PARADIGM SHIFT OF GNN EXPLAINER FROM LABEL SPACE TO PROTOTYPICAL REPRESENTATION SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-hoc instance-level graph neural network (GNN) explainers are developed to identify a compact subgraph (i.e., explanation) that encompasses the most influential components for each input graph. A fundamental limitation of existing methods lies in the insufficient utilization of structural information during GNN explainer optimization. They typically optimize the explainer by aligning the GNN predictions of input graph and its explanation in the graph label space which inherently lacks expressiveness to describe various graph structures. Motivated by the powerful structural expression ability of vectorized graph representations, we for the first time propose to shift the GNN explainer optimization from the graph label space to the graph representation space. However, the paradigm shift is challenging due to both the entanglement between the explanatory and non-explanatory substructures, and the distributional discrepancy between the input graph and the explanation subgraph. To this end, we meticulously design **IDEA**[1], a universal dual-stage optimization framework grounded in a prototypical graph representation space, which can generalize across diverse existing GNN explainer architectures. Specifically, in the Structural Information Disentanglement stage, a graph tokenizer equipped with a structure-aware disentanglement objective is designed to disentangle the explanatory substructures and encapsulate them into explanatory prototypes. In the Explanatory Prototype Alignment stage, IDEA aligns the representational distributions of the input graph and its explanation unified in the prototypical representation space, to optimize the GNN explainer. Comprehensive experiments on real-world and synthetic datasets demonstrate the effectiveness of IDEA, with the average improvements of ROC-AUC by 4.45% and precision by 48.71%. We further integrate IDEA with diverse explainer architectures and achieve an improvement by up to 10.70%, which verifies its generalizability.

1 INTRODUCTION

Post-hoc instance-level graph neural network (GNN) explainer (Ying et al., 2019; Luo et al., 2020; Schlichtkrull et al., 2021; Wang et al., 2021; Chen et al., 2023; Wang et al., 2023b; Zhang et al., 2023; Zhao et al., 2023; Chen et al., 2024) is a prominent research line to reveal the opaque decision-making mechanism of GNNs utilized in different domains (Fan et al., 2019; He et al., 2020b; Wu et al., 2023b; Liu et al., 2021; Yang et al., 2024b; Li et al., 2020; Mao et al., 2020). For each input graph, post-hoc instance-level GNN explainer aims to identify a compact explanation subgraph that is the most influential to the prediction made by the target GNN model.

Most existing GNN explainers are developed under the *label preserving framework* (Zhao et al., 2023; Zhang et al., 2023) as illustrated in Figure 1(a). Within this framework, a variety of explainer architectures have been proposed. For example, GNNExplainer (Ying et al., 2019) determines the importance of edges and nodes through optimizable soft masks. PGExplainer (Luo et al., 2020) introduces a parametric graph generator to capture global explanatory structures. D4Explainer (Chen et al., 2023) combines the explanation search process with the denoising diffusion model (Ho et al., 2020). V-InFoR (Wang et al., 2023b) and ProxyExplainer (Chen et al., 2024) incorporate the variational graph auto-encoder (Kipf & Welling, 2016) to improve the robustness of GNN explainer.

[1]Our code and datasets are available at https://anonymous.4open.science/r/Idea-2736

Despite promising achievements, the label preserving framework exhibits a fundamental limitation in utilizing structural information to identify the explanation subgraphs, thus restricting the performance of GNN explainers. As shown in Figure 1(a), the label preserving framework optimizes the explainer by aligning the GNN predictions of the input graph and the explanation subgraph in the graph label space. Nevertheless, the graph label inherently lacks expressiveness to capture the characteristics of topological structures (Yang et al., 2024a; Wang et al., 2023a). During the GNN explanation process, the topological structures are critical, especially for complex graph domains such as molecular property prediction (Kazius et al., 2005; Agarwal et al., 2023; Wu et al., 2023a; Funke et al., 2023), where multiple distinct substructures can correspond to the same label.

In order to mitigate the limitation of label preserving framework, we advocate, for the first time, to shift the GNN explainer optimization framework from the graph label space to the graph representation space. Compared with discrete graph labels, the continuous graph representations can provide fine-grained descriptions of topological structures (Sun et al., 2020; Thakoor et al., 2022; Tian et al., 2022; Yang et al., 2024a). Consequently, developing a graph representation space based optimization framework is promising to facilitate the GNN explainer to sufficiently utilize structural information during explanation process. As shown in Figure 1(b), a straightforward implementation of this blueprint is the *direct alignment framework*, which optimizes the explainer by aligning the GNN encoded representations of the input graph and the corresponding explanation. However, the direct alignment framework is far from an effective optimization framework for GNN explainers, due to the following two critical challenges.

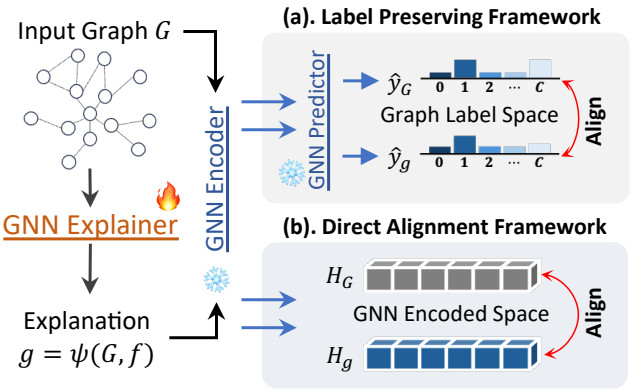


Figure 1: Overview of (a). Currently prevalent label preserving framework and (b). Direct alignment framework in GNN encoded representation space.

The first challenge lies in the entanglement between the explanatory and non-explanatory substructures of the input graph. As revealed by causal inference theory (Wu et al., 2022; Sui et al., 2022), the explanatory substructure causally determines the GNN prediction, while the non-explanatory counterpart merely exhibits statistical correlations. Due to the message passing mechanism (Kipf & Welling, 2017; Veličković et al., 2018; Xu et al., 2019), the GNN encoded representation of the input graph inevitably aggregates explanatory and non-explanatory substructures. Directly aligning the representations of the input graph and the explanation risks misleading the GNN explainer to non-explanatory substructures. The second challenge arises from the distributional discrepancy between the input graph and its explanation subgraph within the GNN encoded representation space. Since the explanation subgraph is a structurally reduced version of the input graph, the explanation representation naturally follows a deviated distribution in the GNN encoded representation space (Zhang et al., 2023; Chen et al., 2024). Simplistically enforcing the representation similarity within the GNN encoded space tends to obscure the most influential subgraph rather than reveal it.

To overcome the challenges above, we propose **IDEA**, a universal dual-stage GNN explainer optimization framework grounded in a prototypical graph representation space, which is generalizable across various existing GNN explainer architectures. Specifically, IDEA consists of a Structural Information Disentanglement stage and an Explanatory Prototype Alignment stage. In the structural information disentanglement stage, we design a hierarchical graph tokenizer equipped with a customized structure-aware disentanglement objective, to disentangle the explanatory substructures from confounding non-explanatory counterpart and then cluster them into prototypical representations. In the explanatory prototype alignment stage, IDEA first unifies the GNN encoded representations of the input graph and the explanation in the prototypical representation space, to mitigate the distributional discrepancy. Subsequently, IDEA aligns the unified representational distributions to optimize the GNN explainer, enabling accurate identification of GNN explanations.

The main contributions of this work are summarized as follows.

- We propose, for the first time, the paradigm shift of GNN explainer optimization framework from the graph label space to the graph representation space. Furthermore, we design IDEA, the first graph representation space based GNN explainer optimization framework.
- We propose a hierarchical graph tokenizer equipped with a structure-aware disentanglement objective, to disentangle the explanatory substructures and encapsulate them into prototypical representations. We formulate a novel explanation identification strategy based on the prototypical representation space, which aligns the unified representational distributions of the input graph and the explanation, to circumvent the deviated distribution of the explanation subgraph.
- Extensive experiments conducted on real-world and synthetic datasets validate the effectiveness of IDEA compared with SOTA GNN explainers, with the average improvements of ROC-AUC by 4.45% and precision by 48.71%. Meanwhile, the consistent superiority of the collaboration between IDEA and various explainer architectures demonstrates the generalizability of IDEA.

2 Notation and Problem Formulation

Notation. We use $G = (A, X)$ with the adjacency matrix $A \in \mathbb{R}^{N\times N}$ and the feature matrix $X \in \mathbb{R}^{N\times D}$ to denote a graph data of N nodes, where D represents the graph feature dimension. If node v_i and node v_j are connected, the element in the i-th row and the j-th column $A_{ij} = 1$, and 0 otherwise. Without losing generality, in this work, we focus on the *graph classification* task (Hu et al., 2022), since node classification can be converted into a computation graph classification problem (Chen et al., 2024). For graph classification, each graph G is associated with a label $y \in \mathbb{R}^{1\times C}$ where C denotes the total number of classes. The target graph neural network model $f(\cdot)$ has been well-trained to predict the class of any given graph G. Generally, the to-be-explained GNN model consists of the following three modules, the feature encoder $f_e(\cdot)$, the pooling function $\mathrm{Pool}(\cdot)$ (e.g., mean pooling and max pooling) (Ying et al., 2018), and the task predictor $f_p(\cdot)$. The GNN prediction procedure can be represented as follows,

$$H_N = f_e(G), H_G = \mathrm{Pool}(H_N), \hat{y} = f_p(H_G), \tag{1}$$

where $H_N \in \mathbb{R}^{N\times d}$ is the matrix of d-dimensional node representations, $H_G \in \mathbb{R}^{1\times d}$ is the pooled graph representation, and $\hat{y}$ is the predicted label. Refer to Appendix A for notation summary.

Problem Formulation. Given a well-trained GNN model $f(\cdot)$ to be explained and an input graph G, the post-hoc instance-level GNN explainer $\psi(\cdot,\cdot)$ aims to identify a compact subgraph $g^* = \psi(G, f) \subset G$, which retains the most influential components during the GNN predicting procedure. Within the label preserving framework, the identified subgraph is reinforced to maintain the original prediction of G. Typically, the optimization objective of the label preserving framework is defined as the mutual information between the predictions of the input graph and the explanation subgraph, i.e., $\mathrm{MI}(f(g), f(G))$. In this work, we shift the GNN explainer paradigm from the label space to the graph representation space, to sufficiently utilize the structural information for GNN explanations.

3 Methodology

Procedurally, **IDEA** consists of two successive stages, the Structural Information Disentanglement and the Explanatory Prototype Alignment, centered on the hierarchical graph tokenizer (HGTokenizer). To tackle the structural entanglement problem, in the first stage, we design a structure-aware disentanglement (SAD) objective for HGTokenizer to stratify the explanatory and non-explanatory substructures. During the disentanglement process, the explanatory substructures are clustered into a collection of explanatory prototypes. In the second stage, based on the HGTokenizer and the prototypes, we first unify the representations of the input graph and the explanation subgraphs into the prototypical representation space, to circumvent the distribution discrepancy problem. Afterwards, the GNN explainer is optimized by aligning the two unified representational distributions.

3.1 Structural Information Disentanglement

In Figure 2, we outline the overview of structural information disentanglement, which empowers the HGTokenizer with the ability to decouple the explanatory substructures from the non-explanatory counterpart. Technically, the HGTokenizer is consist of two cascade-connected graph quantizers

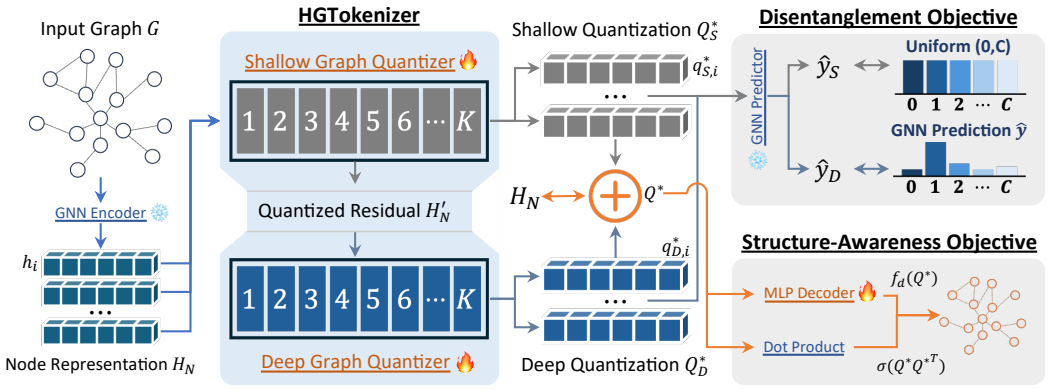


Figure 2: Overview of the Structural Information Disentanglement in IDEA. The node representation H_N is decomposed into two quantization representations Q^*_S and Q^*_D, by the cascade-connected graph quantizers in HGTokenizer. The two quantizers aim to capture the explanatory and non-explanatory substructures respectively, following the guidance of the SAD objective.

(Zeghidour et al., 2021) which take insights of the semantic tokenization (Rajput et al., 2023; Yin et al., 2025), to compactly represent the structural information with discrete codebooks.

Given the node representation matrix H_N, HGTokenizer approximates it based on the shallow and the deep graph quantizers. For the representation to be quantized, the graph quantizer looks up the nearest codeword in the codebook. Since the codebook scale K is significantly smaller than the total number of nodes, it can serve as a collection of prototypes (Dai & Wang, 2025; Zhu et al., 2025) that succinctly summarizes the input representations. Using the representation h_i of node v_i as an example, the cascade quantization procedure of HGTokenizer is formulated as follows,

$$q^*_{S,i} = \mathrm{GQ_S}(h_i) = \arg\min_{q \in \mathcal{C}_S} \mathcal{D}(h_i, q), \; h'_i = h_i - q^*_{S,i} \tag{2}$$

$$q^*_{D,i} = \mathrm{GQ_D}(h'_i) = \arg\min_{q \in \mathcal{C}_D} \mathcal{D}(h'_i, q), \; q^*_i = q^*_{S,i} + q^*_{D,i}, \tag{3}$$

where $\mathrm{GQ_S}(\cdot)$ and $\mathrm{GQ_D}(\cdot)$ denote the shallow graph quantizer and the deep graph quantizer, $\mathcal{C}_S$ and $\mathcal{C}_D$ denote the codebooks of quantizers, q denotes the codeword inside, and $\mathcal{D}(\cdot,\cdot)$ is the distance metric for quantization. The deep graph quantizer takes the residual of the shallow one, in order to spontaneously dichotomize the fused representations encoded by the target GNN model.

