-trained GNN node feature extraction part, and Pool denotes the pooling operation.

To bridge the semantic distance between h_G and $h_{\bar{G}}$, we adopt a contrastive learning strategy. We treat $(h_G, h_{\bar{G}})$ as a positive pair, while treating other unrelated embeddings in the same batch as negative samples. We optimize this using the InfoNCE loss function [40].

$$\begin{aligned} S &= \sum_{k=1}^{B-1} \exp(\text{sim}(h_G, h_k)/\eta), \\ L_{con} &= -\log \frac{\exp(\text{sim}(h_G, h_{\bar{G}})/\eta)}{\exp(\text{sim}(h_G, h_{\bar{G}})/\eta) + S}, \end{aligned} \tag{13}$$

where sim$(\cdot, \cdot)$ is the cosine similarity function, η is the temperature coefficient, and B is the batch size. This contrastive objective encourages alignment between the original and proxy graph embeddings. Under the standard marginal negative-sampling assumptions, minimizing the expected InfoNCE loss maximizes a lower bound on the mutual information between these embeddings.

4.4 Parameterized Explainer

In this section, we introduce a subgraph learning method. This method first concatenates node features to construct edge features, and then applies MLP to estimate the weight $\omega_{i,j}$ for each edge (i, j). To enhance the learning process, a reparameterization trick is utilized to achieve efficient and differentiable subgraph sampling [41, 42].

Since this paper focuses on graph classification tasks, treating the entire graph as a single instance, importance scores are calculated based on the global structure of the graph. The weight of edge (i, j) is defined as follows:

$$\omega_{i,j} = \text{MLP}_{\Psi}[z_i, z_j], \tag{14}$$

where z_i and z_j are node embedding representations obtained through the graph encoder, and MLP is a multi-layer perceptron parameterized by Ψ.

To obtain differentiable subgraph samples, we use a Binary Concrete relaxation of the Bernoulli distribution. We combine a standard uniform random variable ϵ with the edge weight $\omega_{i,j}$ through a sigmoid function to ensure continuity and differentiability during the sampling process:

$$e_{ij} = \sigma\left(\frac{\log\epsilon - \log(1-\epsilon) + \omega_{i,j}}{\tau}\right), \tag{15}$$

where $\epsilon \sim \mathrm{Uniform}(0,1)$ is a noise term sampled from a uniform distribution, $\sigma(\cdot)$ is the sigmoid function, and τ is the temperature parameter controlling sampling smoothness.

4.5 Model Training

The method proposed in this paper comprises two interdependent core modules: the explainer Ψ, aimed at extracting the discriminative explanatory subgraph G', and the conditional proxy graph generator, aimed at generating a distribution-consistent and semantically aligned proxy graph based on G' to provide accurate supervisory signals.

Since the update of the explainer depends on accurate predictions provided by the proxy graph, and the generation of the proxy graph depends on the explanatory subgraph for reconstruction, a coupling relationship exists between the two. Therefore, following existing works, we adopt an alternating optimization strategy, dividing the overall training process into two cyclic alternating phases: generator optimization and explainer optimization.

Generator Optimization Phase: We fix the parameters of the explainer Ψ and maximize the reconstruction quality of the proxy graph. In this phase, we train for ten iterations to update the parameters of the conditional variational generator and the topological boundary optimization network, ensuring the generation of graphs that are topologically reasonable and conform to the data distribution. We define the loss function for this phase as:

$$L_{recon} = L_{dist} + \beta L_{KL}. \tag{16}$$

Explainer Optimization Phase: We fix the parameters of the conditional variational generator and the continuous topology refinement layer. Using the generated proxy graph as input, we optimize the explainer to identify the optimal explanatory subgraph. In this phase, the explainer parameters are updated once. The total loss function for this phase is defined as:

$$L_{exp} = L_{pred} + \lambda_1 L_{con} + \lambda_2 L_{sparse} + \lambda_3 L_{ent}, \tag{17}$$

where L_{pred} is the cross-entropy loss of the prediction results, L_{con} is the contrastive loss, and L_{sparse} and L_{ent} are sparsity and entropy regularization losses, respectively, used to constrain the compactness and binarization characteristics of the subgraph.

