ITXGNN: LEARNING STABLE AND CLASS-CONSISTENT EXPLANATIONS FOR GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-Interpretable Graph Neural Networks have become quite effective at being able to retain the prediction accuracy of comparable black box methods while also introducing interpretability to their decision making processes. However, most self-interpretable methods do not generate stable explanations. In this work, we define a key notion of stability that is absent in the literature and provide a framework that addresses this problem head on. Our framework **Information-Theoretic Self-Explainable Graph Neural Network** (ITXGNN) is a method that utilizes subgraph extraction with prototypes, clustering, and representation learning in order to satisfy an objective function that prioritizes stability while maintaining accuracy. In addition to our newfound definition of stability we provide theoretical justification of our objective function and derivations highlighting the natural choice of objective function as well as a mathematical bridge from theory to implementation. We conduct comprehensive experiments on several datasets showcasing our method's strength on stability in comparison to leading existing self-interpretable GNNs while maintaining competitive prediction performance. We also conduct ablation studies that break down our frameworks key components and provides quantitative analysis for each components contribution and we also provide a qualitative case study showcasing explanation quality.

1 INTRODUCTION

Graph structured data is ubiquitous and prevalent in numerous domains, which has led to the advent of Graph Neural Networks (GNNs) to address handling of such data (Henaff et al., 2015; Kipf & Welling, 2016; Veličković et al., 2017; Hamilton et al., 2017; Xu et al., 2018). GNNs have enjoyed tremendous success in the fields which they are deployed on in predictive performance. Standard black box GNNs can successfully handle prediction tasks such as outlier/fraud detection, graph classification, node classification, and ranking/recommendation tasks to name a few. However, a lot of the great successes in the field have been enjoyed by black box methods that fail to provide any explanation to the underlying logic or reason for why they make the decisions they do. This is particularly problematic for domains that are critical or high stakes where decisions need to be justified. This has led to the recent surge in interpretable or explainable machine learning which tries to give an explanation in addition to the output of a model in order to justify the model's learned behavior (Ribeiro et al., 2016; Selvaraju et al., 2016; Sundararajan & Najmi, 2020; Ji et al., 2025). It is evident that there are real legal, societal, and ethical implications on key decisions that mandate the quality of explanations to enhance the trustworthiness of systems being deployed.

Interpretability or explainable machine learning focuses on providing explanations to explain the decision making process of a machine learning model. Many graph interpretable models have emerged just recently to various degrees of success (Huang et al., 2022; Schlichtkrull et al., 2020; Ying et al., 2019). Currently, the focus on self-explainable models has been on the quality of the explanations. Specifically, do the extracted

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

explanations look plausible and correlate highly to a label. The current quantitative metrics mainly quantify the accuracy of an explanation, how sparse/compact an explanation be, or how faithful an explanation can be (Pope et al., 2019). However, we argue that there is an orthogonal angle that has been underexplored when quantifying the quality of an explanation. Currently, there is limited work done on quantifying or even considering the **stability** of the explanations. We argue that for an explanation to be considered truly trustworthy an explanation should be stable. A stable explanation should have several key desiderata, stated plainly; similar explanations should be given for similar graphs, and conversely differing explanations for differing graphs. We argue that in the context where there is a identifiable underlying phenomenon in a graph such as for example an active ingredient in a molecule, the quality of an explanation should not just rest upon identifying a superset of the active ingredient but consistently being able to explain the active ingredient when it is the primary reason the molecule has such properties. Numerous existing methods will claim high quality explanations since their method occasionally perfectly isolates the ground-truth explanation but we argue this is not enough and to truly be considered a trustworthy system the model should be able to consistently give stable explanations.

This has led to our work; we seek to address the gap in the literature and address stability by introducing our framework with stability and accuracy of the predictions as paramount goals in our framework. We introduce our method Information-Theoretic Self-Explainable Graph Neural Network (ITXGNN) which uses an objective function that prioritizes explanations that are consistent for clusters of similar graphs while differentiating between differing clusters of graphs. Each cluster of graphs utilizes a prototype to anchor explanatory subgraphs and we use this to design a subgraph extractor and guide the downstream representation learning.

1.1 Our contributions

The core of our contributions is in identifying the gap in the literature of stability in explanations that we argue is much needed. The rest of our contributions are given below:

(i) We propose and formalize a notion of stability that is currently absent from the literature. We argue its necessity in improving the trustworthiness of graph based machine learning systems.
(ii) We then introduce our framework which is built upon a optimization objective that at its core attempts to optimize for explanatory subgraphs that are simultaneously stable and correlate to their respective label. We provide theoretical analysis that shows the derivation of our high-level objective function to its practical computation ready implementation. We also provide theoretical analysis of our objective function's behavior in idealized scenario highlighting its stability.
(iii) We conduct comprehensive experiments to quantify and gauge the stability and predictive performance of our framework and relevant baselines. We also conduct ablation and hyperparameter analysis to further analyze the performance of our method and justify design choices.

2 Related Work

Interpretable machine learning is a relatively new field in machine learning, and has long been considered as an important component in establishing trust in machine learning systems. Initially, the work in interpretable or explainable machine learning was mainly focused on post-hoc explanations (Ying et al., 2019; Vu & Thai, 2020; Luo et al., 2020; Schlichtkrull et al., 2020; Yuan et al., 2021; Lucic et al., 2022). Post-hoc explainers take a black-box model and attempt to explain the model's decision making process after the black box model has already been trained. The issue with these methods is that they explain the behavior of a black-box model whose training process has already concluded and despite a model with excellent predictive performance its decision boundaries may not be reflective of the ground truth.