The SAD objective utilized to optimize the HGTokenizer consists of three terms, i.e., the structure-awareness objective, the disentanglement objective, and the standard quantization objective. The *structure-awareness objective* $\mathcal{L}_S$ aims to recover the topological structures and node features based on the quantized node representations, enhancing the ability of HGTokenizer to capture the graph structural characteristics. Formally, $\mathcal{L}_S$ is defined as follows,

$$\mathcal{L}_S = \left\| A - \sigma(Q^* Q^{*T}) \right\|_2^2 + \left\| X - f_d(Q^*) \right\|_2^2, \tag{4}$$

where Q^* is the matrix of quantized node representations q^*_i, $\sigma(\cdot)$ is the sigmoid function, and $f_d(\cdot)$ is a linear decoder. The *disentanglement objective* $\mathcal{L}_D$ enforces the prediction of the non-explanatory substructures towards a uniform distribution, and guides the prediction of the explanatory substructures towards the original prediction. Formally, $\mathcal{L}_D$ is defined as follows,

$$\mathcal{L}_D = \mathrm{KL}\left[\hat{y}_S \| \mathcal{U}_C\right] + \mathrm{CrossEntropy}\left(\hat{y}_D, \hat{y}\right), \tag{5}$$

where $\hat{y}_S$ and $\hat{y}_D$ denote the GNN predictions of the shallow and deep quantized representations, respectively. $\mathcal{U}_C$ denotes the uniform distribution in the graph label space.

By minimizing the Kullback-Leibler divergence between $\hat{y}_S$ and $\mathcal{U}_C$, IDEA reinforces the shallow graph quantizer to capture non-explanatory substructures that are unable to determine the GNN decision-making process. Meanwhile, the second term instructs the deep graph quantizer to encapsulate the explanatory substructures that are more influential. Consequently, the codebook $\mathcal{C}_D$ inside

$\mathrm{GQ_D}$ can not only maintain the GNN prediction of the input graph, but also recover the graph topological structures along with $\mathcal{C}_S$. IDEA regards $\mathcal{C}_D$ as a collection of explanatory prototypes which naturally induces a prototypical representation space for the GNN explainer optimization.

In addition, following the standard vector quantization process (van den Oord et al., 2017; Zeghidour et al., 2021), the *quantization objective* $\mathcal{L}_Q$ below is adopted for the basic quantization ability,

$$\mathcal{L}_Q = \left\| H_N - Q^* \right\|_2^2. \tag{6}$$

Hence, the structure-aware disentanglement objective is defined as follows,

$$\mathcal{L}_{\mathrm{SAD}} = \mathcal{L}_D + \lambda_S \cdot \mathcal{L}_S + \lambda_Q \cdot \mathcal{L}_Q, \tag{7}$$

where λ_S, λ_Q are the weighted hyper-parameters. We present a hyper-parameter analysis on the weights λ_S and λ_Q of the structure-aware disentanglement objective in Appendix C.1.

3.2 Explanatory Prototype Alignment

Following the guidance of the SAD objective, the HGTokenizer can disentangle the explanatory information from the fused graph representation encoded by the target GNN. The deep quantizer further encompasses a collection of prototypes to describe the explanatory information. To circumvent the deviated distribution of the explanation subgraphs, we formulate a novel explanation identification strategy on the basis of the prototypical representation space. The overview of the explanatory prototype alignment is illustrated in Figure 3.

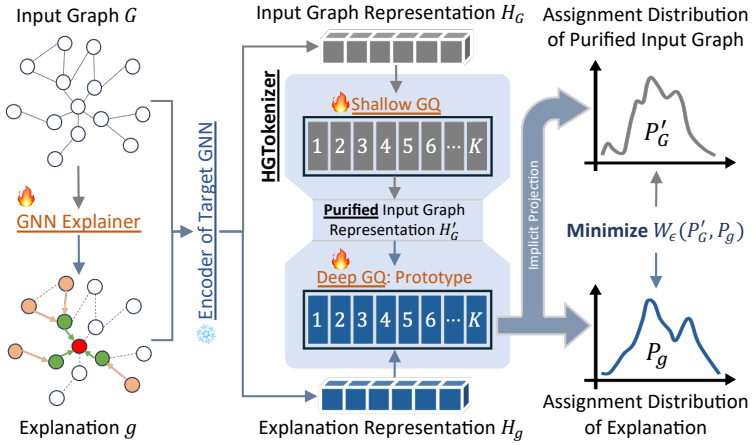


Figure 3: Overview of the Explanatory Prototype Alignment in IDEA. The input graph representation H_G is first purified by the shallow graph quantizer, to eliminate the non-explanatory information. Then, the explanation representation H_g and the purified input graph representation H'_G are implicitly projected into the prototypical space. At last, IDEA aligns the assignment distributions $\mathcal{P}'_G$ and $\mathcal{P}_g$ to optimize the explainer.

Given the target GNN model $f(\cdot)$ to be explained and an input graph G, the explanation subgraph g is generated by $\psi(G, f)$, where ψ denotes the GNN explainer. In our implementation, a typical probabilistic generator, which is well-investigated among the GNN explanation community (Luo et al., 2020; Wang et al., 2021; 2023b; Wang & Shen, 2023), is adopted as the GNN explainer backbone. The implementation details are elaborated in Appendix B.3. Formally, we denote the GNN encoded representation of the input graph as H_G and that of the explanation subgraph as H_g.

To filter out the non-explanatory information from the input graph representation, we feed H_G to the HGTokenizer (i.e., the cascade of $\mathrm{GQ_S}$ and $\mathrm{GQ_D}$), formulated as follows,

$$H_{S,G} = \mathrm{GQ_S}(H_G),\ H'_G = H_G - H_{S,G},\ H_{D,G} = \mathrm{GQ_D}(H'_G), \tag{8}$$

where $H_{S,G}$ is the non-explanatory fraction of the input graph representation and H'_G is the purified input graph representation after removing $H_{S,G}$. For the explanation representation H_g, we directly feed it into the deep graph quantizer $\mathrm{GQ_D}$, formulated as follows,

$$H_{D,g} = \mathrm{GQ_D}(H_g). \tag{9}$$

Based on the quantization procedure of $\mathrm{GQ_D}$, we can implicitly unify the purified representation of the input graph H'_G and the explanation representation H_g into the prototypical representation space, instead of explicit representation projection. To be more specific, the assignment distribution

of the to-be-quantized representation (i.e., H'_G or H_g) over the explanatory codebook $\mathcal{C}_D \in \mathbb{R}^{K\times d}$ is able to indicate its location within the prototypical representation space. Formally, the assignment distributions corresponding to H'_G and H_g are measured as follows,

$$\mathcal{P}'_G = \mathrm{Norm}\big(\mathcal{D}(H'_G, \mathcal{C}_D)\big),\ \mathcal{P}_g = \mathrm{Norm}\big(\mathcal{D}(H_g, \mathcal{C}_D)\big), \tag{10}$$

where $\mathcal{P}'_G$ and $\mathcal{P}_g$ denote the assignment distributions and $\mathrm{Norm}(\cdot)$ normalizes the quantization distance to the probability value. Theoretical justification for this practice is presented in Appendix G. Since the assignment distributions $\mathcal{P}'_G$ and $\mathcal{P}_g$ are identically measured over the implicit prototypical representation space, the distribution discrepancy of the explanation subgraph in the GNN encoded space is ingeniously circumvented.

Subsequently, IDEA adopts the entropy-regularized Wasserstein distance (Reshetova et al., 2024) between $\mathcal{P}'_G$ and $\mathcal{P}_g$ as the optimization objective of the GNN explainer ψ. The Wasserstein distance not only encourages the consistency between the two assignment probabilities $\mathcal{P}'_G$ and $\mathcal{P}_g$, but also is insensitive to the sparsity issue of probabilistic distributions. For the stability of the explainer optimization, IDEA adopts the symmetric variant defined as follow,

$$\mathcal{L}_{\mathrm{IDEA}} = W_\epsilon(\mathcal{P}'_G, \mathcal{P}_g) + \frac{1}{2}\Big(W_\epsilon(\mathcal{P}'_G, \mathcal{P}'_G) + W_\epsilon(\mathcal{P}_g, \mathcal{P}_g)\Big). \tag{11}$$

$$W_\epsilon(\mathcal{P}'_G, \mathcal{P}_g) = \min_{\gamma\in\Pi(\mathcal{P}'_G, \mathcal{P}_g)} \sum_{i,j} \gamma_{ij} S_{ij} + \epsilon \sum_{i,j} \gamma_{ij} \log \gamma_{ij}. \tag{12}$$

$\Pi(\mathcal{P}'_G, \mathcal{P}_g)$ denotes the transport polytope and S denotes the cost matrix defined as follows,

$$\Pi(\mathcal{P}'_G, \mathcal{P}_g) = \big\{\Pi \in \mathbb{R}_+^{K\times K} | \Pi\mathbf{1} = \mathcal{P}'_G, \Pi^T\mathbf{1} = \mathcal{P}_g\big\},\ S_{ij} = (\mathcal{P}'_{G,i} - \mathcal{P}_{g,j})^2. \tag{13}$$

We implement IDEA as a dual-stage framework in order to avoid the counteraction between the optimization terms within $\mathcal{L}_{\mathrm{SAD}}$ and $\mathcal{L}_{\mathrm{IDEA}}$. In Appendix F.2, we further investigate a variant of IDEA where the two stages are conducted jointly.

4 Experiment

To comprehensively validate the practicality of IDEA , we conduct extensive experiments which are designed to investigate the following research questions.

- **RQ1:** How effective is IDEA compared to the label preserving based SOTA baselines?
- **RQ2:** How generalizable is IDEA collaborated with different explainer architectures?
- **RQ3:** How does each component of IDEA influence the overall explanation performance?

Furthermore, we present the hyper-parameter analysis, the explanation visualization, and the time complexity analysis in Appendix C, D, and E, respectively.

4.1 Experimental Setup

Dataset. We evaluate IDEA and baselines on both real-world and synthetic datasets. The evaluated real-world datasets include Mutagenicity (Kazius et al., 2005), Benzene (Sanchez-Lengeling et al., 2020), Fluoride-Carbonyl (Sanchez-Lengeling et al., 2020), and Alkane-Carbonyl (Sanchez-Lengeling et al., 2020). The synthetic datasets is BA-2Motifs (Luo et al., 2020).

Baseline. The baselines include SOTA post-hoc instance-level GNN explainers based on various techniques, i.e., GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), GraphMask (Schlichtkrull et al., 2021), ReFine (Wang et al., 2021), V-InFoR (Wang et al., 2023b), D4Explainer (Chen et al., 2023), MixupExplainer (Zhang et al., 2023), ProxyExplainer (Chen et al., 2024).

Evaluation. Following the standard experimental settings (Luo et al., 2020; Chen et al., 2024), we train a 3-layer Graph Convolutional Network (GCN) model (Kipf & Welling, 2017) on each dataset, as the target model to be explained. To evaluate the explanation quality, we reformulate the explanation task as an edge binary classification task. Edges that belong to the expert-notated ground truth are labeled as positive, and negative otherwise. Hence, we adopt the ROC-AUC score as the main metric to evaluate the explanation performance (Ying et al., 2019; Luo et al., 2020). Refer to Appendix B for the experimental details.

Table 1: Explanation performance (ROC-AUC ↑) of IDEA and eight SOTA post-hoc instance-level GNN explainers on five datasets, in the form of mean$_{\pm \text{std}}$. **Average** reports the mean result over all the evaluated datasets. *Improvement* is defined as $(\texttt{IDEA} - \texttt{Best-Baseline})/\texttt{Best-Baseline}$. The superscript * indicates the improvement is statistically significant with the p-value less than 0.01. **Bold** font and underline highlight the best and the runner-up performance, respectively.