4.6 Theoretical Analysis

Proposition 1 (Variational Objective Capacity)**.** *Let $\mathcal{J}(Q)$ be a common variational objective, and let $\mathcal{H}_{\mathrm{Ind}}$ and $\mathcal{H}_{\mathrm{Cond}}$ be two nonempty feasible families of variational distributions. Assume that $\mathcal{H}_{\mathrm{Ind}} \subseteq \mathcal{H}_{\mathrm{Cond}}$ and that the following supremum values are finite. Define*

$$\mathcal{J}_{\mathrm{Ind}} = \sup_{Q\in\mathcal{H}_{\mathrm{Ind}}} \mathcal{J}(Q), \qquad \mathcal{J}_{\mathrm{Cond}} = \sup_{Q\in\mathcal{H}_{\mathrm{Cond}}} \mathcal{J}(Q).$$

Then

$$\mathcal{J}_{\mathrm{Cond}} \geq \mathcal{J}_{\mathrm{Ind}}. \tag{18}$$

This result compares the best attainable values of the same objective under a nested-family assumption. It does not guarantee strict improvement or convergence of the training procedure to either optimum. We assess embedding alignment using cosine similarity and Euclidean distance, and evaluate explanation faithfulness using robust fidelity metrics. The proof is provided in Appendix C.

5 Experiments

To demonstrate the effectiveness of our proposed method, we conduct comprehensive experiments on five benchmark datasets against nine state-of-the-art baselines. We evaluate explanation quality, distribution shift, robust fidelity, and computational efficiency. Full details of datasets, baselines, and evaluation metrics are provided in Appendix A.

5.1 Comparative Analysis

Table 1 presents the AUC-ROC evaluation results of various explanation methods on both synthetic and real-world datasets. Results for GradCAM, GNNExplainer, PGExplainer, ReFine, MixupExplainer, and ProxyExplainer are taken from [15]. The experimental results demonstrate that our method achieves superior performance across multiple datasets. Particularly on the MUTAG and Fluoride-Carbonyl datasets, the AUC-ROC scores reached 0.994 and 0.776, respectively. Compared with the best baseline methods, these represent improvements of 1.7% and 2.4%, respectively. This validates the enhanced effectiveness of our method in explaining real-world datasets. Furthermore, we conducted a visualization analysis in Appendix B.

Table 1: AUC-ROC scores on five datasets.

Method	BA-2motifs	BA-3motifs	MUTAG	Alkane-Carbonyl	Fluoride-Carbonyl
GradCAM	0.714±0.000	0.709±0.000	0.727±0.000	0.448±0.000	0.694±0.000
GNNExplainer	0.644±0.007	0.511±0.002	0.682±0.009	0.551±0.003	0.574±0.002
PGM-Explainer	0.520±0.007	0.542±0.016	0.611±0.004	0.498±0.022	0.505±0.018
SubgraphX	0.610±0.016	0.631±0.025	0.529±0.021	0.531±0.023	0.495±0.030
ReFine	0.698±0.001	0.629±0.005	0.612±0.004	0.768±0.001	0.571±0.000
PGExplainer	0.734±0.117	0.796±0.010	0.832±0.032	0.660±0.036	0.702±0.018
MixupExplainer	0.906±0.059	0.859±0.019	0.863±0.103	0.811±0.006	0.706±0.013
ProxyExplainer	0.935±0.008	0.960±0.008	0.977±0.009	0.934±0.005	0.758±0.068
GSCExplainer	0.889±0.001	0.742±0.031	0.931±0.077	0.627±0.048	0.727±0.025
Ours	**0.950±0.004**	**0.991±0.001**	**0.994±0.012**	**0.936±0.009**	**0.776±0.005**

Table 2: AP scores and Fidelity metrics on two datasets.