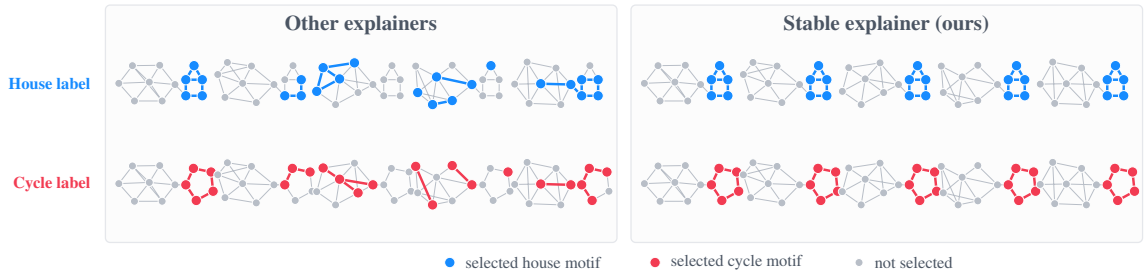

Figure 1: Schematic illustration of explanation stability on BA-2Motif. Other explainers may identify the class defining motif for an individual graph while producing inconsistent explanations across graphs of the same class. In contrast, the stable explainer consistently selects the house motif (blue) or cycle motif (red) despite variation in the background graph. Gray nodes and edges are not selected.

Currently, much of interpretable machine learning has shifted away from post hoc explainers and more towards self-interpretable methods (Wu et al., 2022). These methods incorporate interpretability into the model so that practitioners can be made aware of why the model made the choices it did in addition to the actual prediction. In the area of practice regarding graph machine learning, self interpretable models distinguish themselves because they provide an explanation in the form of a subgraph in addition to a prediction.

Most self-interpretable methods in graph ML have utilized an information-theoretic objective function (Wu et al., 2020; Yu et al., 2022; 2020; Miao et al., 2022). Generally, these methods have a subgraph extractor to give an explanation which maximizes the mutual information between the resulting explanation and the label (*relevance*) while minimizing the mutual information between the explanation and the original graph (*compression*). The issue with these methods is that there are numerous explanations that satisfy the objective without being tied to the ground truth. There is also no term to ensure that the explanations are consistent across the data, see figure 1.

Recently, some work has introduced the idea of using prototypes to help guide the subgraph extraction to be more consistent across the data (Zhang et al., 2022; Seo et al., 2024). These methods optimize an objective that guides representation learning and resulting subgraphs to be close to their respective prototypes. This can help stabilize explanations but if the objective function does not provide any pressure to force similar explanations to be similar for similar graphs the resulting explanations just need to be relatively close to the prototypes in an embedding space in order to satisfy the objective. There is nothing to actually force the explanations to be similar to prototypes or to force the explanations to be stable.

There is very limited work on stability in explainable graph machine learning. Some initial work in self-interpretable machine learning did define a notion of stability (Alvarez Melis & Jaakkola, 2018). However, this notion focused highly on robustness of explanations in the face of adversarial perturbations and also focus more on the if the model behaves coherently locally around each input. There has also been work on the consistency of graph explanation methods (Tai et al., 2026). This is a related but distinct field where the focus is more on the training process of an interpretable method; across different seeds does the model generate similar or the same explanation consistently? Additionally, a recent theoretical work sought to formalize the notion of stability and focused on formalization of stability. This work specifically focused on perturbation-based, localized stability around an instance.

Prior work has primarily considered stability as the robustness of an explanation to small perturbations of an individual input Bajaj et al. (2021); Funke et al. (2022); Agarwal et al. (2022); Fang et al. (2023); Li et al. (2024; 2025); Tai et al. (2025). We argue that for models to be truly trustworthy we should expect to see the model identify a coherent, consistent, and stable explanation for the same underlying predictive

phenomenon while producing distinct explanations for instances governed by different phenomena. To this end, we propose a framework in which across instances explanation stability is treated as a central learning objective rather than merely an evaluation criterion.

3 STABILITY

3.1 PRELIMINARIES

In this section we formally introduce the notation that will be used throughout the document. We denote a graph as $G = (X, E)$ where a graph is defined by its node features X and edge list E. Denote a subgraph or explanation as G_S which is defined by its corresponding node features X_S and edge set E_S. let the graph exist in a graph space $G \in \mathcal{G}$ and let the explanation belong to the explanation space: $G_S \in \mathcal{S}$. Finally, we denote embeddings as $\mathbf{z}$, which is a d-dimensional vector: $\mathbf{z} \in \mathbb{R}^d$.

3.2 DEFINITION OF STABILITY

Now we formally define the definition of stability. As discussed several works have attempted to define stability in the local neighbor of individual instances. This is a good first step but we argue it is not enough. We want to see coherent, stable explanations for similar input graphs however, we also need to see a distinction for graphs that clearly belong to a distinct group. This is important because if we simply define stability as giving similar explanations then a trivial constant explainer would be considered the most stable explanation method yet its explanations would not be meaningful.

Definition 1 (Stability)**.** *Let G, G', and G'' denote graphs; let $\Phi : \mathcal{G} \to \mathcal{S}$ denote the mapping from graphs into the explanation space. Let $d_{\mathcal{E}}$ represent some distance measure. Let $\sim$ be the governing equivalence (assignment to the same prototype/label or cluster). We define the intra- and inter-group dispersions*

$$\delta = \mathbb{E}[d_{\mathcal{E}}(\Phi(G), \Phi(G'))] \qquad \text{if } G \sim G' \text{ (continuity)}, \tag{1}$$

$$\Delta = \mathbb{E}[d_{\mathcal{E}}(\Phi(G), \Phi(G''))] \qquad \text{if } G \nsim G'' \text{ (discrimination)}. \tag{2}$$

The explanation mapping Φ is stable if and only if $\delta < \Delta$; equivalently, when the ratio $\delta/\Delta < 1$.

4 METHODOLOGY

In this section, we introduce our method ITXGNN. We define the objective information-theoretic function and discuss each component of the framework. We will also discuss how the model improves stability.

4.1 FRAMEWORK OVERVIEW

ITXGNN is made up of several key components working in unison. ITXGNN learns a subgraph extractor $\Phi : \mathcal{G} \to \mathcal{S}$ to generate explanations. After obtaining an explanation the model learns a mapping $\phi : \mathcal{S} \to \mathbb{R}^d$, practically this mapping is a GNN that learns an embedding space for the explanations. ITXGNN then learns a suitable clustering and set of prototypical embeddings for each given clustering to maximize the objective function on the explanation embeddings. The distances computed in between the explanation embeddings and the prototypical embeddings are then used to define the probabilities that are used to compute the tractable objective.