Model	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs	Average
GNNExplainer	0.6155$_{\pm 0.0087}$	0.6863$_{\pm 0.0126}$	0.6884$_{\pm 0.0055}$	0.5399$_{\pm 0.0102}$	0.5619$_{\pm 0.0162}$	0.6184$_{\pm 0.0103}$
PGExplainer	<u>0.7016</u>$_{\pm 0.0201}$	<u>0.8855</u>$_{\pm 0.0023}$	0.7446$_{\pm 0.0086}$	0.8091$_{\pm 0.0209}$	0.8594$_{\pm 0.0072}$	0.8000$_{\pm 0.0115}$
GraphMask	0.6377$_{\pm 0.0083}$	0.5523$_{\pm 0.0062}$	0.6311$_{\pm 0.0139}$	0.5843$_{\pm 0.0028}$	0.6119$_{\pm 0.0035}$	0.6035$_{\pm 0.0068}$
ReFine	0.6833$_{\pm 0.0052}$	0.8720$_{\pm 0.0262}$	0.7293$_{\pm 0.0077}$	0.5600$_{\pm 0.0117}$	0.6115$_{\pm 0.0027}$	0.6912$_{\pm 0.0104}$
V-InfoR	0.6075$_{\pm 0.0149}$	0.6642$_{\pm 0.0112}$	0.6507$_{\pm 0.0162}$	0.6437$_{\pm 0.0169}$	0.7755$_{\pm 0.0243}$	0.6683$_{\pm 0.0156}$
D4Explainer	0.5467$_{\pm 0.0279}$	0.7239$_{\pm 0.0165}$	0.7736$_{\pm 0.0059}$	0.7484$_{\pm 0.0099}$	0.7478$_{\pm 0.0174}$	0.7081$_{\pm 0.0128}$
MixupExplainer	0.5428$_{\pm 0.0074}$	0.5399$_{\pm 0.0020}$	0.7385$_{\pm 0.0043}$	0.5400$_{\pm 0.0002}$	0.8355$_{\pm 0.0129}$	0.6393$_{\pm 0.0035}$
ProxyExplainer	0.6948$_{\pm 0.0035}$	0.8593$_{\pm 0.0127}$	<u>0.9334</u>$_{\pm 0.0033}$	<u>0.8804</u>$_{\pm 0.0126}$	<u>0.8717</u>$_{\pm 0.0028}$	<u>0.8479</u>$_{\pm 0.0068}$
Direct-Align	0.6567$_{\pm 0.0068}$	0.8809$_{\pm 0.0008}$	0.3562$_{\pm 0.0160}$	0.7988$_{\pm 0.0042}$	0.8653$_{\pm 0.0060}$	0.7116$_{\pm 0.0056}$
IDEA	**0.7379**$^{*}_{\pm 0.0084}$	**0.9138**$^{*}_{\pm 0.0002}$	**0.9355**$_{\pm 0.0030}$	**0.8868**$_{\pm 0.0018}$	**0.9541**$^{*}_{\pm 0.0107}$	**0.8856**$^{*}_{\pm 0.0047}$
Improvement	5.17%	3.20%	0.22%	0.73%	9.45%	4.45%

Table 2: Explanation performance (Precision ↑) of IDEA and SOTA baselines across five datasets.

Model	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs	Average
GNNExplainer	0.0736$_{\pm 0.0030}$	0.1901$_{\pm 0.0024}$	0.0104$_{\pm 0.0013}$	0.0652$_{\pm 0.0019}$	0.1373$_{\pm 0.0034}$	0.0953$_{\pm 0.0022}$
PGExplainer	0.1038$_{\pm 0.0067}$	0.4484$_{\pm 0.0041}$	0.0761$_{\pm 0.0077}$	0.3253$_{\pm 0.0176}$	0.6072$_{\pm 0.0016}$	0.3122$_{\pm 0.0072}$
GraphMask	0.0748$_{\pm 0.0070}$	0.1373$_{\pm 0.0075}$	0.0104$_{\pm 0.0082}$	0.0443$_{\pm 0.0029}$	0.2337$_{\pm 0.0043}$	0.1001$_{\pm 0.0036}$
ReFine	0.0833$_{\pm 0.0058}$	0.1951$_{\pm 0.0272}$	0.1304$_{\pm 0.0123}$	0.3027$_{\pm 0.0117}$	0.5054$_{\pm 0.0033}$	0.2434$_{\pm 0.0119}$
V-InFoR	0.1230$_{\pm 0.0075}$	0.3195$_{\pm 0.0134}$	0.1304$_{\pm 0.0010}$	0.2374$_{\pm 0.0019}$	0.1380$_{\pm 0.0161}$	0.1897$_{\pm 0.0075}$
D4Explainer	0.2087$_{\pm 0.0299}$	0.3538$_{\pm 0.0107}$	0.0109$_{\pm 0.0061}$	0.3685$_{\pm 0.0003}$	0.3153$_{\pm 0.0173}$	0.2514$_{\pm 0.0106}$
MixupExplainer	0.0682$_{\pm 0.0083}$	0.1385$_{\pm 0.0018}$	0.0652$_{\pm 0.0038}$	0.2929$_{\pm 0.0034}$	0.3194$_{\pm 0.0105}$	0.1768$_{\pm 0.0034}$
ProxyExplainer	<u>0.3365</u>$_{\pm 0.0058}$	<u>0.5908</u>$_{\pm 0.0135}$	<u>0.3261</u>$_{\pm 0.0035}$	0.1486$_{\pm 0.0032}$	<u>0.6229</u>$_{\pm 0.0089}$	<u>0.4050</u>$_{\pm 0.0067}$
Direct-Align	0.0805$_{\pm 0.0050}$	0.5443$_{\pm 0.0009}$	0.0109$_{\pm 0.0057}$	<u>0.4890</u>$_{\pm 0.0028}$	0.5872$_{\pm 0.0054}$	0.3424$_{\pm 0.0025}$
IDEA	**0.4020**$^{*}_{\pm 0.0063}$	**0.7523**$^{*}_{\pm 0.0003}$	**0.4565**$^{*}_{\pm 0.0161}$	**0.6119**$^{*}_{\pm 0.0183}$	**0.7885**$^{*}_{\pm 0.0201}$	**0.6022**$^{*}_{\pm 0.0119}$
Improvement	19.47%	27.34%	39.99%	25.13%	26.59%	48.71%

4.2 Explanation Performance (RQ1)

The evaluation result of IDEA and SOTA post-hoc instance-level GNN explainers is presented in Table 1, in terms of the ROC-AUC score. *Direct-Align* corresponds to the direct alignment framework in Figure 1(b), which optimizes the GNN explainer by directly aligning the GNN encoded representations of the input graph and the explanation.

The result sufficiently demonstrates the effectiveness of IDEA, which can consistently achieve the supreme performance on all the evaluated datasets. On average, the improvement of IDEA over the best baseline is 4.45%. For the Mutagenicity dataset, which is a complex molecular property prediction dataset, IDEA advances the explanation quality by up to 5.17%, compared to the benchmark-leading baseline PGExplainer (Luo et al., 2020). Despite the primitive explanation identification strategy, *Direct-Align* achieves the second-tier performance among the evaluated explainers, showcasing the considerable potential of GNN explainer optimization framework based on the graph representation space. On the other hand, the inferiority of *Direct-Align* to the top-tier explainers, including PGExplainer, ProxyExplainer (Chen et al., 2024), and IDEA, justify the necessity of further advance on the direct alignment framework.

In light of the critical importance of precision in the GNN explanation evaluation, we further report the result of IDEA and SOTA baselines in Table 2. In general, the average precision of IDEA is 0.6022, achieving a significant improvement by 48.69% over the runner-up ProxyExplainer. Specifically, for the Alkane-Carbonyl dataset, whose ground-truth explanation is the union of an alkane chain (C_nH_{2n+2}) and a carbonyl group (C=O), the improvement of IDEA is the highest over the five evaluated datasets, by up to 39.99%. This advancement demonstrates the ability of IDEA to explain graphs from complex domains. Similarly, the naive contestant *Direct-Align* maintains the moderate position among the evaluated GNN explainers.

Table 3: Explanation performance (ROC-AUC ↑) of IDEA with different explainer architectures.

Model	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs	Average
PGExplainer	0.7016$_{\pm 0.0201}$	0.8855$_{\pm 0.0023}$	0.7446$_{\pm 0.0086}$	0.8091$_{\pm 0.0209}$	0.8594$_{\pm 0.0072}$	0.8000$_{\pm 0.0115}$
+IDEA	**0.7379***$_{\pm 0.0084}$	**0.9138***$_{\pm 0.0002}$	**0.9355***$_{\pm 0.0030}$	**0.8868***$_{\pm 0.0018}$	**0.9541***$_{\pm 0.0107}$	**0.8856***$_{\pm 0.0047}$
Improvement	5.17%	3.20%	25.64%	9.60%	11.02%	10.70%
ReFine	0.6833$_{\pm 0.0052}$	0.8720$_{\pm 0.0262}$	0.7293$_{\pm 0.0077}$	0.5600$_{\pm 0.0117}$	0.6115$_{\pm 0.0027}$	0.6912$_{\pm 0.0104}$
+IDEA	**0.7832***$_{\pm 0.0028}$	**0.8759**$_{\pm 0.0197}$	**0.8428***$_{\pm 0.0018}$	**0.5809***$_{\pm 0.0094}$	**0.6861***$_{\pm 0.0016}$	**0.7538***$_{\pm 0.0067}$
Improvement	14.62%	0.45%	15.56%	3.73%	12.20%	9.05%
V-InfoR	**0.6075**$_{\pm 0.0149}$	0.6642$_{\pm 0.0112}$	0.6507$_{\pm 0.0162}$	0.6437$_{\pm 0.0169}$	0.7755$_{\pm 0.0243}$	0.6683$_{\pm 0.0156}$
+IDEA	0.5734$_{\pm 0.0057}$	**0.6713**$_{\pm 0.0103}$	**0.6776***$_{\pm 0.0008}$	**0.6483**$_{\pm 0.0111}$	**0.7772**$_{\pm 0.0058}$	**0.6696**$_{\pm 0.0059}$
Improvement	-5.61%	1.07%	4.13%	0.71%	0.22%	1.38%
ProxyExplainer	0.6948$_{\pm 0.0035}$	0.8593$_{\pm 0.0127}$	0.9334$_{\pm 0.0033}$	0.8804$_{\pm 0.0126}$	0.8717$_{\pm 0.0028}$	0.8479$_{\pm 0.0068}$
+IDEA	**0.7215***$_{\pm 0.0134}$	**0.8864***$_{\pm 0.0099}$	**0.9509***$_{\pm 0.0099}$	**0.8931**$_{\pm 0.0104}$	**0.8930***$_{\pm 0.0047}$	**0.8690***$_{\pm 0.0081}$
Improvement	3.84%	3.15%	1.87%	1.44%	2.44%	2.48%

4.3 Generalizability across Explainer Architecture (RQ2)

We scrutinize the generalizability of IDEA by integrating various leading GNN explainer architectures and the evaluation result in terms of ROC-AUC is presented in Table 3. In detail, PGExplainer (Luo et al., 2020) adopts a well-investigated subgraph generator based on the concrete distribution (Maddison et al., 2017). ReFine (Wang et al., 2021) implements a subgraph generator for each graph class to capture the contrastive information. V-InFoR (Wang et al., 2023b) introduces a graph vairational auto-encoder (GVAE) to refine the GNN encoded representations for robustness to structural corruptions. ProxyExplainer (Chen et al., 2024) merges a GAVE and a standard graph auto-encoder as the proxy generator to resist distribution discrepancy caused by the explanation subgraph.

The evaluation result sufficiently demonstrates that IDEA is generalizable across four different GNN explainer architectures, with the average improvement by 10.70%, 9.05%, 1.38%, and 2.48%, respectively. The greatest advancement is achieved by the IDEA-enhanced PGExplainer, whose average performance (0.8856) even slightly exceeds the counterpart of the current leading baseline ProxyExplainer (0.8690). For ProxyExplainer that already exhibits strong performance, IDEA can further advance its explanation capacity. The IDEA-enhanced ProxyExplainer provides the best explanations for the Alkane-Carbonyl and Fluoride-Carbonyl datasets, among all the evaluated explainers. The sole exception occurs with the IDEA-enhanced V-InFoR on the Mutagenicity dataset, where the explanation performance drops by 5.61%. The possible reason for the degradation and the marginal improvement of IDEA-enhanced V-InFoR is that V-InFoR is specialized for structurally corrupted graphs, while the evaluated graphs are uncorrupted.

4.4 Ablation Study (RQ3)

In this section, we probe into the influence of each component in the IDEA framework. First, we replace the Wasserstein distance in Eq.11 with the Kullback-Leibler divergence $\mathrm{KL}[P_g \| P'_G]$ and denote the variant as *IDEA-KL*. Afterwards, to validate the effectiveness of the Explanatory Prototype Alignment stage, we implement two variants, *ID-MSE* and *ID-InfoNCE*, which optimize the GNN explainer by aligning the purified representation of input graph H'_G and the explanation representation H_g. *ID-MSE* adopts the mean square error $\mathrm{MSE}(H_g, H'_G)$ as the loss function, and *ID-InfoNCE* adopts the InfoNCE loss function (He et al., 2020a) for in-batch contrastive learning. At last, we omit the Structural Information Disentanglement stage and denote the variant as *EA*. The evaluation result of IDEA and four variants is presented in Figure 4.

We can draw the following conclusions according to the ablation result. First, the distributional discrepancy caused by the explanation subgraph deteriorates the explanation performance. By unifying the representations of the input graph and the explanation subgraph, IDEA and *IDEA-KL* significantly surpass the two variants *ID-MSE* and *ID-InfoNCE* that straightforwardly align the disunified representations. Second, although *EA* is a competitive baseline, structural information disentanglement can further boost the explanation performance. *EA* is inferior to the unabridged IDEA, with an average performance gap by 0.0682. Third, compared with KL divergence, Wasserstein distance is more suitable for GNN explainer optimization in the prototypical representation space. IDEA consistently outperforms *IDEA-KL*, with an average improvement of 3.33%.

In Appendix F, we investigate the cooperation of IDEA and the label preserving framework, where the optimization objective is defined as the convex combination of $\mathcal{L}_{\mathrm{IDEA}}$ and $\mathrm{MI}(f(g), f(G))$.

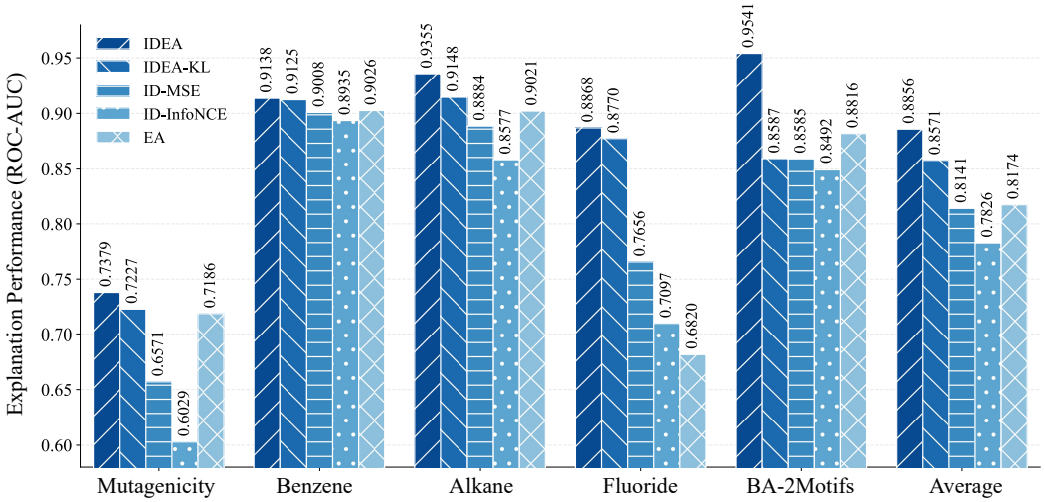


Figure 4: Explanation performance (ROC-AUC ↑) of IDEA and its four variants.