Method	BA-2motifs			MUTAG		
	AP	$\text{Fid}_{\alpha_1,+} \uparrow$	$\text{Fid}_{\alpha_2,-} \downarrow$	AP	$\text{Fid}_{\alpha_1,+} \uparrow$	$\text{Fid}_{\alpha_2,-} \downarrow$
GradCAM	0.664±0.000	0.072±0.000	0.107±0.000	0.247±0.000	0.004±0.000	0.162±0.000
GNNExplainer	0.608±0.004	0.057±0.002	0.132±0.001	0.232±0.001	0.031±0.001	0.148±0.001
ReFine	0.619±0.002	0.060±0.005	0.125±0.001	0.227±0.001	0.003±0.000	0.160±0.001
PGExplainer	0.682±0.117	0.065±0.017	0.126±0.009	0.611±0.024	0.034±0.011	0.148±0.005
MixupExplainer	0.787±0.073	0.074±0.005	0.112±0.003	0.647±0.083	0.037±0.006	0.146±0.003
ProxyExplainer	0.839±0.036	0.086±0.003	**0.106±0.002**	0.756±0.107	0.040±0.002	0.145±0.001
Ours	**0.871±0.011**	**0.088±0.002**	0.110±0.001	**0.900±0.135**	**0.043±0.004**	**0.141±0.002**

As shown in Table 2, our method achieves the best AP scores on both the BA-2motifs and MUTAG datasets, indicating its ability to more accurately rank truly important edges and effectively identify discriminative features. Meanwhile, the results on robust fidelity metrics demonstrate that our method not only precisely localizes key structures but also maintains higher explanation faithfulness under perturbation settings. The baseline results in Table 2 are taken from [15].

Table 3: Time efficiency comparison on two datasets.

Method	BA-2motifs	MUTAG
Time(ms)	Avg. Time	Avg. Time
GNNExplainer	369.7	367.3
PGExplainer	**15.6**	167.7
MixupExplainer	55.2	628.2
ProxyExplainer	19.2	25.3
GSCExplainer	31.2	46.7
Ours	18.2	**24.8**

Table 4: Cosine similarity and euclidean distance.

Dataset	Avg. Cosine (↑)		Avg. Euclidean (↓)	
	(h, h^*)	$(h, \bar{h})$	(h, h^*)	$(h, \bar{h})$
BA-2motifs	0.571	**0.928**	1.210	**0.502**
BA-3motifs	0.686	**0.924**	1.199	**0.590**
MUTAG	0.883	**0.985**	0.975	**0.373**
Alkane-Carbonyl	0.889	**0.950**	0.940	**0.677**
Fluoride-Carbonyl	0.904	**0.919**	0.806	**0.731**

Table 3 shows our method achieves the best or second-best single-instance explanation times across datasets, demonstrating its high efficiency in practical applications. Furthermore, following [15], Table 4 compares the embeddings of the original graph, the ground-truth explanatory subgraph, and

the refined proxy graph. Compared with the embedding of the ground-truth explanatory subgraph h^*, our refined proxy-graph embedding $\bar{h}$ exhibits higher cosine similarity and lower Euclidean distance to the original graph embedding h across all five datasets. These results indicate closer alignment in the target GNN's representation space.

Computational Complexity. We further analyze the computational complexity of SSDGExplainer. Let n and m be the per-graph upper bounds on the numbers of nodes and edges, respectively; let d, K, and B denote the hidden dimension, the number of GNN layers, and the batch size. The graph encoding operations in the conditional generation require $O(K(md + nd^2))$ time, while pooling and conditional concatenation require $O(nd)$. The MLP decoder reconstructs dense adjacency matrices with a time complexity of $O(n^2 d)$ and a memory complexity of $O(n^2)$.

The topology refinement module introduces two GCN layers and a lightweight edge predictor, requiring $O(n^2 d + nd^2)$ and $O(\tilde{m}d^2)$ time, respectively, where $\tilde{m}$ denotes the per-graph upper bound on the number of candidate connections evaluated by the edge predictor. Graph-wise operations are performed independently across the B graphs, whereas the contrastive learning module computes pairwise similarities across the batch at a cost of $O(B^2 d)$. Therefore, the overall per-batch forward-pass complexity is:

$$T_{\text{batch}} = O\left(B[K(md + nd^2) + n^2 d + nd^2 + \tilde{m}d^2] + B^2 d\right). \tag{19}$$

Under the current dense adjacency reconstruction implementation, the quadratic dependence on the number of nodes introduces the main computational and memory overhead as graph size increases.