4.2 THE FRAMEWORK

Subgraph Extraction The first step of our framework in order to be self-interpretable is to give an explanation in addition to a prediction. The explanation comes in the form of a subgraph, so we utilize a

learnable subgraph extractor to retain a subset of nodes/edges. In our framework this is done by using a Gumbel-Softmax stochastic attention mechanism similar to methods such as GSAT, PGIB (Jang et al., 2016; Maddison et al., 2016). Similar to these methods we try to learn a subgraph extractor denoted Φ that learns a parameter that defines a Bernoulli distribution. In contrast to existing methods, we use a node scorer that first learns an attention score for each node $u \in V$ given its features $\mathbf{h}_u \in X$. The key idea is when τ is relatively large the Gumbel-Softmax operator allows the attention mechanism to explore what nodes matter for our task and once τ drops it approaches binary decisions. The "keep" probability or hard sample derived from $\mathbf{m}_i$ is then used to construct the discovered subgraph. Given node representation $\mathbf{h}_u$ and node mask m_u the masked node feature is given as $\widetilde{\mathbf{h}}_u = m_u \mathbf{h}_u$ and edges are masked using the two nodes connecting the edge's mask: $w_{u,v} = m_u m_v$. We now have an explanation derived in the form of a subgraph: $G_S = (X_S, E_S)$. The technical details are provided in Appendix C.1.1.

Explanation Representation Now given a masked graph we can extract an explanation and then utilize a downstream GNN to obtain meaningful representations. Let $\phi : \mathcal{S} \to \mathbb{R}^d$ denote the GNN that learns an embedding space for our subgraphs. We utilize a two-layer edge aware subgraph encoder that uses a custom GIN layer (that can take mask derived edge weights). The final subgraph embedding is obtained via global add pooling.

Prototype Layer & Clustering Given a set of subgraph embeddings $\mathbf{z}_i \in \mathbb{R}^d$ for $i \in 1, ..., N$ we can now learn a suitable clustering and prototypical embeddings for each cluster. In our methodology, prototypes are always tied a clusters as a bijection. The prototype for cluster k is updated using the mean of subgraph embeddings weighted by the cluster assignment probabilities:

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{B} q_{ik} \mathbf{z}_i}{\sum_{i=1}^{B} q_{ik}},$$

where $\boldsymbol{\mu}_k \in \mathbb{R}^d$ is the prototypical embedding for cluster k, and $q_{ik} \in [0, 1]$ is the learnable clustering weight for graph G_i corresponding to cluster k. Finally, we define a prototype posterior which is the conditional probability that graph i belongs to the prototype $\boldsymbol{\mu}_k$ given by:

$$r_{ik} = \mathbb{P}(p = k \mid G_i) = \frac{e^{(s_{ik})}}{\sum_{j=1}^{K} e^{(s_{ij})}}.$$

Hence $r_i = (r_{i1}, \ldots, r_{iK})$ is a probability distribution over prototypes. The similarity score s_{ik} uses the dataset specific distance listed in Appendix D. In the squared Euclidean setting analyzed in Theorem 1, $s_{ik} = -\|\mathbf{z}_i - \boldsymbol{\mu}_k\|_2^2$.

4.3 Objective Function

Let $\hat{C}$ denote a clustering and let P denote a set of prototypes the goal of ITXGNN is to maximize the following objective function:

$$\max_{\Phi, \phi, \hat{C}, P} I(P; \hat{C}) + I(P; Y) \tag{3}$$

The first term is the mutual information (MI) between the set of prototypes and clustering; a cluster should be highly informative of a prototype and vice versa. The second term is maximized when a set of prototypes is highly informative of the corresponding labels and vice versa.

In order to actually computationally use and optimize eq (3) we will break it down into its implementation ready version. Let us consider the first term; the MI between prototypes and clustering. The detailed

derivation can be found in Appendix C.2. We have the joint distribution which is defined by the product $r_{ik}q_{il}$:

$$P(p = k, c = \ell) = \frac{1}{N}\sum_{i=1}^{N} r_{ik}q_{i\ell}.$$

Therefore the first term gives us:

$$I(P;\hat{C}) = \sum_{k=1}^{K}\sum_{\ell=1}^{K} P(p = k, c = \ell)\log\frac{\mathbb{P}(p = k, c = \ell)}{\mathbb{P}(p = k)\mathbb{P}(c = \ell)}.$$

The second term in eq (3) is obtained following the standard variational treatment of label information (Poole et al., 2019), we use cross-entropy on predictions from the prototype scores as a tractable objective for prototype-label alignment:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N}\log\hat{p}(y_i \mid G_i).$$

So, our final implementation ready objective function simplifies to:

$$\min_{\Phi,\phi,C,\hat{P}} -\left(\sum_{k=1}^{K}\sum_{\ell=1}^{K} p(p = k, c = \ell)\log\frac{\mathbb{P}(p = k, c = \ell)}{\mathbb{P}(p = k)\mathbb{P}(c = \ell)}\right) - \sum_{i=1}^{N}\log\hat{p}(y_i \mid G_i).$$

5 Analysis

With the definition of stability and a detailed look at our framework, we can now provide analysis in an ideal setting where we assume that the governing equivalence for our framework will be prototypes

Assumption 1 (One-dimensional common-noise model)**.** Let μ_1, μ_2 be the two prototypes, $\gamma = \|\mu_1 - \mu_2\|$ their separation, and $\hat{\mu} = (\mu_1 - \mu_2)/\gamma$. Let $t = \langle z, \hat{\mu}\rangle$ be the projection of the embedding onto the prototype axis. Conditioned on the cluster assignment $\hat{C}$, the projected coordinate satisfies

$$t \mid \hat{C} = k \;=\; \tau_k + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, s^2), \quad \varepsilon \perp \hat{C}, \tag{4}$$

i.e. the noise is homoscedastic, independent of the cluster, with variance exactly s^2, and $|\tau_1 - \tau_2| = \gamma$.