5 Related Work

Post-hoc Instance-level GNN Explainers have become a primary approach to explain GNN models, with various methods proposed to identify the critical substructures responsible for predictions. GNNExplainer (Ying et al., 2019) perturbs graph components to estimate their importance. PGExplainer (Luo et al., 2020) introduces a parametric generator to capture global explanatory signals. GraphMask (Schlichtkrull et al., 2021) and ReFine (Wang et al., 2021) advance explanations through edge selection and multi-grained analysis, respectively. D4Explainer (Chen et al., 2023) adopts diffusion models to generate explanations from random noise. MixupExplainer (Zhang et al., 2023) leverages data augmentation to resist distribution shift. V-InFoR (Wang et al., 2023b) and ProxyExplainer (Chen et al., 2024) employ variational autoencoders to enhance explanation robustness.

Prototype-based GNN explanation methods aim to improve the intrinsic interpretability of GNN models. ProtGNN (Zhang et al., 2022) introduces prototype learning into GNNs, enabling class-specific prototypical subgraphs to serve as intuitive analogical explanations. PAGE (Shin et al., 2024) extends this idea to model-level interpretability by constructing a global prototype dictionary in latent space, offering explanations of the overall decision boundary. Ragno et al. (2024) refine prototype separability and semantic consistency through enhanced architectures and training strategies. Dai & Wang (2025) further integrates prototype learning with self-explaining mechanisms, jointly optimizing prediction and interpretability.

Vector Quantization (VQ) techniques provide a powerful way to discretize continuous embeddings into discrete codewords. Since the number of the codewords tends to be significantly smaller than that of the embeddings to be quantized, VQ thereby clusters similar embeddings into a collection of prototypes. Early successes in domains such as audio (Zeghidour et al., 2021) highlight the capacity of VQ to encode complex signals into compact tokens. Combined with large language models, VQ facilitates the revolution of generative recommender systems (Rajput et al., 2023; Yin et al., 2025). Recent advances in graph community (Yang et al., 2024a) extend this principle to graph data, developing the structure-aware codebooks by tokenizing local substructures.

6 Conclusion

We for the first time propose the paradigm shift of GNN explainer optimization framework from the graph label space to the graph representation space, and we design IDEA, the first GNN explainer optimization framework grounded in a prototypical graph representation space. IDEA consists of a structural information disentanglement stage, which disentangles and encapsulates the explanatory substructures into prototypes, and an explanatory prototype alignment stage, which aligns the representational distributions of input graph and explanation unified in the prototypical space. Extensive experiments demonstrate the effectiveness and generalizability of IDEA.

REFERENCES

Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating Explainability for Graph neural Networks. *Scientific Data*, 10, 2023.

Kenza Amara, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. GraphFramEx: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks. In *Learning on Graphs Conference*, 2022.

Jialin Chen, Shirley Wu, Abhijit Gupta, and Rex Ying. D4Explainer: In-distribution Explanations of Graph Neural Network via Discrete Denoising Diffusion. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2023.

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 *Proceedings of the International Conference on Machine Learning*, 2024.

Enyan Dai and Suhang Wang. Towards Prototype-Based Self-Explainable Graph Neural Network. *ACM Transactions on Knowledge Discovery from Data*, 19(2), 2025.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph Neural Networks for Social Recommendation. In *Proceedings of The World Wide Web Conference*, 2019.

Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. Zorro: Valid, Sparse, and Stable Explanations in Graph Neural Networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(8), 2023.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020a.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020b.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017.

Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and Validation of Toxicophores for Mutagenicity Prediction. *Journal of Medicinal Chemistry*, 48(1), 2005.

Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2016.

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

Xiangsheng Li, Maarten de Rijke, Yiqun Liu, Jiaxin Mao, Weizhi Ma, Min Zhang, and Shaoping Ma. Learning Better Representations for Neural Information Retrieval with Graph Information. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, 2020.

Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and Choose: A GNN-based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the Web Conference*, 2021.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized Explainer for Graph Neural Network. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

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.

Kelong Mao, Xi Xiao, Jieming Zhu, Biao Lu, Ruiming Tang, and Xiuqiang He. Item Tagging for Information Retrieval: A Tripartite Graph Neural Network based Approach. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, , and Heiko Hoffmann. Explainability Methods for Graph Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Alessio Ragno, Biagio La Rosa, and Roberto Capobianco. Prototype-Based Interpretable Graph Neural Networks. *IEEE Transactions on Artificial Intelligence*, 5(4), 2024.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender Systems with Generative Retrieval. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2023.

Daria Reshetova, Yikun Bai, Xiugang Wu, and Ayfer Özgür. Understanding entropic regularization in GANs. *The Journal of Machine Learning Research*, 25(1), 2024.

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 *Proceedings of the International Conference on Neural Information Processing Systems*, 2020.

Michael Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking. In *International Conference on Learning Representations*, 2021.

Yong-Min Shin, Sun-Woo Kim, and Won-Yong Shin. PAGE: Prototype-Based Model-Level Explanations for Graph Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10), 2024.

Teague Sterling and John J. Irwin. ZINC 15 – Ligand Discovery for Everyone. *Proceedings of the International Conference on Neural Information Processing Systems*, 55(11), 2015.

Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal Attention for Interpretable and Generalizable Graph Classification. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*, 2020.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Veličković, and Michal Valko. Large-Scale Representation Learning on Graphs via Bootstrapping. In *International Conference on Learning Representations*, 2022.

Yijun Tian, Chuxu Zhang, Zhichun Guo, Xiangliang Zhang, and Nitesh V. Chawla. NOSMOG: Learning Noise-robust and Structure-aware MLPs on Graphs. In *NeurIPS 2022 New Frontiers in Graph Learning Workshop*, 2022.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86), 2008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018.

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 ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023a.

Senzhang Wang, Jun Yin, Chaozhuo Li, Xing Xie, and Jianxin Wang. V-InFoR: A Robust Graph Neural Networks Explainer for Structurally Corrupted Graphs. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2023b.

Xiang Wang, Ying-Xin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards Multi-Grained Explainability for Graph Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2021.

Xiaoqi Wang and Han Wei Shen. GNNInterpreter: A Probabilistic Generative Model-Level Explanation for Graph Neural Networks. In *International Conference on Learning Representations*, 2023.

Fang Wu, Siyuan Li, Xurui Jin, Yinghui Jiang, Dragomir Radev, Zhangming Niu, and Stan Z. Li. Rethinking Explaining GNNs via Non-parametric Subgraph Matching. In *Proceedings of the International Conference on Machine Learning*, 2023a.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph Neural Networks in Recommender Systems: A Survey. *ACM Computing Surveys*, 55(5), 2023b.

Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering Invariant Rationales for Graph Neural Networks. In *International Conference on Learning Representations*, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2019.

Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, Bin CUI, Muhan Zhang, and Jure Leskovec. VQGraph: Rethinking Graph Representation Space for Bridging GNNs and MLPs. In *International Conference on Learning Representations*, 2024a.

Ling Yang, Jiayi Zheng, Heyuan Wang, Zhongyi Liu, Zhilin Huang, Shenda Hong, Wentao Zhang, and Bin Cui. Individual and Structural Graph Information Bottlenecks for Out-of-Distribution Generalization. *IEEE Transactions on Knowledge and Data Engineering*, 36(2), 2024b.

Jun Yin, Zhengxin Zeng, Mingzheng Li, Hao Yan, Chaozhuo Li, Weihao Han, Jianjin Zhang, Ruochen Liu, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, Shirui Pan, and Senzhang Wang. Unleash LLMs Potential for Sequential Recommendation by Coordinating Dual Dynamic Index Mechanism. In *Proceedings of the ACM on Web Conference*, 2025.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical Graph Representation Learning with Differentiable Pooling. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2018.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating Explanations for Graph Neural Networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2019.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31, 2021.

Jiaxing Zhang, Dongsheng Luo, and Hua Wei. MixupExplainer: Generalizing Explanations for Graph Neural Networks with Data Augmentation. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. ProtGNN: Towards Self-Explaining Graph Neural Networks. In *AAAI Conference on Artificial Intelligence*, 2022.

Tianxiang Zhao, Dongsheng Luo, Xiang Zhang, and Suhang Wang. Faithful and Consistent Graph Neural Network Explanations with Rationale Alignment. *ACM Transactions on Intelligent Systems and Technology*, 14(5), 2023.

Zhiqiang Zhong, Yangqianzi Jiang, and Davide Mottin. On the Robustness of Post-hoc GNN Explainers to Label Noise. In *Proceedings of the Learning on Graphs Conference*, 2023.

Zhijie Zhu, Lei Fan, Maurice Pagnucco, and Yang Song. Interpretable Image Classification via Non-parametric Part Prototype Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

A NOTATION

In Table 4, we summarize the notations used throughout this manuscript and their descriptions.

Table 4: Notations and corresponding descriptions.

Notation	Description
G	Graph instance
A, X	Adjacency matrix, node feature matrix
N	Number of graph nodes
D	Node feature dimension
v_i	The i-th node
A_{ij}	Element at the i-th row, j-th column of adjacency matrix A
$y, \hat{y}$	Graph label, GNN prediction
C	Number of graph classes
f, f_e, f_p	Graph neural network model, encoder of f, predictor of f
H_N, H_G	Node representation matrix, graph representation vector
h_i	Node representation of v_i
ψ	Post-hoc instance-level GNN explainer
g, g^*	Explanatory subgraph (i.e., explanation)
$\mathcal{L}_\psi$	Label preserving loss of ψ
Ω	Regularization term
λ_Ω	Weighted hyper-parameter of Ω
$\mathrm{GQ_S}, \mathrm{GQ_D}$	Shallow graph quantizer, deep graph quantizer
$\mathcal{C}_S, \mathcal{C}_D$	Codebook of $\mathrm{GQ_S}$, codebook of $\mathrm{GQ_D}$
$\mathcal{D}$	Distance metric of vector quantization
q, q^*	Codeword, the nearest codeword
q_S^*, q_D^*	The nearest codeword in $\mathrm{GQ_S}$, the nearest codeword in $\mathrm{GQ_D}$
h_i'	Residual representation after $\mathrm{GQ_S}$ quantization
$\mathcal{L}_Q, \mathcal{L}_S, \mathcal{L}_D$	Quantization objective, structure-aware objective, disentanglement objective
Q^*	Quantization matrix
σ	Sigmoid function
f_d	Linear decoder
$\hat{y}_S$	GNN prediction of $\mathrm{GQ_S}$ quantized representation
$\hat{y}_D$	GNN prediction of $\mathrm{GQ_D}$ quantized representation
$\mathcal{U}_C$	Uniform distribution
$\mathcal{L}_{\mathrm{SAD}}$	Structure-aware disentanglement objective
λ_Q, λ_S	Weighted hyper-parameter of $\mathcal{L}_Q$ and $\mathcal{L}_S$
H_G, H_g	Representation of original graph, representation of explanation
H_G'	Residual representation of H_o after $\mathrm{GQ_S}$ quantization
$\mathcal{P}_G', \mathcal{P}_g$	Assignment probability of H_o' and H_e representation of explanation
W_ϵ	Entropy-regularized Wasserstein distance
Π, S	Transport polytope and cost matrix of W_ϵ
$\mathcal{L}_{\mathrm{IDEA}}$	IDEA optimization objective
$\mathcal{L}_{\mathrm{Mix}}$	Weighted combination of $\mathcal{L}_\psi$ and $\mathcal{L}_{\mathrm{IDEA}}$
α	Weighted parameter in $\mathcal{L}_{\mathrm{Mix}}$

B EXPERIMENTAL DETAIL

Here, we elaborate the details of the evaluated datasets, baselines, and IDEA implementation.

B.1 DATASET

The dataset details are introduced as follows and the dataset statistics are summarized in Table 5.

- **Mutagenicity** (Kazius et al., 2005) is a collection of molecular compounds labeled for their ability to cause mutations (i.e., mutagenic vs. non-mutagenic), widely used in cheminformatics and toxicology for developing predictive models. Mutagenicity contains 4,337 molecule graphs with NO_2 and NH_2 chemical groups notated as ground truth explanations.

Table 5: The statistics of the evaluated datasets.

Statistic	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs
Graphs	4,337	12,000	4,326	8,671	1,000
Average Nodes	30.32	20.58	21.13	21.36	25.00
Average Edges	30.77	43.65	44.95	45.37	50.90
Node Features	14	14	14	14	10
GNN Accuracy	0.8300	0.9054	0.9620	0.9340	1.0
GT Explanation	NO_2, NH_2	Benzene	Alkane + C=O	F^- + C=O	Motif

- **Benzene** (Agarwal et al., 2023) is a binary classification dataset of 12,000 molecular graphs sampled from ZINC15 (Sterling & Irwin, 2015). The goal is to decide whether a molecule contains at least one benzene ring. Within this dataset, the atoms that constitute the benzene ring serve as the ground-truth explanation. Multiple disjoint benzene rings are treated as separate explanations.
- **Alkane-Carbonyl** (Agarwal et al., 2023) is a binary classification set of 4,326 molecular graphs. A positive label marks a molecule that simultaneously contains an unbranched alkane chain and a carbonyl (C=O) group. The ground-truth explanation is defined as the arbitrary union of these two functional fragments present in the structure.
- **Fluoride-Carbonyl** (Agarwal et al., 2023) contains 8,671 molecular graphs. A molecule is labeled positive only if it contains both a fluoride atom (F) and a carbonyl group (C=O). The explanation is defined as the arbitrary union of these two functional units found in the structure.
- **BA-2Motifs** (Ying et al., 2019) is a synthetic binary class dataset designed to benchmark GNN explanation methods. Each graph is labeled by the presence of either a house or a cycle motif, and the respective motif itself provides the ground truth explanation for that class.