5.2 Ablation Study

To validate our key components, we evaluate four variants: (1) w/o Cond (removes conditional generation); (2) w/o Ref (removes topology boundary refinement); (3) w/o Lcon (removes the semantic consistency constraint); and (4) Full. As shown in Figure 3, the Full model consistently outperforms all variants, verifying the necessity of each module. Notably, w/o Cond suffers a substantial AUC-ROC drop on MUTAG, indicating that preserving realistic chemical semantics is crucial for reliable real-world explanations.

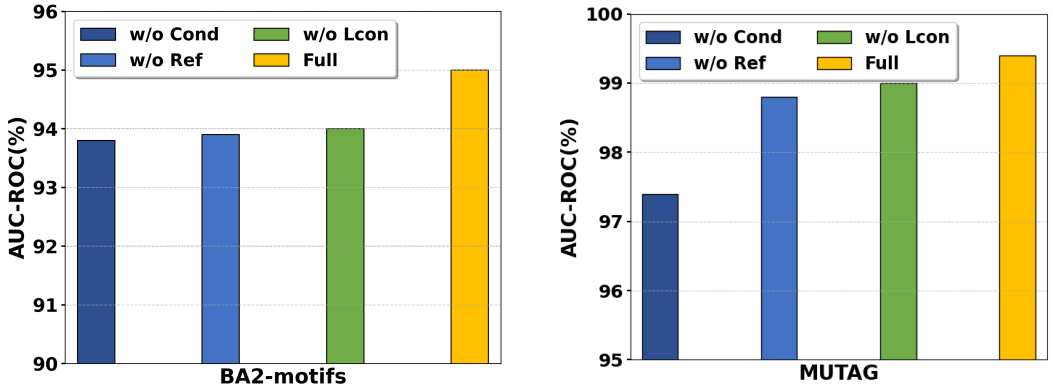


Figure 3: AUC-ROC scores for the ablation of different components

5.3 Parameter Analysis

We further analyzed the impact of the contrastive loss hyperparameter λ_1 and the structural refiner step size γ on model performance using the BA-2motifs and MUTAG datasets. Specifically, we varied λ_1 within the range of $[0.1, 0.9]$ with increments of 0.2. As shown in Figure 4, the highest AUC-ROC is obtained at $\lambda_1 = 0.5$ and $\gamma = 0.1$. These results indicate that the selected parameter values provide the best performance within the investigated search ranges.

To further investigate the sensitivity of the selected hyperparameters around their default values, we conduct fine-grained parameter searches on the BA-2motifs dataset. Specifically, we vary λ_1 around

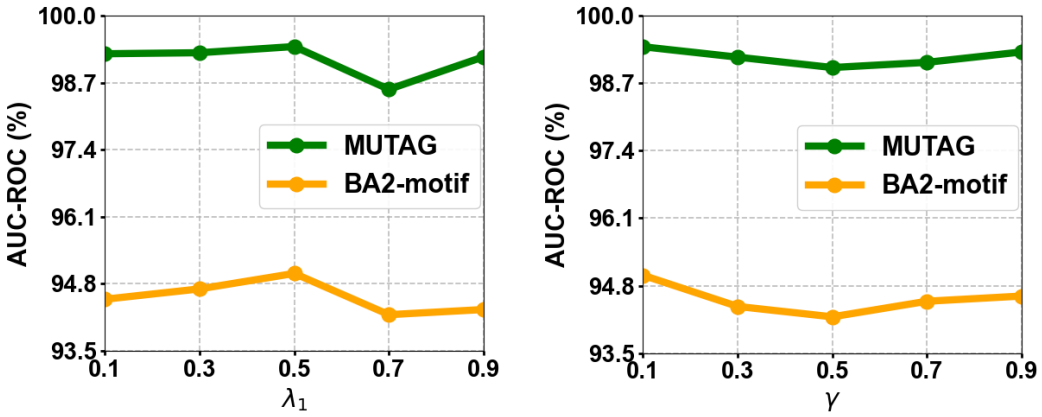


Figure 4: AUC-ROC scores under different parameters

Table 5: Fine-grained parameter sensitivity analysis on BA-2motifs.