Theorem 1 (Prototype-SNR stability certificate)**.** *Suppose the prototype posterior is the distance-softmax $r_{i1} = \sigma\big(2\gamma(t_i - m)\big)$, with m the midpoint of the projected prototypes, and let $I(\hat{C};P)$ denote the clustering–prototype mutual information. Then, at any configuration and any training epoch, under Assumption 1,*

$$I(\hat{C};P) \;\le\; \tfrac{1}{2}\log\left(1 + \frac{\gamma^2}{4s^2}\right), \tag{5}$$

or equivalently, the embedding separation to scatter ratio obeys the floor under Assumption 1

$$\frac{\gamma}{s} \;\ge\; 2\sqrt{e^{2I(\hat{C};P)} - 1} \;\;\ge\; 2\sqrt{2\,I(\hat{C};P)}. \tag{6}$$

The proof is given in Appendix A. This theorem assumes a simple idealized setting and under the stated two prototype common noise assumptions, the achieved clutsering prototype MI gives a lower bound on seperation relative to spread in the learned embedding space. This does not directly bound distances between extracted subgraphs in Definition 1; the relation between the embedding statistic and the measured explanation is empirically evaluated in Section 6.

6 EXPERIMENTS

6.1 SETUP

We conduct comprehensive experiments on several datasets. We use the BA-2Motifs, MUTAG, SP-Motif, and GraphXAI datasets (Alkane-Carbonyl, Benzene, and Fluoride Carbonyl) for our experiments (Debnath et al., 1991; Luo et al., 2020; Wu et al., 2022; Agarwal et al., 2022). We include relevant baseline methods such as ProtGNN, PGIB, GSAT, GNNExplainer, PGExplainer, and SubgraphX in addition to our method. Predictive performance experiments are conducted on all datasets with label accuracy and AUC as primary reporting metrics. On datasets with fixed planted ground-truth motifs (SP-Motif and BA-2Motif) we report isomorphism rate (isorate) and normalized graph edit distance (nGED) for each label/motif. For chemical data since there is no fixed ground-truth motif, in order to gauge similarity/discrimination between explanations of similar/differing labels we utilized both intra and inter Frechet Inception Distance (FID). We utilized several choice of embeddings in these experiments as to not bias our result by the choice of embedding space. Following Definition 1, the intra-group dispersion δ splits a group's explanations in half and takes the FID between the halves, averaged over $\sim$100 resamples; the inter-group dispersion Δ is the FID between distinct groups; the stability ratio is $\rho = \delta/\Delta$, and a method is stable iff $\rho < 1$. Molecular explanations are embedded two ways: ECFP fingerprints and a trained GIN encoder, both under an L_2 metric to check that the ordering is not an artifact of a single embedding. Unless otherwise stated all reported values are means over 10 seeds. Finally, a qualitative case study on explanations can be seen in Appendix B to visually verify explanation stability.

6.2 STABILITY ON PLANTED-MOTIF GRAPHS

On synthetic graphs with an exact ground-truth motif (Table 1) ITXGNN attains the highest isomorphism rate on both datasets and the lowest nGED on BA-2Motif; on SPMotif its inter nGED loses mainly to GSAT (also to ProtGNN). Notably GSAT actually has a lower nGED on graphs from differing class graphs than those of a similar class meaning it is generating drastically different graphs for all graphs. However, ITXGNN still has the lowest nGED ratio on SPMotif. The BA-2Motif margin is large: IsoRate 0.970 versus 0.170 for the next-best method thus indicating that same motif graphs receive near isomorphic explanations, while predictive accuracy and AUC remain at or above the baselines. We use an isomorphism gap to gauge stability instead of a ratio since there are numerous methods that have an inter-isomorphism rate of 0. The numbers are even more stark with ITXGNN's IsoRate gap being more than ten times larger than the next best.

6.3 STABILITY ON MOLECULAR GRAPHS

Table 2 reports the mean intra/inter FID ratio ρ. ITXGNN attains the lowest ratio in every dataset and under both embedders. The contrast is sharp against baselines with ECFP as the choice of embedding, where ITXGNN achieves a ratio substantially lower on several datasets. The per dataset breakdowns in Appendix E (Tables 6-7) show the mechanism: ITXGNN combines a relatively small intra-group FID (same-class explanations are tight) with a relatively large inter-group FID (differing class explanations are far apart), which is exactly the δ-small / Δ-large regime the objective is designed to produce.

6.4 PREDICTIVE PERFORMANCE

Table 3 reports classification accuracy and AUC. ITXGNN is competitive throughout and attains the best accuracy on Benzene and MUTAG and the best AUC on MUTAG, confirming that constraining explanations for stability does not cost predictive performance.

Table 1: Stability on synthetic datasets. The primary summaries are IsoRate gap (intra minus inter; higher is better) and nGED ratio (intra/inter; lower is better). Component values are shown for interpretation. Entries include mean ± standard deviation where available.