We present the accuracy of the to-be-explained GNN model for each dataset in Table 5 as well.

B.2 Baseline

The evaluated baselines include eight SOTA post-hoc instance-level GNN explainers based on various search strategies. The details are introduced as follows.

- **GNNExplainer** (Ying et al., 2019) is a GNN explainer based on data perturbation that jointly masks edges and node features, then scores their contribution by searching for a subgraph G_S that maximizes the mutual information with the model's overall prediction $\hat{y}$.
- **PGExplainer** (Luo et al., 2020) masks graph topology and uses a learnable neural network to assign edge importance scores, optimizing the same mutual-information objective.
- **GraphMask** (Schlichtkrull et al., 2021) learns an amortized classifier that predicts whether the edge can be dropped (replaced by a learned baseline) for every edge in every GNN layer, without changing the model output, yielding a sparse post-hoc explanation.
- **ReFine** (Wang et al., 2021) adopts a pre-train and fine-tune strategy to probe GNN decisions, delivering multi-granularity insights into the model's reasoning process.
- **V-InfoR** (Wang et al., 2023b) is a robust GNN explainer specialized for the structurally corrupted graphs, which employs the variational inference to learn the robust graph representations and generalizes the GNN explanation exploration to a graph information bottleneck (GIB) optimization task without any predefined rigorous constraints.
- **D4Explainer** (Chen et al., 2023) a generative explainer for counterfactual and model-level explanations based on a discrete denoising diffusion model, which frames the explanation problem as a distribution learning task for more reliable explanations with better in-distribution property.
- **MixupExplainer** (Zhang et al., 2023) addresses the distribution shifting issue by mixing up the explanation with a randomly sampled base graph structure.
- **ProxyExplainer** (Chen et al., 2024) extends the GIB by innovatively including in-distributed proxy graphs and derives a tractable objective function for practical implementations, where two graph auto-encoders are utilized to generate proxy graphs.

B.3 IDEA IMPLEMENTATION

Within the main experiment, we adopt a well-investigated subgraph generator as the backbone implementation of the IDEA framework. According to the Gilbert random graph theory, an arbitrary graph G can be represented as a random graph variable, and each edge of G is associated with a binary random variable r to reveal its existence. Additionally, the existence of one edge is conditionally independent of the other edges. $\varepsilon_{ij} = 1$ means there is an edge (i, j) from v_i to v_j, otherwise $\varepsilon_{ij} = 0$. Hence, an arbitrary graph G can be represented as follows,

$$p(G) = \prod_{(i,j)} p(\varepsilon_{ij}). \tag{14}$$

A common instantiation of the binary variable ε_{ij} is the Bernoulli distribution $\varepsilon_{ij} \sim \mathrm{Bern}(\varrho_{ij})$, where $\varrho_{ij} = p(\varepsilon_{ij} = 1)$ is the probability of edge (i, j) existing in the random graph G. Since the Bernoulli distribution cannot be directly optimized, we introduce categorical reparameterization (Jang et al., 2017) to ε_{ij}. The continuous relaxation of ε_{ij} can be formulated as follows,

$$\varepsilon_{ij} = \sigma\Big(\frac{\log \mathcal{U} - \log\big(1-\mathcal{U}\big) + \mu_{ij}}{\tau}\Big),\ \mu_{ij} = \log\frac{\varrho_{ij}}{1-\varrho_{ij}},\ \mathcal{U} \sim \mathrm{Uniform}(0, 1). \tag{15}$$

where τ controls the approximation between the relaxed distribution and $\mathrm{Bern}(\varrho_{ij})$. When τ approaches 0, the limitation of Eq.(15) is $\mathrm{Bern}(\varrho_{ij})$.

According to Eq.(15), the Bernoulli parameter ϱ_{ij} is associated with the parameter μ_{ij}. To enable end-to-end optimization, we use a multi-layer perceptron (MLP) to compute μ_{ij}. The MLP takes the GNN node representation as input and concatenates the representations of two nodes v_i, v_j as the representation of the corresponding edge (i, j), which can be formulated as follows,

$$\mu_{ij} = \mathrm{MLP}\big([h_i \| h_j]\big), \tag{16}$$

where $[\cdot\|\cdot]$ is the concatenation operator. Based on $\mu = \{\mu_{ij} | i, j = 1, 2, \cdots, N\}$ and Eq.(15), we obtain the probability matrix M_μ whose elements indicate the existence of the corresponding edges. Afterwards, we can sample the explanation g based on the probabilities in the matrix M_μ as follows,

$$g = \big(X_S, A_S = M_\mu \odot A\big). \tag{17}$$

So far, we have derived the optimizable representation of g utilized in IDEA. All experiments are finished on a machine with 4 *NVIDIA GeForce RTX 3090 24GiB* GPUs.

B.4 STEP-BY-STEP BREAKDOWN OF HGTOKENIZER

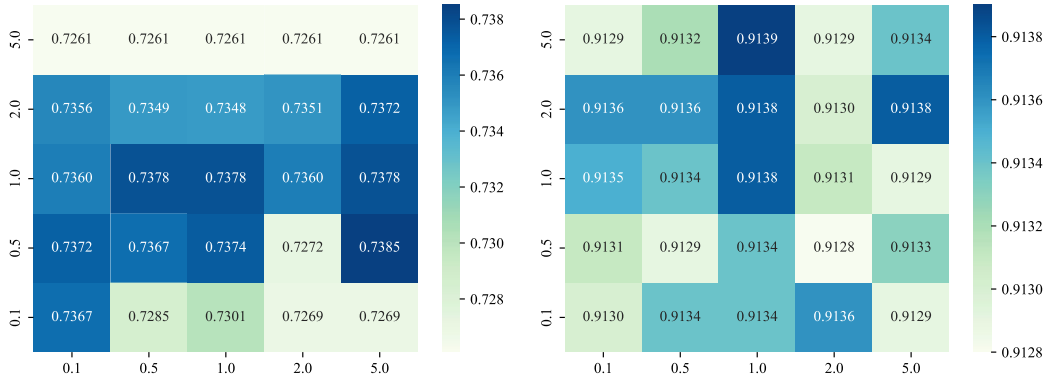
Given the input embedding $h_i \in \mathbb{R}^{1\times d}$, the shallow codebook $\mathcal{C}_S \in \mathbb{R}^{K\times d}$, and the deep codebook $\mathcal{C}_D \in \mathbb{R}^{K\times d}$, the details of HGTokenizer process are elaborated as follows.

- Step 1. Feed the input embedding h_i into the shallow quantizer. The shallow quantizer first calculates the pair-wise distance between h_i and each codeword $q \in \mathbb{R}^{1\times d}$ within the shallow codebook $\mathcal{C}_S$. Afterwards, the shallow quantizer select the closest codeword to h_i according to the K-dimensional distance vector. Step 1 is formulated by the formula $q^*_{S,i} = \mathrm{GQ}_S(h_i) = \arg\min_{q\in\mathcal{C}_S} \mathcal{D}(q, h_i)$ in Equation 2.
- Step 2. Calculate the quantization residual of the shallow quantizer, which is formulated by the formula $h'_i = h_i - q^*_{S,i}$ in Equation 2.
- Step 3. Feed the residual embedding h'_i into the deep quantizer, the detailed quantization process in the same the that in Step 1. The deep quantizer will select the closest codeword $q^*_{D,i} = \mathrm{GQ}_D(h'_i) = \arg\min_{q\in\mathcal{C}_D} \mathcal{D}(q, h'_i)$, as shown in Equation 3.

Subsequently, the quantized representations $q^*_{S,i}$ and $q^*_{D,i}$ provide by the shallow and deep quantizers are used to compute the disentanglement loss $\mathcal{L}_D$. The sum of them, i.e., $q^*_i = q^*_{S,i} + q^*_{D,i}$ in Equation 3, is used to compute the structure-awareness loss $\mathcal{L}_S$.

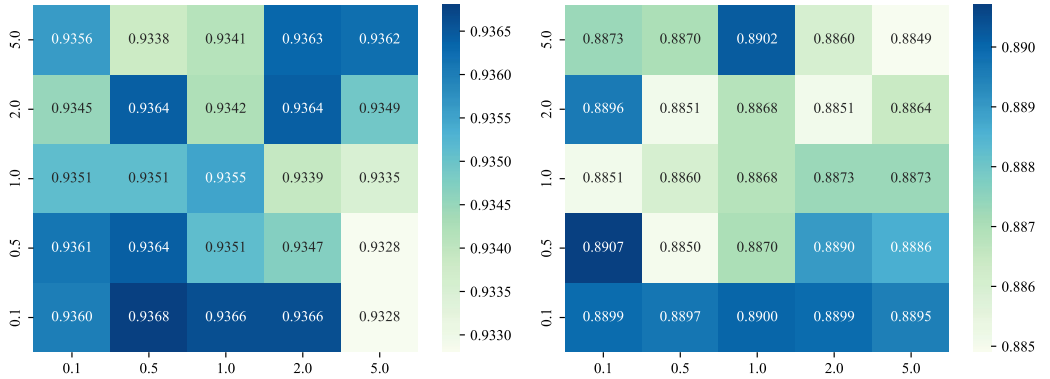
Table 6: The optimal configuration of hyper-parameters in IDEA.

Hyper-parameter	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs
ID Epochs	10	10	5	5	15
ID Learning Rate	0.01	0.01	0.005	0.0005	0.001
EA Epochs	20	10	15	10	10
EA Learning Rate	0.0001	0.0005	0.001	0.001	0.0001
Batch Size	20	64	64	32	20
Codebook Size	16	32	64	32	48



(a). Mutagenicity

(b). Benzene



(c). Alkane

(d). Fluoride

Figure 5: Explanation performance (ROC-AUC ↑) versus the weighted parameter λ_Q (y-axis) and λ_S (x-axis) in the SAD objective, on (a). Mutagenicity, (b). Benzene, (c).Alkane, and (d). Fluoride.

C Hyper-parameter Analysis

In Table 6, we summarize the optimal configuration of hyper-parameters in IDEA for each dataset.

C.1 Structure-aware Disentanglement Objective

Here, we investigate the impact of the weighted parameters λ_Q and λ_S in the strcuture-aware disentanglement objective defined as Eq.7,

$$\mathcal{L}_{\text{SAD}} = \mathcal{L}_D + \lambda_S \cdot \mathcal{L}_S + \lambda_Q \cdot \mathcal{L}_Q.$$

The evaluated result is presented in Figure 5. As the weighted hyper-parameters range from 0.1 to 5.0, we can notice that the optimal performance is more likely to be achieved when the objective weights are balanced, i.e., along the diagonal direction of the heatmap. For the Mutagenicity and Benzene datasets, the best performance is achieved by setting $\lambda_S = \lambda_Q = 1.0$. For the Alkane and Fluoride datasets, the best configurations of weighted parameters are $\lambda_S = 0.5, \lambda_Q = 0.1$ and $\lambda_S = 0.1, \lambda_Q = 0.5$, without severe unbalance.

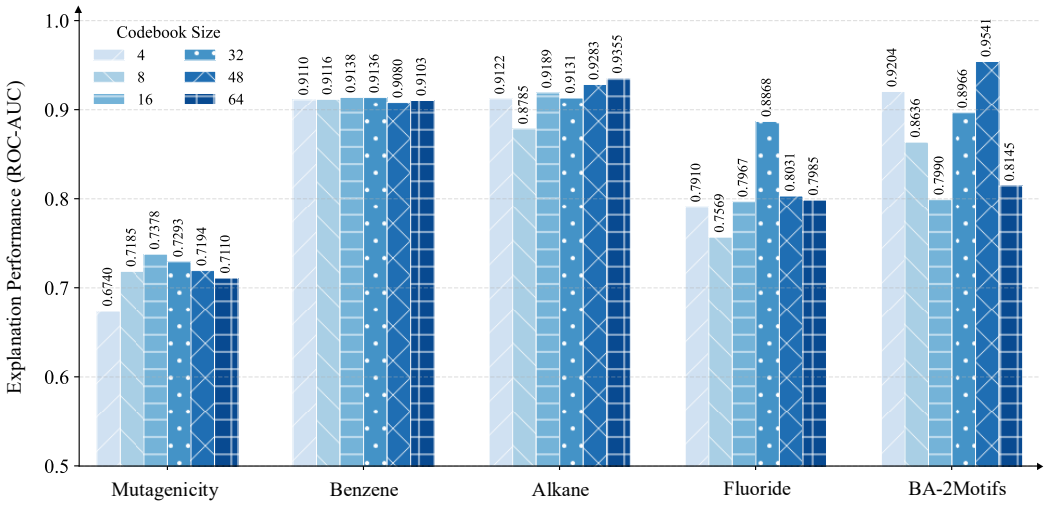


Figure 6: Explanation performance (ROC-AUC ↑) versus the codebook size in IDEA framework, on Mutagenicity, Benzene, Alkane, Fluoride, and BA-2Motifs datasets.

Table 7: Runtime (Second ↓) of four native explainers with different architectures and the corresponding IDEA variants. *Times* is defined as `IDEA/Native`.