(a) λ_1 ($\gamma = 0.1$)		(b) γ ($\lambda_1 = 0.5$)	
λ_1	AUC-ROC	γ	AUC-ROC
0.30	0.946 ± 0.006	0.02	0.941 ± 0.007
0.40	0.947 ± 0.006	0.05	0.943 ± 0.004
0.45	0.947 ± 0.009	0.08	0.948 ± 0.005
0.50	$\mathbf{0.950 \pm 0.004}$	0.10	$\mathbf{0.950 \pm 0.004}$
0.55	0.947 ± 0.006	0.12	0.946 ± 0.008
0.60	0.944 ± 0.005	0.15	0.943 ± 0.007
–	–	0.30	0.941 ± 0.003

0.5 while fixing $\gamma = 0.1$, and evaluate γ around 0.1 while fixing $\lambda_1 = 0.5$. The corresponding results are reported in Table 5.

As shown in Table 5, $\lambda_1 = 0.5$ achieves the highest mean AUC-ROC of 0.950 among the evaluated settings. Similarly, the performance peaks at $\gamma = 0.1$, also achieving an AUC-ROC of 0.950. The neighboring parameter settings exhibit relatively similar performance, suggesting that SSDGExplainer is not overly sensitive to small hyperparameter variations. These results further support the selected default values and indicate that the observed performance is not merely an artifact of the lower boundary of the original search range for γ.

6 Conclusion

We propose SSDGExplainer to address semantic drift and topological boundary fragmentation in proxy-graph methods. By integrating semantic conditioning, topology refinement, and contrastive alignment, SSDGExplainer improves explanation quality and robust fidelity on the evaluated graph-classification benchmarks. Embedding comparisons further indicate closer alignment between proxy and original graphs. While highly effective for graph classification, extending to node-level tasks remains challenging due to overlapping receptive fields. Future work will adapt our mechanism for node classification and explore stronger chemical priors.

Acknowledgments and Disclosure of Funding

This work is supported by the National Natural Science Foundation of China (62272285, 62376142, U25A20529), the Fundamental Research Program of Shanxi Province (202403021211086), and the Shanxi "Sanjin Talents" Support Program (Young Top-notch Talent in Scientific and Technological Innovation). We extend our heartfelt gratitude to the anonymous reviewers for their invaluable and constructive comments. The authors declare no competing interests.

References

[1] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, page 1025–1035, 2017.

[2] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[3] Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. Graph neural networks. *Nature Reviews Methods Primers*, 4(1), 2024.

[4] 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, 2023.

[5] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017.

[6] Zhiwei Guo and Heng Wang. A deep graph neural network-based mechanism for social recommendations. *IEEE Transactions on Industrial Informatics*, 17(4):2776–2783, 2021.

[7] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174, 2019.

[8] Yihong Wu, Le Zhang, Fengran Mo, Tianyu Zhu, Weizhi Ma, and Jian-Yun Nie. Unifying graph convolution and contrastive learning in collaborative filtering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3425–3436, 2024.

[9] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, 2019.

[10] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems*, pages 19620–19631, 2020.

[11] Junfeng Fang, Wei Liu, Yuan Gao, Zemin Liu, An Zhang, Xiang Wang, and Xiangnan He. Evaluating post-hoc explanations for graph neural networks via robustness analysis. In *Advances in Neural Information Processing Systems*, pages 72446–72463, 2023.

[12] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 35(7): 7328–7340, 2023.

[13] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. Cooperative explanations of graph neural networks. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*, pages 616–624, 2023.

[14] 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.

[15] Zhuomin Chen, Jiaxing Zhang, Jingchao Ni, Xiaoting Li, Yuchen Bian, Md Mezbahul Islam, Ananda Mohan Mondal, Hua Wei, and Dongsheng Luo. Generating in-distribution proxy graphs for explaining graph neural networks. In *International Conference on Machine Learning*, pages 7712–7730, 2024.

[16] Krzysztof Maziarz, Henry Richard Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs. In *International Conference on Learning Representations*, 2022.

[17] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pages 2323–2332, 2018.

[18] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[19] Philip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10772–10781, 2019.