Dataset	Method	**Iso. gap ↑**	**nGED ratio ↓**	Intra Iso.	Inter Iso.	Intra nGED	Inter nGED
BA-2Motif	**ITXGNN (ours)**	**0.970**	**0.041**	0.970 ± 0.053	0.000 ± 0.000	0.004 ± 0.007	0.097 ± 0.010
	PGExplainer	0.081	0.847	0.084 ± 0.025	0.003 ± 0.005	0.364 ± 0.267	0.430 ± 0.260
	GNNExplainer	0.068	0.777	0.097 ± 0.012	0.029 ± 0.026	0.192 ± 0.026	0.247 ± 0.049
	PGIB	0.044	0.885	0.170 ± 0.182	0.126 ± 0.080	0.371 ± 0.036	0.419 ± 0.032
	ProtGNN	0.011	0.868	0.011 ± 0.003	0.000 ± 0.000	0.486 ± 0.021	0.560 ± 0.010
	GSAT	0.003	0.844	0.003 ± 0.002	0.000 ± 0.000	0.540 ± 0.020	0.640 ± 0.006
SPMotif-0.5	**ITXGNN (ours)**	**0.324**	**0.825**	0.346 ± 0.149	0.022 ± 0.034	0.504 ± 0.097	0.611 ± 0.077
	PGExplainer	0.062	0.940	0.077 ± 0.013	0.015 ± 0.002	0.553 ± 0.012	0.588 ± 0.007
	GNNExplainer	0.019	0.990	0.053 ± 0.006	0.034 ± 0.005	0.593 ± 0.010	0.599 ± 0.014
	PGIB	0.001	0.998	0.005 ± 0.002	0.004 ± 0.003	0.548 ± 0.019	0.549 ± 0.025
	ProtGNN	0.017	0.990	0.033 ± 0.018	0.016 ± 0.015	0.686 ± 0.020	0.693 ± 0.028
	GSAT	0.000	1.007	0.000 ± 0.000	0.000 ± 0.000	0.903 ± 0.007	0.897 ± 0.001

Table 2: Chemical explanation stability under ECFP and GIN embeddings. ρ is the intra-class to inter-class ChemFID ratio (lower is better). ITX, PGE, GNNE, Prot, and SubX denote ITXGNN (ours), PGExplainer, GNNExplainer, ProtGNN, and SubgraphX, respectively. Full intra- and inter-class results are in Appendix E.

	$\rho_{\text{ECFP}} \downarrow$							$\rho_{\text{GIN}} \downarrow$						
Dataset	ITX	PGE	GNNE	PGIB	GSAT	Prot	SubX	ITX	PGE	GNNE	PGIB	GSAT	Prot	SubX
Alkane-Carbonyl	**.095**	.539	.548	.636	.638	.761	.898	**.171**	.330	.382	.474	.439	.339	.548
Benzene	**.026**	.201	.468	.651	.192	.520	.504	**.007**	.027	.190	.149	.029	.090	.205
MUTAG	**.366**	.855	.596	.712	.644	.555	.619	**.107**	.497	.413	.783	.277	.279	.799
Fluoride-Carbonyl	**.282**	.371	.371	.978	.794	.516	.546	**.045**	.108	.166	.894	.454	.297	.139

6.5 Ablation Studies

We conduct ablation studies to isolate the contribution of the prototype layer, stochastic Gumbel mechanism, and learnable clustering to predictive performance and explanation stability. Simply removing a component can render the framework ill-defined, so we use controlled alternatives. We freeze clustering in its initial state, replace the prototype layer with either a linear or 2-layer MLP head, and replace Gumbel-Softmax with deterministic $\mathrm{softmax}(\mathrm{logits}/\tau)$ during training. We compare all ablations to the full framework over 5 seeds (Table 4). Accuracy and AUC are retained except with the MLP head, while explanation stability decreases across the ablations.

Table 3: Predictive performance (mean ± sample standard deviation across seeds). Boldface marks the highest reported mean in each dataset column; ties at the displayed precision are all marked.

(a) Accuracy (↑)

Method	BA-2M	SPM-0.5	MUTAG	Benzene	Alkane-C	Fluoride-C
ITXGNN (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.990 \pm 0.002}$	$\mathbf{0.937 \pm 0.021}$	$\mathbf{0.940 \pm 0.001}$	$\mathbf{0.968 \pm 0.017}$	0.923 ± 0.051
GIN	0.970 ± 0.010	0.775 ± 0.015	0.921 ± 0.026	0.936 ± 0.003	0.947 ± 0.025	$\mathbf{0.963 \pm 0.013}$
GNNExplainer	0.913 ± 0.087	0.836 ± 0.049	0.926 ± 0.026	0.935 ± 0.005	0.963 ± 0.013	0.956 ± 0.004
PGExplainer	0.897 ± 0.132	0.764 ± 0.018	0.895 ± 0.033	0.935 ± 0.004	0.950 ± 0.013	0.960 ± 0.005
GSAT	0.983 ± 0.009	0.840 ± 0.057	0.779 ± 0.052	0.903 ± 0.009	0.947 ± 0.006	0.840 ± 0.029
PGIB	0.883 ± 0.066	0.421 ± 0.096	0.684 ± 0.197	0.645 ± 0.038	0.874 ± 0.082	0.820 ± 0.026
ProtGNN	0.997 ± 0.005	0.794 ± 0.005	0.884 ± 0.077	0.924 ± 0.013	0.947 ± 0.006	0.953 ± 0.006
SubgraphX	0.835 ± 0.145	0.791 ± 0.017	0.895 ± 0.033	0.934 ± 0.010	0.961 ± 0.012	0.959 ± 0.004

(b) AUC (↑)

Method	BA-2M	SPM-0.5	MUTAG	Benzene	Alkane-C	Fluoride-C
ITXGNN (ours)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.998 \pm 0.002}$	$\mathbf{0.974 \pm 0.008}$	0.973 ± 0.003	0.976 ± 0.020	0.963 ± 0.020
GIN	$\mathbf{1.000 \pm 0.000}$	0.892 ± 0.012	$\mathbf{0.974 \pm 0.026}$	$\mathbf{0.986 \pm 0.001}$	0.988 ± 0.009	0.982 ± 0.007
GNNExplainer	$\mathbf{1.000 \pm 0.000}$	0.952 ± 0.028	0.969 ± 0.031	0.985 ± 0.002	0.993 ± 0.003	$\mathbf{0.983 \pm 0.002}$
PGExplainer	$\mathbf{1.000 \pm 0.001}$	0.875 ± 0.025	$\mathbf{0.974 \pm 0.014}$	$\mathbf{0.986 \pm 0.001}$	0.994 ± 0.003	$\mathbf{0.983 \pm 0.003}$
GSAT	$\mathbf{1.000 \pm 0.000}$	0.962 ± 0.020	0.905 ± 0.013	0.967 ± 0.004	0.987 ± 0.010	0.841 ± 0.057
PGIB	0.989 ± 0.011	0.802 ± 0.016	0.879 ± 0.050	0.769 ± 0.016	0.979 ± 0.030	0.934 ± 0.013
ProtGNN	$\mathbf{1.000 \pm 0.000}$	0.855 ± 0.016	0.923 ± 0.020	0.983 ± 0.003	$\mathbf{0.995 \pm 0.005}$	0.971 ± 0.006
SubgraphX	$\mathbf{1.000 \pm 0.000}$	0.899 ± 0.043	0.962 ± 0.018	$\mathbf{0.986 \pm 0.001}$	0.994 ± 0.003	0.982 ± 0.001