Model	Mutagenicity	Benzene	Alkane	Fluoride	Average
PGExplainer	4.92	3.80	2.24	6.67	4.41
+IDEA	14.53	10.34	5.43	19.13	12.36
Times	2.95	2.72	2.42	2.87	2.80
ReFine	13.14	39.44	13.21	28.10	23.47
+IDEA	21.04	61.90	20.29	50.45	38.42
Times	1.60	1.57	1.54	1.80	1.64
V-InFoR	5.31	11.70	7.17	18.69	10.72
+IDEA	13.66	29.19	19.25	45.48	26.90
Times	2.57	2.49	2.68	2.43	2.51
ProxyExplainer	20.90	15.23	7.33	14.82	14.57
+IDEA	21.52	16.52	7.90	15.85	15.45
Times	1.03	1.08	1.08	1.07	1.06

C.2 Codebook Size

In this section, we investigate the impact of the codebook size, i.e., the number of the codewords within the graph quantizer. As shown in Figure 6, the codebook size ranges among $\{4, 8, 16, 32, 48, 64\}$. In general, a codebook with appropriate size can improve the explanation performance, since it serves as the foundation during structural information disentanglement and explanatory prototype modeling. For the Fluoride dataset, a codebook consisting of 8 codewords causes a performance degradation by 0.1299.

D Explanation Visualization

From Figure 7 to Figure 10, we present the explanation visualization of the ground truth, IDEA, and four SOTA GNN explainers based on the label preserving framework.

As shown in Figure 7, only IDAE assigns the highest importance score to NH_2. PGExplainer and ProxyExplainer assign medium scores to NH_2, V-InFoR identifies part of the NH_2 group, and GNNExplainer fails to detect NH_2. In Figure 8, IDEA, PGExplainer, and ProxyExplainer successfully identify the benzene ring. In particular, IDEA detects the two rings within the molecule, while

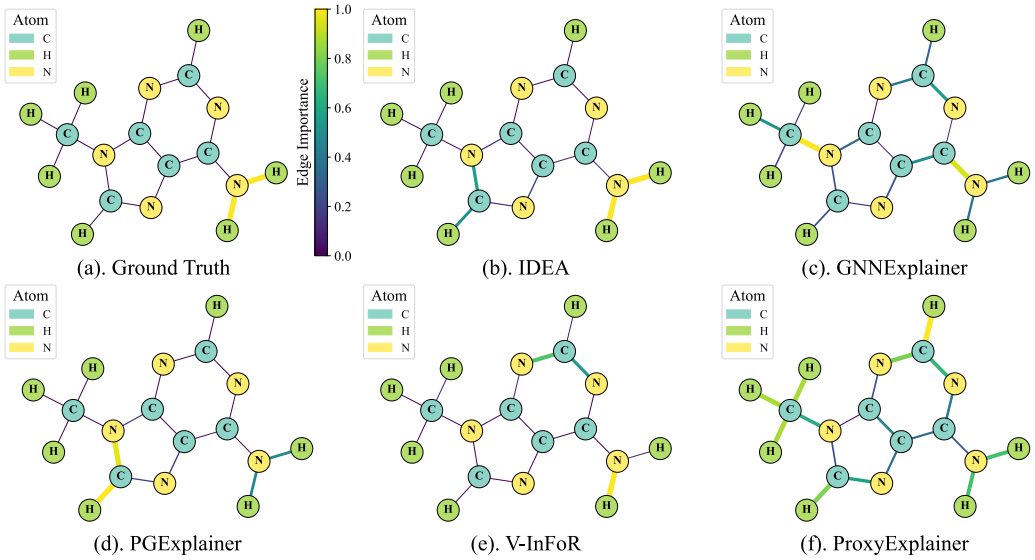


Figure 7: Explanation visualization of ground truth, IDAE, and four baselines on Mutagenicity.

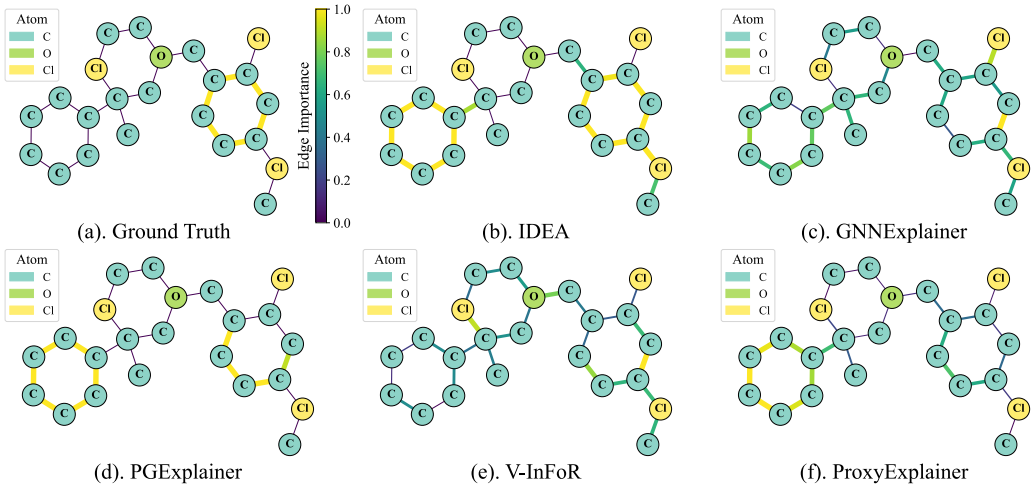


Figure 8: Explanation visualization of ground truth, IDAE, and four baselines on Benzene.

PGExplainer and ProxyExplainer notice only part of the second benzene ring. GNNExplainer and V-InFoR fail to assign high scores to the benzene rings. For the Alkane dataset, IDEA and GNNExplainer identify the chlorine atom Cl as the explanation, yet the other three explainers completely ignore the influential substrcutures. In Figure 10, only IDEA and V-InFoR can discriminate the explanatory structure from the confounding structures to some extent. The other three explainers assign nearly identical scores to all edges.

E Time Complexity

In this section, we first provide a theoretical analysis of the time complexity of the IDEA framework. Then, we report the runtime of diverse explainer architectures, incorporating with both the IDEA framework and the label preserving framework. Given the node presentation matrix $H_N \in \mathbb{R}^{N\times d}$, HGTokenizer approximates it by two graph quantizers, including two matrix multiplication operations and two `argmin` operations. The codebook within the graph quantizer belongs to a matrix in $\mathbb{R}^{K\times d}$. Hence, the time complexity of quantization distance $\mathcal{D}$ is

$$\vartheta_{\mathcal{D}} = NKd. \tag{18}$$

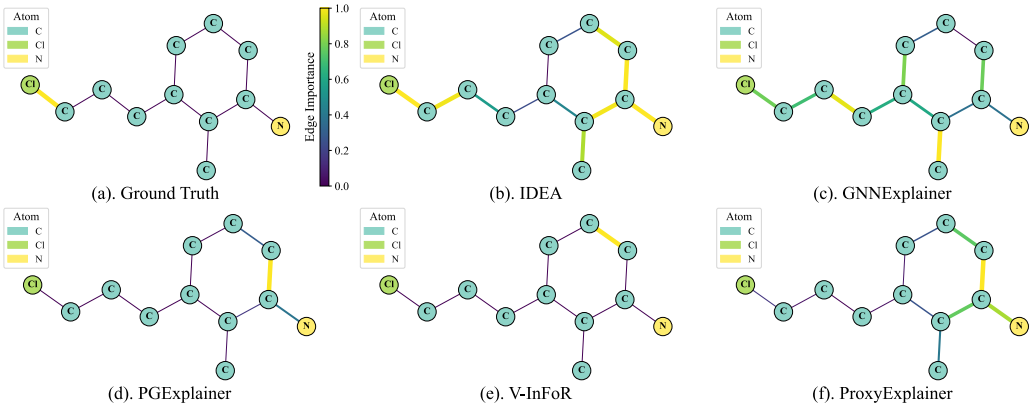


Figure 9: Explanation visualization of ground truth, IDAE, and four baselines on Alkane.

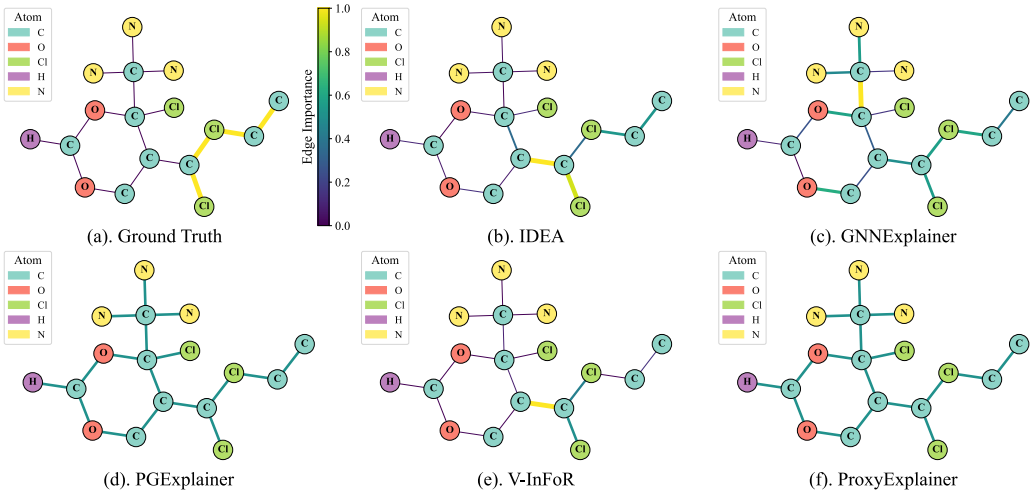


Figure 10: Explanation visualization of ground truth, IDAE, and four baselines on Fluoride.

Since the time complexity of `argmin` is NK, the total complexity of HGTokenizer is

$$\vartheta_{\text{HGT}} = 2NK(d+1). \tag{19}$$

Therefore, the complexity of IDEA, including the structural information disentanglement and the explanatory prototype alignment, is derived as,

$$\vartheta_{\text{IDEA}} = 2NK(d+1) + 3K(d+1) = O(NKd). \tag{20}$$

According to Eq.20, the time complexity is linear to the node number of input graph, the codebook size, and the hidden dimension of target GNN.

In Table 7, we report the runtime of four different GNN explainers and their counterparts shifted to the IDEA framework. One can note that the runtime of IDEA variants is of the same magnitude, compared with the native explainer adopting the label preserving framework.

F Supplementary Experiment

F.1 Weighted Combination

As a natural expansion, we integrate IDEA with the label preserving framework and the mixed optimization objective is defined as the convex combination as follows,

$$\mathcal{L}_{\text{Mix}} = \alpha \cdot \mathcal{L}_{\text{IDEA}} + (1-\alpha) \cdot \mathcal{L}_{\psi},\ 0 \leq \alpha \leq 1, \tag{21}$$

where $\mathcal{L}_{\psi}$ denotes the label preserving loss. Typically, $\mathcal{L}_{\psi}$ is defined as the mutual information between the predictions of the input graph and the explanation subgraph, i.e., $\text{MI}(\hat{y}, \hat{y}_g)$. The evaluate

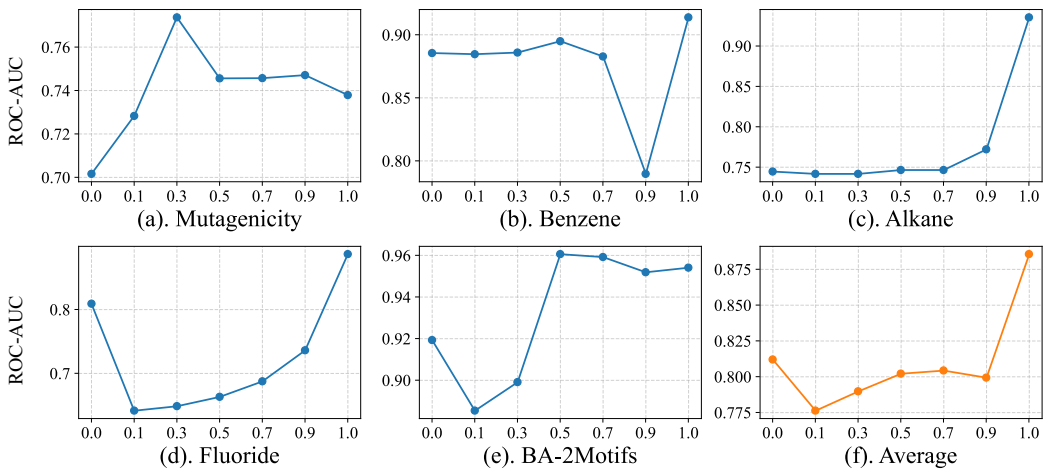


Figure 11: Explanation performance (ROC-AUC ↑) of the combination of the IDEA objective and the label preserving objective, on Mutagenicity, Benzene, Alkane, Fluoride, and BA-2Motifs.

Table 8: Explanation performance (ROC-AUC ↑) of IDEA and the conjoint variant.

Model	Mutagenicity	Benzene	Alkane	Fluoride	Average
IDEA	0.7379	0.9138	0.9319	0.8868	0.8676
IDEA-Joint	0.4805	0.5447	0.8725	0.8349	0.6832

results reveal that the effectiveness of the integration depends on the specific dataset. For the Mutagenicity dataset, the integration with α equals 0.3 achieves a significant improvement over both IDEA and the label preserving framework. However, for the Benzene, Alkane, and Fluoride datasets, the weighted integration is inferior to both IDEA and the label preserving framework, which might be caused by the counteract effect between the two optimization objectives.

F.2 Conjoint Optimization of IDEA

In our main experiment, IDEA is a dual-stage framework, where the Structural Information Disentanglement and the Explanatory Prototype Alignment are conducted separately. The dual stage implementation not only reduces the difficulty of IDEA optimization, but also avoids the counteraction effect between the optimization objectives. To empirically validate the rationality of dual-stage IDEA, we further implement an IDAE variant *IDEA-Joint* where the two stages are conducted jointly. The optimization objective of *IDEA-Joint* is defined as follows,

$$\mathcal{L}_{\text{Joint}} = \mathcal{L}_{\text{IDAE}} + \lambda_{\text{SAD}} \cdot \mathcal{L}_{\text{SAD}}, \tag{22}$$

with $\mathcal{L}_{\text{IDEA}}$ and $\mathcal{L}_{\text{SAD}}$ defined by Eq.11 and Eq.7, respectively. In Table 8, we present the performance comparison between IDEA and the conjoint variant, within the same hyper-parameter search range. We can notice the evident gap between *IDEA-Joint* and IDEA, which implies the difficulty of *IDEA-Joint* optimization, despite a possible performance upper bound better than IDEA.