[20] Yuyang Gao, Tong Sun, Rishab Bhatt, Dazhou Yu, Sungsoo Hong, and Liang Zhao. Gnes: Learning to explain graph neural networks. In *2021 IEEE International Conference on Data Mining*, pages 131–140, 2021.

[21] 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, 2021.

[22] Zhiqiang Wang, Jiayu Guo, Jianqing Liang, Jiye Liang, Shiying Cheng, and Jiarong Zhang. Graph segmentation and contrastive enhanced explainer for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21393–21401, 2025.

[23] Lutong Wu, Shiying Cheng, Zhiqiang Wang, Jianqing Liang, Peng Song, Xizhao Luo, and Jiye Liang. Multi-scale explainer for graph neural networks. In *International Conference on Machine Learning*, pages 135733–135745, 2026.

[24] Yaochen Xie, Sumeet Katariya, Xianfeng Tang, Edward Huang, Nikhil Rao, Karthik Subbian, and Shuiwang Ji. Task-agnostic graph explanations. In *Advances in Neural Information Processing Systems*, pages 12027–12039, 2022.

[25] Jihong Wang, Minnan Luo, Jundong Li, Yun Lin, Yushun Dong, Jin Song Dong, and Qinghua Zheng. Empower post-hoc graph explanations with information bottleneck: A pre-training and fine-tuning perspective. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 2349–2360, 2023.

[26] Minh N. Vu and My T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, pages 12225–12235, 2020.

[27] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. Graphlime: Local interpretable model explanations for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6968–6972, 2023.

[28] Yue Zhang, David Defazio, and Arti Ramesh. Relex: A model-agnostic relational model explainer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1042–1049, 2021.

[29] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof Schütt, Klaus-Robert Mueller, and Gregoire Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11): 7581–7596, 2022.

[30] 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 Conference on Knowledge Discovery and Data Mining*, pages 430–438, 2020.

[31] Xiaoqi Wang and Han Wei Shen. GNNInterpreter: A probabilistic generative model-level explanation for graph neural networks. In *International Conference on Learning Representations*, 2023.

[32] Zhaoning Yu and Hongyang Gao. MAGE: Model-level graph neural networks explanations via motif-based graph generation. In *International Conference on Learning Representations*, 2025.

[33] Burouj Armgaan, Eshan Jain, Harsh Pandey, Mahesh Chandran, and Sayan Ranu. Gnnxemplar: Exemplars to explanations - natural language rules for global GNN interpretability. In *Advances in Neural Information Processing Systems*, 2025.

[34] Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Liò. Global concept-based interpretability for graph neural networks via neuron analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10675–10683, 2023.

[35] Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. In *Advances in Neural Information Processing Systems*, pages 20437–20448, 2020.

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

[37] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[38] Seong Jin Ahn and MyoungHo Kim. Variational graph normalized autoencoders. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, page 2827–2831, 2021.

[39] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, page 3483–3491, 2015.

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[41] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[42] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

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

[44] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Y. Wang, Wesley Wei Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. Evaluating attribution for graph neural networks. In *Advances in Neural Information Processing Systems*, pages 5898–5910, 2020.

[45] Xiang Wang, Ying-Xin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. In *Advances in Neural Information Processing Systems*, pages 18446–18458, 2021.

[46] 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 *International Conference on Learning Representations*, 2024.

A Experimental Settings

A.1 Datasets

To evaluate our method, we conducted experiments on five benchmark datasets. BA-2motifs [10] contains 1,000 synthetic Barabási–Albert (BA) graphs, each augmented with either a house or a cycle motif. The classification depends on the motif type. BA-3motifs [36] extends this with a third class (grid motif), totaling 3,000 graphs. MUTAG [43] consists of 4,337 molecular graphs labeled as mutagenic or non-mutagenic, with atoms as nodes and bonds as edges. Alkane-Carbonyl and Fluoride-Carbonyl [44] contain 4,326 and 8,671 molecular graphs, respectively, each split into two classes based on the presence of specific functional groups—alkane and carbonyl groups, or fluorine and carbonyl groups—with ground-truth explanations corresponding to these functional group combinations.

Table 6: The statistics of datasets.