Table 4: Ablation study on BA-2Motif over 5 seeds. IsoRate gap is intra-IsoRate minus inter-IsoRate; higher is better. nGED ratio is intra/inter; lower is better. Values are mean ± sample standard deviation.

Variant	Accuracy ↑	AUC ↑	IsoRate gap ↑	nGED ↓	nGED ratio ↓
Full	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.954 \pm 0.124}$	$\mathbf{0.008 \pm 0.021}$	**0.040**
Frozen clustering	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.934 ± 0.062	0.009 ± 0.009	0.088
No Gumbel	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.820 ± 0.240	0.050 ± 0.081	0.156
No prototype (linear)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$	0.746 ± 0.351	0.071 ± 0.142	0.345
No prototype (MLP)	0.850 ± 0.242	0.850 ± 0.242	0.602 ± 0.432	0.206 ± 0.301	0.625

7 Conclusion

We identified explanation stability across instances as an underexplored axis of explanation quality in self-interpretable graph learning: a trustworthy explainer must give consistent explanations to graphs governed by the same phenomenon while distinguishing different ones. We formalized this as a single intra/inter dispersion criterion, $\rho = \delta/\Delta < 1$ (Definition 1), capturing continuity and discrimination while ruling out the constant explainer. We introduced ITXGNN, whose objective maximizes $I(\hat{C}; P)$ alongside $I(P; Y)$, and showed under an idealized common-noise model (Theorem 1) a bound relating prototype information to embedding separation relative to scatter. Empirically, ITXGNN attains the strongest stability across planted motif and molecular benchmarks under both embedders while remaining competitive in accuracy and AUC. Our analysis is limited to two prototypes, and measured stability depends on explanation distance and grouping: extending the theory to multiple prototypes and evaluating under distribution shift remain future work.

AI USE STATEMENT

An LLM was usd to identify grammatical errors and suggest minor wording corrections. The authors reviewed all suggestions and take responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide as much of the method, experimental details, and technical details as possible in the appendix and paper. We will publicly release all code, configurations, and scripts needed to reproduce our results on GitHub.

REFERENCES

Chirag Agarwal, Marinka Zitnik, and Himabindu Lakkaraju. Probing gnn explainers: A rigorous theoretical and empirical analysis of gnn explanation methods. In *International conference on artificial intelligence and statistics*, pp. 8969–8996. PMLR, 2022.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. *Advances in Neural Information Processing Systems*, 34:5644–5655, 2021.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

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

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. *Advances in neural information processing systems*, 36:72446–72463, 2023.

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): 8687–8698, 2022.

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

Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

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, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Yang Ji, Ying Sun, Yuting Zhang, Zhigaoyuan Wang, Yuanxin Zhuang, Zheng Gong, Dazhong Shen, Chuan Qin, Hengshu Zhu, and Hui Xiong. A comprehensive survey on self-interpretable neural networks. *arXiv preprint arXiv:2501.15638*, 2025.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Jiate Li, Meng Pang, Yun Dong, Jinyuan Jia, and Binghui Wang. Graph neural network explanations are fragile. *arXiv preprint arXiv:2406.03193*, 2024.

Jiate Li, Meng Pang, Yun Dong, Jinyuan Jia, and Binghui Wang. Provably robust explainable graph neural networks against graph perturbation attacks. *arXiv preprint arXiv:2502.04224*, 2025.

Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 4499–4511. PMLR, 2022.

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

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

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

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Phillip 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*, pp. 10772–10781, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for nlp with differentiable edge masking. *arXiv preprint arXiv:2010.00577*, 2020.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Sangwoo Seo, Sungwon Kim, and Chanyoung Park. Interpretable prototype-based graph information bottleneck. *Advances in Neural Information Processing Systems*, 36, 2024.

Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pp. 9269–9278. PMLR, 2020.

Wenxin Tai, Ting Zhong, Goce Trajcevski, and Fan Zhou. Redundancy undermines the trustworthiness of self-interpretable gnns. In *Forty-second International Conference on Machine Learning*, 2025.

Wenxin Tai, Ting Zhong, Goce Trajcevski, and Fan Zhou. Self-consistency improves the trustworthiness of self-interpretable gnns. In *The Fourteenth International Conference on Learning Representations*, 2026.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

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

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

Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872*, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

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

Junchi Yu, Tingyang Xu, Yu Rong, Yatao Bian, Junzhou Huang, and Ran He. Graph information bottleneck for subgraph recognition. *arXiv preprint arXiv:2010.05563*, 2020.

Junchi Yu, Jie Cao, and Ran He. Improving subgraph recognition with variational graph information bottleneck. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19396–19405, 2022.

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*, pp. 12241–12252. PMLR, 2021.

Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9127–9135, 2022.