F.3 Robustness to Label Noise

To investigate the robustness of the IDEA explainer to label noise (Zhong et al., 2023), we perturb the information disentanglement stage by flipping the GNN prediction $\hat{y}$ in Eq.5 and present the result in Table 9. For comparision, we evaluate the explanation performance of two typical explainers, i.e., GNNExplainer (Ying et al., 2019) and PGExplainer (Luo et al., 2020), with the same setting of label noise. Specifically, the intensity of the label noise ranges from 0.00% to 50.00%, with an interval of 5.00%. As shown by the result, the IDEA explainer stably maintains the high quality of the generated explanation, with the maximum performance degradation of 0.0076 and 0.0070 in the Mutagenicity and Benzene datasets, respectively. In contrast, two typical GNN explainers based on

Table 9: Explanation performance (ROC-AUC ↑) versus label noise intensity on Mutag and Benzene datasets. $\Delta_{\max}$ presents the maximum performance degradation with noise intensity increasing.

Noise Intensity	Mutagenicity			Benzene		
	GNNExplainer	PGExplainer	IDEA	GNNExplainer	PGExplainer	IDEA
0.00%	0.6155	0.7016	**0.7379**	0.6886	0.8855	**0.9138**
5.00%	0.6140	0.6989	**0.7358**	0.6662	0.8856	**0.9128**
10.00%	0.6063	0.6824	**0.7359**	0.6505	0.8856	**0.9135**
15.00%	0.5937	0.6819	**0.7363**	0.6326	0.8860	**0.9139**
20.00%	0.5954	0.6810	**0.7320**	0.6149	0.5784	**0.9132**
25.00%	0.5965	0.6805	**0.7366**	0.5966	0.5931	**0.9128**
30.00%	0.6050	0.6802	**0.7319**	0.5788	0.5932	**0.9128**
35.00%	0.6048	0.6801	**0.7303**	0.5636	0.6302	**0.9132**
40.00%	0.6065	0.6798	**0.7366**	0.5431	0.6591	**0.9131**
45.00%	0.6048	0.6795	**0.7327**	0.5282	0.7033	**0.9068**
50.00%	0.6058	0.6791	**0.7328**	0.5056	0.7340	**0.9130**
$\Delta_{\max}\downarrow$	0.0218	0.0225	**0.0076**	0.1830	0.3071	**0.0070**

the label preserving framework, GNNExplainer and PGExplainer, suffer from severe performance degradation, which is 14.03× times and 22.58× times greater than that of IDEA.

G Theoretical Justification

During the explanatory prototype alignment stage, we adopt the assignment probability of the input representation (H'_G or H_g) over the explanatory codebook $\mathcal{C}_D$ to reflect its location within the prototypical representation space. In this section, we elaborate the justification of this practice. Within the explanatory codebook $\mathcal{C}_D$, we have K prototype codewords $\{q_1, q_2, \cdots, q_K\} \subset \mathbb{R}^d$, which expand the prototypical representation space. Taking the prototype codewords $\{q_1, q_2, \cdots, q_K\}$ as the anchors, the L_2 distance between the input representation h and the anchor q_k is

$$\varphi_k = \|h - q_k\|^2 = h^T h + q_k^T q_k - 2q_k^T h. \tag{23}$$

For $k \geq 2$, by subtracting $\varphi_1 = \|h - q_1\|^2$, we can derive the following equation

$$\varphi_k - \varphi_1 = (q_k^T q_k - q_1^T q_1) - 2(q_k - q_1)^T h, \tag{24}$$

which is equivalent to the equation below,

$$(q_k - q_1)^T h = \frac{1}{2}\Big(q_k^T q_k - q_1^T q_1 + \varphi_1 - \varphi_k\Big). \tag{25}$$

For $k = 2, 3 \cdots, K$, stacking $(q_k - q_1)^T h$ induces the following equation in matrix formulation,

$$\begin{bmatrix} (q_2 - q_1)^T \\ (q_3 - q_1)^T \\ \vdots \\ (q_K - q_1)^T \end{bmatrix} h = \frac{1}{2} \begin{bmatrix} q_2^T q_2 - q_1^T q_1 + \varphi_1 - \varphi_2 \\ q_3^T q_3 - q_1^T q_1 + \varphi_1 - \varphi_3 \\ \vdots \\ q_K^T q_K - q_1^T q_1 + \varphi_1 - \varphi_K \end{bmatrix}, \tag{26}$$

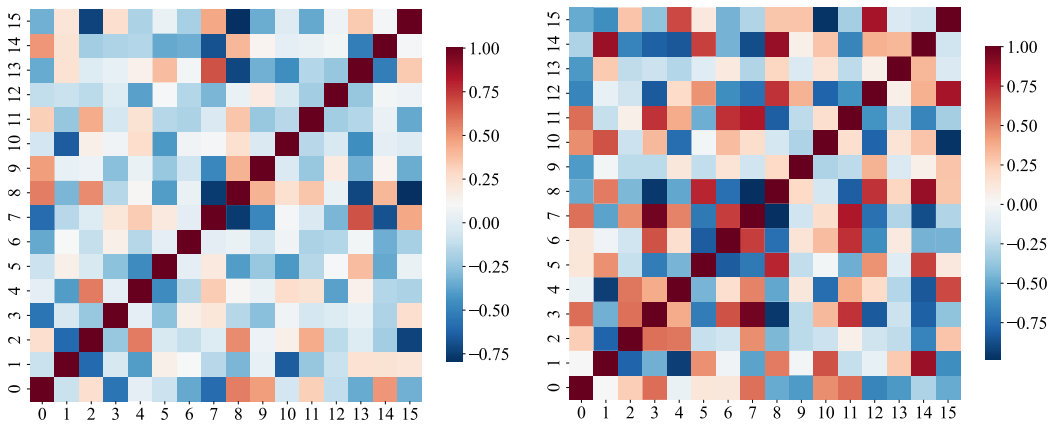
which can be briefly noted as

$$Ah = b,\ A \in \mathbb{R}^{(K-1)\times d},\ b \in \mathbb{R}^{(K-1)}. \tag{27}$$

Theoretically, the prototype codewords and the induced prototypical representation space are generated by a collection of latent variables $\{z_1, z_2, \cdots, z_t\} \subset \mathbb{R}^{d'}$ with $d' < d$, which objectively determine the explanatory substructures while being unobservable. Therefore, Eq.27 implies a counterpart in the latent space $\mathbb{R}^{d'}$ as follows,

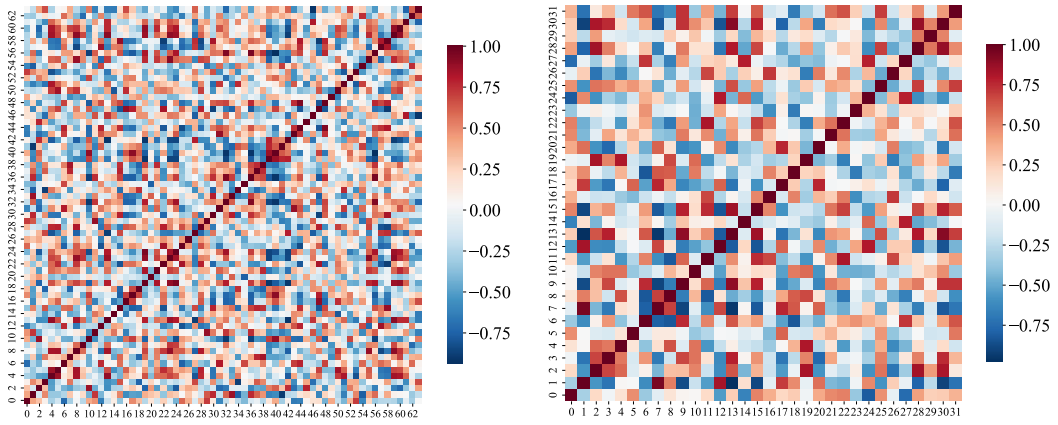
$$A'h' = b',\ A' \in \mathbb{R}^{(K-1)\times d'},\ b' \in \mathbb{R}^{(K-1)}. \tag{28}$$

In this equation, when $K \geq d' + 1$, h' has a unique solution. Therefore, we adopt the assignment probability based on the quantization distance to indicate the location of the input representation, instead of training an additional projector.



(a). Mutagenicity: -0.059±0.31

(b). Benzene: -0.062±0.51



(c). Alkane: 0.016±0.44

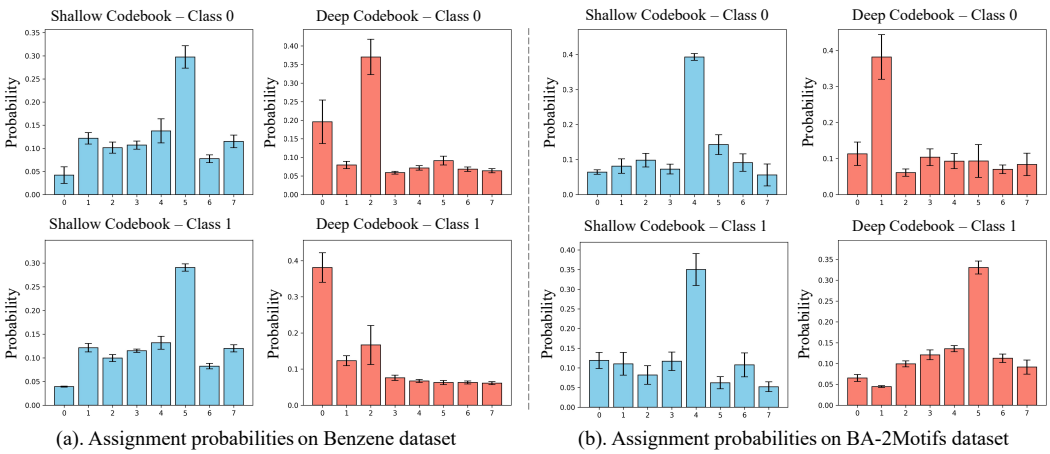
(d). Fluoride: -1.83e-05±0.44

Figure 12: Pair-wised cosine similarity of codewords. The number behind the dataset name represents `Mean` ± `Std`.

We denote the unknown mapping function from the latent space $\mathbb{R}^{d'}$ to the prototypical space $\mathbb{R}^d$ as $\mathcal{H} : \mathbb{R}^{d'} \to \mathbb{R}^d$. For two representations $x, y \in \mathbb{R}^d$, the corresponding representations in $\mathbb{R}^{d'}$ are denoted as $x' = \mathcal{H}^{-1}(x)$ and $y' = \mathcal{H}^{-1}(y)$. In our method, we minimize the difference between the assignment probabilities of x and y over the prototypes $\{q_1, q_2, \cdots, q_K\}$, in order to minimize the distance between x' and y' in the latent space. Theoretically, the strict validity of this measurement lies in three conditions. First, $K \geq d'$, which holds with large probability. Second, the prototype representations $\{q_1, q_2, \cdots, q_K\}$ are linearly independent. As illustrated in Figure 12, we present the cosine similarity of the codewords, demonstrating that the codewords approximately satisfy the linearly independent requirement. Third, the hypothetical mapping function $\mathcal{H}$ is linear or can be approximated by linear functions.

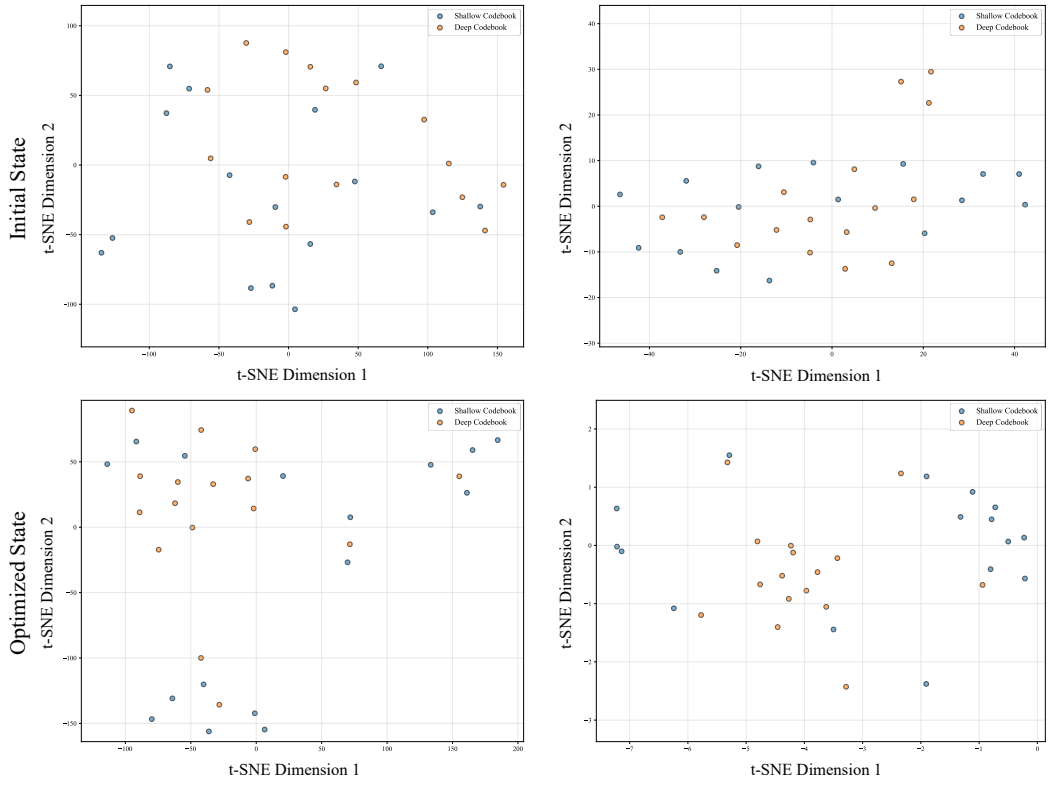
H Prototype Case Study

Assignment Probability. First, to explore the implicit relationship between the prototypical embeddings (i.e., codebooks) and human-intelligible substructures, we present the assignment probabilities distribution in Figure 13. Specifically, for real-world dataset Benzene and synthetic dataset BA-2Motifs, we visualize the average probabilistic distributions of class 0 and class 1 over the shallow and deep codebooks. For the real-world dataset Benzene, the distributions of class 0 and class 1 over the shallow codebook are similar, and the codeword 5 with the largest probability may correspond the most frequent non-explanatory substructure (carbon-chlorine bond). On the deep codebook, the distribution patterns obviously differ. For the deep codebook, the codeword 0 may correspond to



(a). Assignment probabilities on Benzene dataset (b). Assignment probabilities on BA-2Motifs dataset

Figure 13: Average probabilistic distributions over the shallow and deep codebooks on (a). Benzene dataset and (b). BA-2Motifs dataset.