Datasets	Avg.nodes	Avg.edges	Features
BA-2motifs	25.00	25.48	10
BA-3motifs	21.92	29.51	4
MUTAG	29.15	60.83	14
Alkane-Carbonyl	21.13	44.95	14
Fluoride-Carbonyl	21.36	45.37	14

A.2 Baselines

To verify the effectiveness of our proposed method, we compare it against nine state-of-the-art graph neural network explanation methods. These methods fall into four categories: gradient-based methods GradCAM [19], perturbation-based methods GNNExplainer [9], PGExplainer [10], SubgraphX [21], ReFine [45] and GSCExplainer [22], surrogate-based methods PGM-Explainer [26], and methods designed to alleviate out-of-distribution issues MixupExplainer [14] and ProxyExplainer [15].

A.3 Evaluation Metrics

The model's explanation quality is evaluated using multiple metrics. AUC-ROC evaluates the model's ability to distinguish positive (ground-truth) from negative explanation edges using predicted edge weights as scores. Average Precision (AP) reflects how well true important edges are ranked at the top, with values closer to 1 indicating higher fidelity to the ground-truth subgraph. Distribution Shift Metrics include Cosine Similarity, which measures directional consistency between embeddings (higher = more similar), and Euclidean Distance, which measures geometric distance (smaller = closer representations). Robust Fidelity is assessed via $\mathrm{Fid}_{\alpha_1,+}$, which quantifies the confidence drop when explanatory edges are randomly removed with probability α_1, and $\mathrm{Fid}_{\alpha_2,-}$, which measures prediction stability when non-explanatory edges are added with probability α_2 [46]. Additionally, to assess computational efficiency, we evaluate the single-instance explanation time (Avg. Time) , which measures the average time required to generate an explanation for a single graph.

B Visualization results

We qualitatively compare the performance of our method with other approaches through visualizations on the BA-3motifs and MUTAG datasets. Experimental results demonstrate that SSDGExplainer generates explanations that are closer to the ground truth compared to baseline methods while effectively avoiding the inclusion of redundant edges.

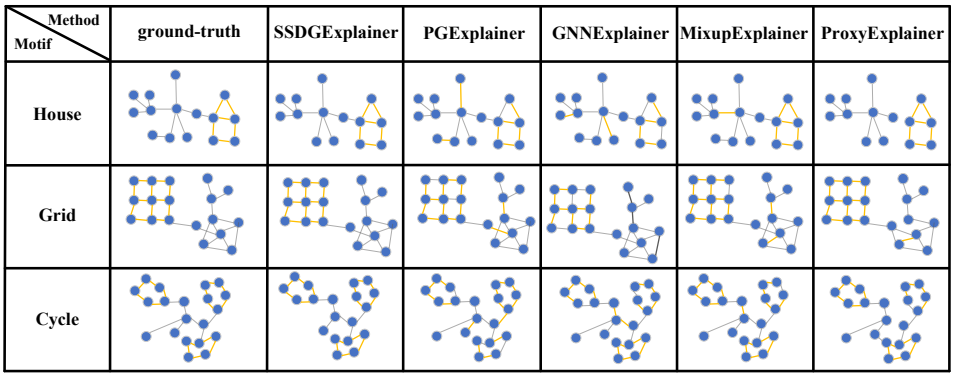


Figure 5: We randomly select one graph from each of the three categories (“house”, “grid”, and “cycle”), where explanations from different methods are highlighted in orange.

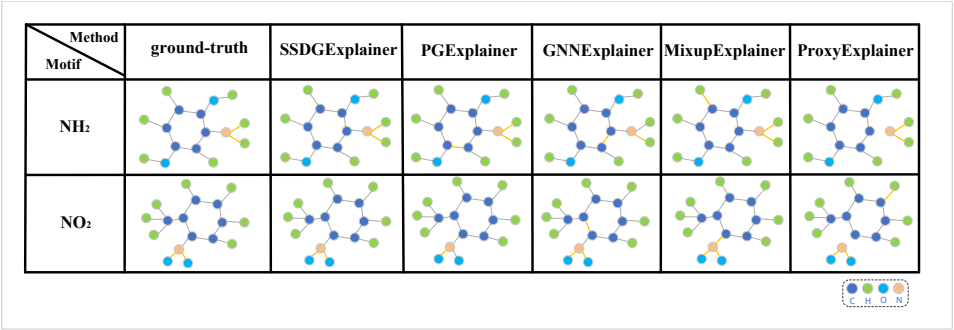


Figure 6: Molecular examples containing amino (“NH_2”) and nitro (“NO_2”) groups. Explanatory edges are highlighted in orange.