A PROOFS

Proof of Theorem 1:

Proof. Recall that each prototype and embedding have a probability belonging to cluster 1 denoted by:

$$\begin{aligned} r_{j,1} &= \frac{e^{-||z_j-\mu_1||_2^2}}{\sum_i e^{-||z_j-\mu_i||_2^2}} \\ &= \frac{e^{-||z_j-\mu_1||_2^2}}{e^{-||z_j-\mu_1||_2^2} + e^{-||z_j-\mu_2||_2^2}} \\ &= \frac{1}{1+e^{||z_j-\mu_1||_2^2-||z_j-\mu_2||_2^2}} \end{aligned}$$

Note that we can express:

$$\begin{aligned} ||x-y||_2^2 &= (x-y)^\top (x-y) \\ &= ||x||_2^2 + ||y||_2^2 - 2x^\top y \end{aligned}$$

by applying it to the above we get:

$$\begin{aligned} \frac{1}{1+e^{||z_j-\mu_1||_2^2-||z_j-\mu_2||_2^2}} &= \frac{1}{1+e^{||z_j||_2^2+||\mu_1||_2^2-2z_j^\top\mu_1-||z_j||_2^2-||\mu_2||_2^2+2z_j^\top\mu_2}} \\ &= \frac{1}{1+e^{||\mu_1||_2^2-||\mu_2||_2^2-2z_j^\top(\mu_1-\mu_2)}} \\ &= \frac{1}{1+e^{(\mu_1-\mu_2)^\top(\mu_2+\mu_1)-2z_j^\top(\mu_1-\mu_2)}} \\ &= \frac{1}{1+e^{2(\mu_1-\mu_2)^\top\left(\frac{\mu_2+\mu_1}{2}-z_j\right)}} \end{aligned}$$

Now recall: $\gamma = \|\mu_1 - \mu_2\|$, and $\hat{\mu} = (\mu_1 - \mu_2)/\gamma$ with $t = \langle z, \hat{\mu}\rangle$

Since $(\mu_1 - \mu_2) = \gamma\hat{\mu}$ we have:

$$\begin{aligned} 2z_j^\top(\mu_1-\mu_2) &= 2\gamma\langle z_j, \hat{\mu}\rangle \\ &= 2\gamma t \end{aligned}$$

For the remaining term, lets introduce m as $m = \langle\frac{\mu_1+\mu_2}{2}, \hat{\mu}\rangle$ so the other term becomes:

$$\begin{aligned} 2(\mu_1-\mu_2)^\top\frac{\mu_2+\mu_1}{2} &= 2\gamma\langle\frac{\mu_2+\mu_1}{2}, \hat{\mu}\rangle \\ &= 2\gamma m \end{aligned}$$

Therefore we can write the previous equation as:

$$\begin{aligned} \frac{1}{1+e^{2(\mu_1-\mu_2)^\top\left(\frac{\mu_2+\mu_1}{2}-z_j\right)}} &= \frac{1}{1+e^{2\gamma(m-t)}} \\ &= \sigma(2\gamma(t-m)) \end{aligned}$$

$$r_{j,1}(z_j) = \sigma(2\gamma(t - m)) \tag{7}$$

Where $\sigma(\cdot)$ denotes the sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$. Therefore, we have now shown that we have reduced the prototype posterior $r_{i,1}$ to a 1-dimensional logistic form that is a function of t alone. So P depends on z_j only through t. This gives us a Markov chain from the clustering to t to P (the prototype assignment): $\hat{C} \longrightarrow t \longrightarrow P$. We can now use the data processing inequality:

$$I(\hat{C}; t) \geq I(\hat{C}; P) \tag{8}$$

Now let

$$\begin{aligned}\tau_k &= \mathbb{E}[\langle z, \hat{\mu}\rangle | \hat{C} = k] \\ &\quad \text{(by linearity of expectation)} \\ &= \langle \mathbb{E}[z | C \hat{=} k], \hat{\mu}\rangle \\ &= \langle \mu_k, \hat{\mu}\rangle\end{aligned}$$

recall, z is an embedding, and since

$$\begin{aligned}\mathbb{E}[z|\hat{C} = k] &= \frac{\sum_i q_{i,k} z_i}{\sum_i q_{i,k}} \\ \Rightarrow |\tau_2 - \tau_1| &= |\langle \mu_1 - \mu_2, \hat{\mu}\rangle| = \gamma\end{aligned}$$

The two conditional means of t are τ_1, τ_2. Now view $\hat{C} \to t$ as a signal. Lets add noise (ε) assume its Gaussian with variance s^2 (using assumption 1).

Recall we defined τ_k as the mean location of cluster $\hat{C} = k$ along the t-axis. So,

$$\begin{aligned}\tau_1 &= \mathbb{E}[t|\hat{C} = 1] \\ \tau_2 &= \mathbb{E}[t|\hat{C} = 2]\end{aligned}$$

Now lets rewrite $X := \tau_{\hat{C}}$

$$X = \begin{cases}\tau_1, & \text{if cluster 1: } \hat{C} = 1, \\ \tau_2, & \text{if cluster 2: } \hat{C} = 2.\end{cases}$$

then,

$$t = X + \varepsilon \quad \text{where } \varepsilon \sim \mathcal{N}(0, s^2)$$

Cluster identity chooses a signal:

$$\hat{C} \longrightarrow X$$

noise gets added:

$$X + \varepsilon$$

and we observe t. Then we center two cluster midpoints; one cluster midpoint is at $+\frac{\gamma}{2}$ and another at $-\frac{\gamma}{2}$. So we center the two signal levels at their midpoints and they sit at $+\frac{\gamma}{2}$ and another at $-\frac{\gamma}{2}$. Therefore their squared magnitude is:

$$(\frac{\gamma}{2})^2$$

Recall, via our assumption the noise variance is s^2 (Assumption 1), so the signal to noise ratio is:

$$\text{SNR} = \frac{(\gamma/2)^2}{s^2} = \frac{\gamma^2}{4s^2}$$

A Gaussian channel with this SNR has capacity:

$$\frac{1}{2}\log(1 + \text{SNR})$$

A fundamental theorem in Information Theory gives us (Cover, 1999):

Theorem 2. *For an additive Gaussian channel $Y = X + N$ with $N \sim \mathcal{N}(0, \sigma^2)$ and power P,*

$$\begin{aligned} I(X;Y) &\leq \frac{1}{2}\log(1 + \frac{P}{\sigma^2}) \\ &= \frac{1}{2}\log(1 + \textit{SNR}) \end{aligned}$$

where, N is independent of X

So by applying Theorem 2 this tells us that our binary input channel cannot exceed this capacity:

$$\begin{aligned} I(\hat{C};t) &\leq \frac{1}{2}\log(1 + \text{SNR}) \\ &= \frac{1}{2}\log(1 + \frac{\gamma^2}{4s^2}) \end{aligned}$$