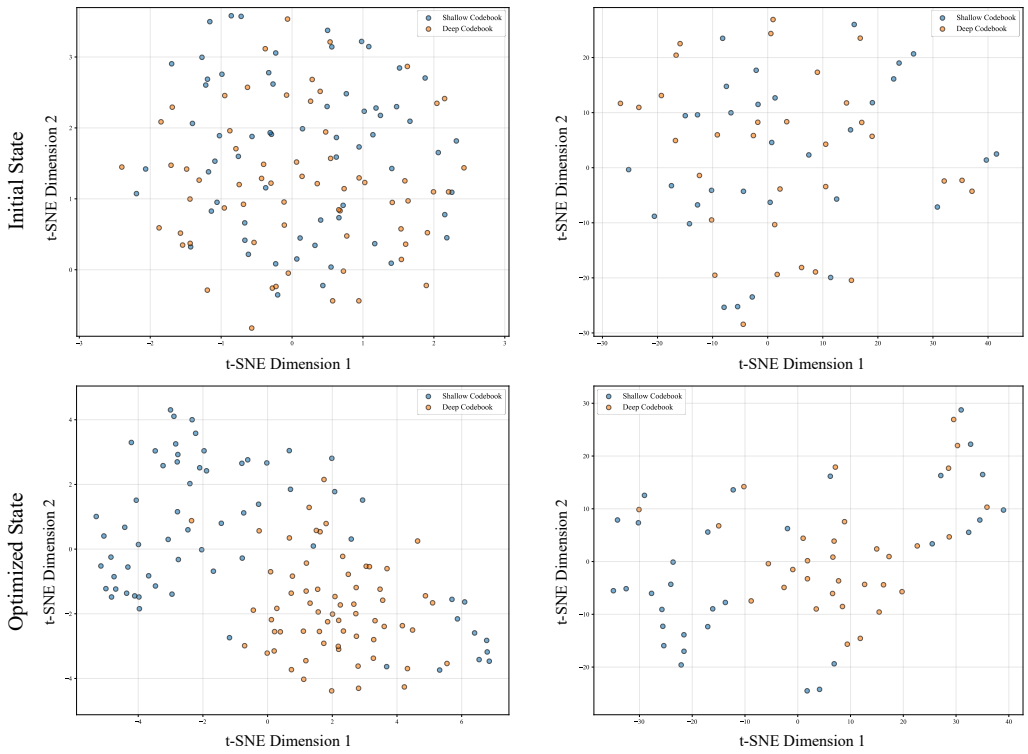


(a). Mutagenicity (b). Benzene

Figure 14: t-SNE visualization of codewords on (a). Mutagenicity dataset and (b) Benzene dataset.

the benzene rings which directly decides the labels of class 1, and the codeword 2 may correspond to the carbon-oxygen bond which is common in class 0. For the synthetic dataset BA-2Motifs, the shallow distribution patterns of class 0 and class 1 are also similar. The deep shallow distribution has two peaks, i.e., codeword 1 and codeword 5, which may correspond to the two kinds of motifs in BA-2Motifs. To sum up, the similar distribution pattern on shallow codebook and significantly different patterns on deep codebook can indicate that the learned prototypes in codebooks are implicitly related to substructures.

t-SNE Visualization. Furthermore, we visualize the learned codewords in shallow and deep codebooks based on t-SNE algorithm (van der Maaten & Hinton, 2008). As shown by Figure 14 and



(a). Alkane (b). Fluoride

Figure 15: t-SNE visualization of codewords on (a). Alkane dataset and (b) Fluoride dataset.

Table 10: Explanation performance (Fidelity$_+$ ↑) of IDEA and SOTA baselines across five datasets.

Fidelity$_+$	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs
GNNExplainer	0.2136$\pm$ 0.0005	0.5614$\pm$ 0.0005	0.5435$\pm$ 0.0130	0.1242$\pm$ 0.0026	0.4067$\pm$ 0.0033
PGExplainer	0.2012$\pm$ 0.0097	0.7250$\pm$ 0.0028	<u>0.7826</u>$\pm$ 0.0063	0.4097$\pm$ 0.0118	<u>0.4375</u>$\pm$ 0.0030
GraphMask	0.0982$\pm$ 0.0080	0.4450$\pm$ 0.0169	0.5659$\pm$ 0.0180	0.2070$\pm$ 0.0029	0.3750$\pm$ 0.0043
ReFine	<u>0.2161</u>$\pm$ 0.0041	0.5690$\pm$ 0.0048	0.6224$\pm$ 0.0033	0.6132$\pm$ 0.0077	0.2068$\pm$ 0.0024
V-InFoR	0.1954$\pm$ 0.0004	0.5265$\pm$ 0.0031	0.6883$\pm$ 0.0013	0.6298$\pm$ 0.0005	0.3793$\pm$ 0.0058
D4Explainer	0.0698$\pm$ 0.0181	0.5248$\pm$ 0.0080	0.6093$\pm$ 0.0058	0.6047$\pm$ 0.0026	0.2127$\pm$ 0.0014
MixupExplainer	0.1277$\pm$ 0.0074	0.4910$\pm$ 0.0047	0.4579$\pm$ 0.0053	0.3672$\pm$ 0.0009	0.2131$\pm$ 0.0082
ProxyExplainer	0.1841$\pm$ 0.0132	<u>0.7473</u>$\pm$ 0.0118	0.6904$\pm$ 0.0052	<u>0.6607</u>$\pm$ 0.0351	0.3064$\pm$ 0.0027
IDEA	**0.2207**$\pm$ 0.0093	**0.8292**$^*\pm$ 0.0081	**0.8043**$^*\pm$ 0.0160	**0.6988**$^*\pm$ 0.0042	**0.4450**$^*\pm$ 0.0004
Improvement	2.13%	10.96%	2.77%	5.77%	1.71%

Figure 15, the first row presents the t-SNE visualization of the initial codewords, and the second row presents that of the codewords after optimization, i.e., prototypes. We can notice that in the initial state, the shallow and deep codewords mix together without clear boundary. After optimization, the deep codewords are approximately separable from the shallow ones. The deep codewords prefer to cluster into a mass, while the shallow codewords still distribute dispersedly.

I Faithfulness Evaluation

To comprehensively evaluate the effectiveness of IDEA, we present faithfulness metrics based on fidelity in this section (Amara et al., 2022). Specifically, Fidelity$_+$ measures the change degree of the GNN prediction after removing the explanation subgraph, Fidelity$_-$ measures the change degree

Table 11: Explanation performance (1-Fidelity$_-$ ↑) of IDEA and SOTA baselines.

1-Fidelity$_-$	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs
GNNExplainer	0.5975± 0.0053	0.4370± 0.0051	0.1658± 0.0070	0.2679± 0.0114	0.7366± 0.0031
PGExplainer	0.7714± 0.0165	0.5222± 0.0037	0.3787± 0.0048	0.2232± 0.0167	0.9055± 0.0033
GraphMask	0.6174± 0.0032	0.4365± 0.0065	0.2683± 0.0058	0.1487± 0.0013	0.5005± 0.0051
ReFine	0.6604± 0.0044	0.5056± 0.0082	0.3237± 0.0022	0.2863± 0.0104	0.8330± 0.0140
V-InFoR	0.6375± 0.0020	0.4524± 0.0039	0.3886± 0.0070	0.2871± 0.0004	0.7872± 0.0093
D4Explainer	0.6451± 0.0240	0.4497± 0.0019	0.3691± 0.0086	0.2577± 0.0151	0.9710± 0.0038
MixupExplainer	0.6745± 0.0115	0.4962± 0.0063	0.3750± 0.0014	0.2665± 0.0051	0.9513± 0.0077
ProxyExplainer	0.7912± 0.0058	0.6483± 0.0156	**0.4191**± 0.0028	0.3594± 0.0177	0.9697± 0.0062
IDEA	**0.8018***± 0.0086	**0.6964***± 0.0148	0.4190± 0.0158	**0.3612***± 0.0010	**0.9981***± 0.0003
Improvement	1.34%	7.42%	-0.02%	5.00%	2.93%

Table 12: Explanation performance (Harmonic mean ↑) of IDEA and SOTA baselines.

Harmonic Mean	Mutagenicity	Benzene	Alkane	Fluoride	BA-2Motifs
GNNExplainer	0.3146± 0.0013	0.4914± 0.0031	0.2541± 0.0096	0.1696± 0.0019	0.5240± 0.0035
PGExplainer	0.3191± 0.0136	0.6071± 0.0035	0.5104± 0.0056	0.2888± 0.0169	0.5899± 0.0034
GraphMask	0.1693± 0.0119	0.4404± 0.0069	0.3640± 0.0089	0.1731± 0.0019	0.4287± 0.0031
ReFine	0.3256± 0.0051	0.5354± 0.0066	0.4259± 0.0027	0.3903± 0.0110	0.3313± 0.0036
V-InFoR	0.2991± 0.0006	0.4866± 0.0035	0.4967± 0.0060	0.3944± 0.0005	0.5119± 0.0063
D4Explainer	0.1254± 0.0301	0.4843± 0.0042	0.4597± 0.0080	0.3612± 0.0152	0.3490± 0.0021
MixupExplainer	0.2146± 0.0106	0.4935± 0.0034	0.4123± 0.0030	0.3088± 0.0037	0.3481± 0.0106
ProxyExplainer	0.2985± 0.0178	0.6943± 0.0140	0.5216± 0.0036	0.4655± 0.0232	0.4657± 0.0036
IDEA	**0.3460***± 0.0114	**0.7569***± 0.0119	**0.5509***± 0.0171	**0.4762**± 0.0018	**0.6156***± 0.0004
Improvement	6.26%	9.02%	5.62%	2.30%	4.36%

of the GNN prediction when only retain the explanation subgraph, formally defined as follows,

$$\text{Fidelity}_+ = 1 - \frac{1}{|\mathcal{G}_{\text{test}}|}\sum_i \mathbb{I}\Big(f(G_i \backslash g_i) = f(G_i)\Big), \tag{29}$$

$$\text{Fidelity}_- = 1 - \frac{1}{|\mathcal{G}_{\text{test}}|}\sum_i \mathbb{I}\Big(f(g_i) = f(G_i)\Big), \tag{30}$$

where $\mathcal{G}_{\text{test}}$ is the test set, f is the target GNN model, G_i is the i-th test graph sample, and g_i is the corresponding explanation subgraph identified by GNN explainer.

For readability, we report Fidelity$_+$, 1-Fidelity$_-$, and their harmonic mean in Tables 10, 11, and 12 respectively, which are better when they are higher and belong to $[0, 1]$. The results also demonstrate the superiority of IDEA when compared with the SOTA baselines.

J Limitation

Accessibility to target GNN. According to the taxonomy of GNN explanation methods, the accessibility of the GNN explainer to the target GNN to be explained can be categorized into black-box, gray-box, and white-box. The black-box accessibility takes the GNN model as an oracle and only requires the GNN predictions. On the contrary, the white-box accessibility demands the permission to the model internal parameters or the model gradients (Pope et al., 2019). Actually, IDEA requires the gray-box accessibility to utilize the GNN encoded representations, which limits the application of IDEA to completely black-box GNN models.

Approximately linear assumption on unknown mapping function $\mathcal{H}$. In Appendix G, we introduce a unobserved function $\mathcal{H}$ that maps the latent space $\mathbb{R}^{d'}$ to our prototypical space $\mathbb{R}^d$. The strict

validity of $\mathcal{L}_{\text{IDEA}}$ in the explanatory prototype alignment stage necessitates that $\mathcal{H}$ is approximately linear at least. Hence, a highly non-linear function $\mathcal{H}$ might become a potential limitation of IDEA.

K Future Work

A promising direction for future work is to extend the proposed quantization-based explanation framework from instance-level to model-level interpretability. One possibility is to construct a global dictionary of reference quantization prototypes that summarizes the model's decision behavior across the entire dataset. By analyzing how deep quantization patterns cluster in latent space, such a dictionary could reveal class-level structural regularities or decision boundaries, analogous to prototype-based global explanations in prior work. Furthermore, integrating hierarchical or dynamic prototype discovery may help capture more nuanced variations in quantized representations, enabling a more comprehensive characterization of the model's reasoning process.

L Use of Large Language Models

Large language models (LLMs) are used in this work solely for auxiliary purposes. Specifically, they assisted in improving the accuracy of writing by identifying and correcting grammatical issues and refining terminology choice, as well as in suggesting appropriate color schemes for figure design. All research ideas, methodological developments, experiments, and the main body of the manuscript are independently conceived, conducted, and written by the authors.