C Proof of Proposition 1

For a fixed explanatory graph G' and target label Y, let $p_0(\tilde{G} \mid G')$ be a normalized reference distribution and $r(Y \mid \tilde{G})$ a fixed predictive distribution. Consider the common variational objective

$$\begin{aligned} \mathcal{J}(Q) = \mathbb{E}_{\tilde{G}\sim Q(\cdot|G')}\big[\log r(Y \mid \tilde{G})\big] \\ - \mathrm{KL}(Q(\cdot \mid G') \,\|\, p_0(\cdot \mid G'))\,. \end{aligned} \tag{20}$$

We use the same p_0 and r for both feasible families and restrict attention to distributions for which the objective is well defined. The argument also applies to the expectation of a common objective over a fixed data distribution.

Proof. By assumption, $\mathcal{H}_{\mathrm{Ind}} \subseteq \mathcal{H}_{\mathrm{Cond}}$. Thus, every $Q \in \mathcal{H}_{\mathrm{Ind}}$ is also feasible in $\mathcal{H}_{\mathrm{Cond}}$, and hence

$$\mathcal{J}(Q) \le \sup_{Q' \in \mathcal{H}_{\mathrm{Cond}}} \mathcal{J}(Q') \qquad \text{for every } Q \in \mathcal{H}_{\mathrm{Ind}}.$$

Taking the supremum over $\mathcal{H}_{\mathrm{Ind}}$ yields

$$\mathcal{J}_{\mathrm{Ind}} = \sup_{Q \in \mathcal{H}_{\mathrm{Ind}}} \mathcal{J}(Q) \quad \le \sup_{Q \in \mathcal{H}_{\mathrm{Cond}}} \mathcal{J}(Q) = \mathcal{J}_{\mathrm{Cond}}. \tag{21}$$

The inclusion assumption applies when the conditional architecture contains the comparator architecture as a special case, for example by ignoring its additional conditioning input while preserving the remaining architecture. This is a conditional capacity result, rather than a claim that all existing proxy-graph methods are unconditional. It does not establish that the alternating generator and explainer losses exactly optimize this idealized objective, or that the resulting graphs are chemically valid.

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: We conducted comparative experiments to validate the effectiveness of our approach.

   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 discuss the limitations of our current method and plan to extend it to interpretable node classification in the future.

   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 complete assumptions and rigorous proofs for the theorems in both the main text and the 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: The details provided for our method allow for the reproduction of our main experimental results.

   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 code of this paper will be released later.

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 dataset descriptions and hyperparameter settings for our method are provided in the Experiments section.

   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 conducted multiple runs with different random seeds, and the standard deviation is included in the result tables.

   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: Our experiments were conducted on an NVIDIA GeForce RTX 3090 (24GB), and the single-sample explanation time is provided in the paper.

   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: We confirm that our work complies with the NeurIPS 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: This paper proposes a general post-hoc explanation method for Graph Neural Networks. As a foundational technique, it is not tied to any specific high-stakes application or deployment. Therefore, there is no direct path to negative societal impacts.

    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: Our work focuses on an explainability method for Graph Neural Networks and does not involve the release of pre-trained generative models or large-scale scraped datasets. As the proposed method poses no significant risk for misuse or dual-use, no specific safeguards are required.

    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: We have properly cited the original papers for all datasets and baseline methods used in this work.

    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 release new datasets, benchmarks, pretrained models, or software assets. The main contribution is a graph neural network explanation method evaluated on existing public benchmark datasets.

    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: Our research does not involve crowdsourcing experiments or human subjects. We utilize standard benchmark datasets for Graph Neural Networks without collecting new data from participants.

    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: As our study does not involve human subjects or crowdsourcing, IRB approval is not applicable.

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: [N/A]

Justification: Large Language Models were used solely for proofreading and editing the manuscript to improve language clarity. They were not used as a component of the core methodology or scientific research.

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.