Combining with the previous bound we get in eq (8). Rearranging, we get:

$$\begin{aligned} I(\hat{C};P) \leq I(\hat{C};t) &\leq \frac{1}{2}\log(1 + \frac{\gamma^2}{4s^2}) \\ 2I(\hat{C};P) &\leq \log(1 + \frac{\gamma^2}{4s^2}) \\ e^{2I(\hat{C};P)} &\leq (1 + \frac{\gamma^2}{4s^2}) \\ e^{2I(\hat{C};P)} - 1 &\leq \frac{\gamma^2}{4s^2} \\ &\implies \\ \frac{\gamma}{s} &\geq 2\sqrt{e^{2I(\hat{C};P)} - 1} \end{aligned}$$

and we have the desired result in Theorem 1. □

B QUALITATIVE CASE STUDIES

In this section, we have a case study showcasing qualitative analysis on explanation quality. We select a representative synthetic dataset (BA2Motif) and a representative molecular dataset (Benzene). We use 3 baselines; GSAT, GNNExplainer, and PGIB. PGIB is one of the more recent developments in self-interpretable GNNs, we note that GSAT is the strongest of the baselines, and GNNExplainer is a representative post-hoc explainer. We use non-cherry picked examples in our case study of 4 graphs per class to demonstrate visually selected explanations (nodes/edges highlighted in red are selected and greyed out nodes are not selected) in comparison to the ground truth (highlighted blue nodes). In addition, to each graph we provide some statistics such as 'GT Jaccard' which refers to ground truth Jaccard. This is a measure (if ground truth is applicable) the Jaccard Index measuring the overlap between the ground truth nodes and the selected explanation nodes.

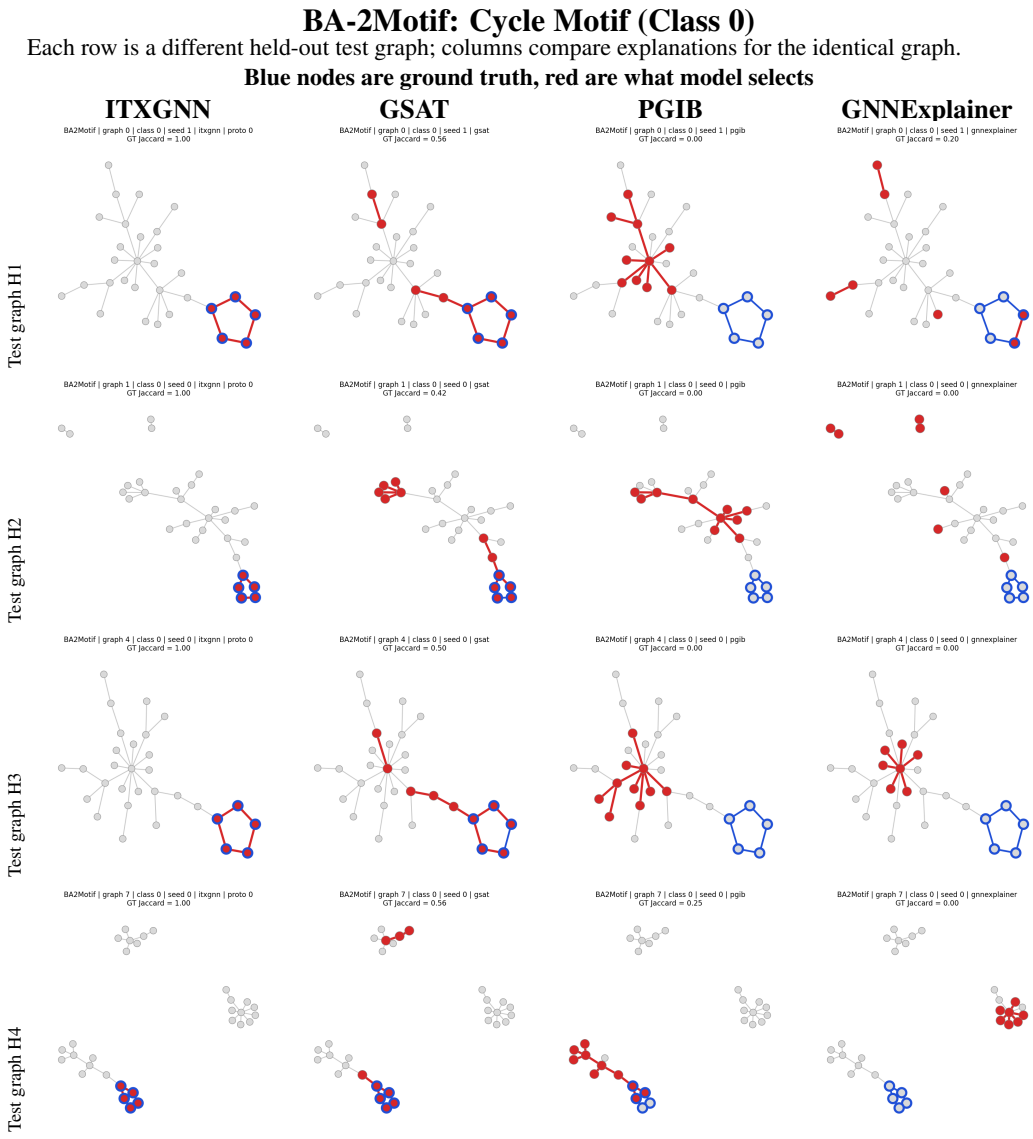

Figure 2: Qualitative comparison on four distinct heldout BA2Motif graphs from the cycle motif class (class 0). Each row corresponds to one fixed graph evaluated by every method. Red denotes the selected explanation; blue denotes the ground truth motif.

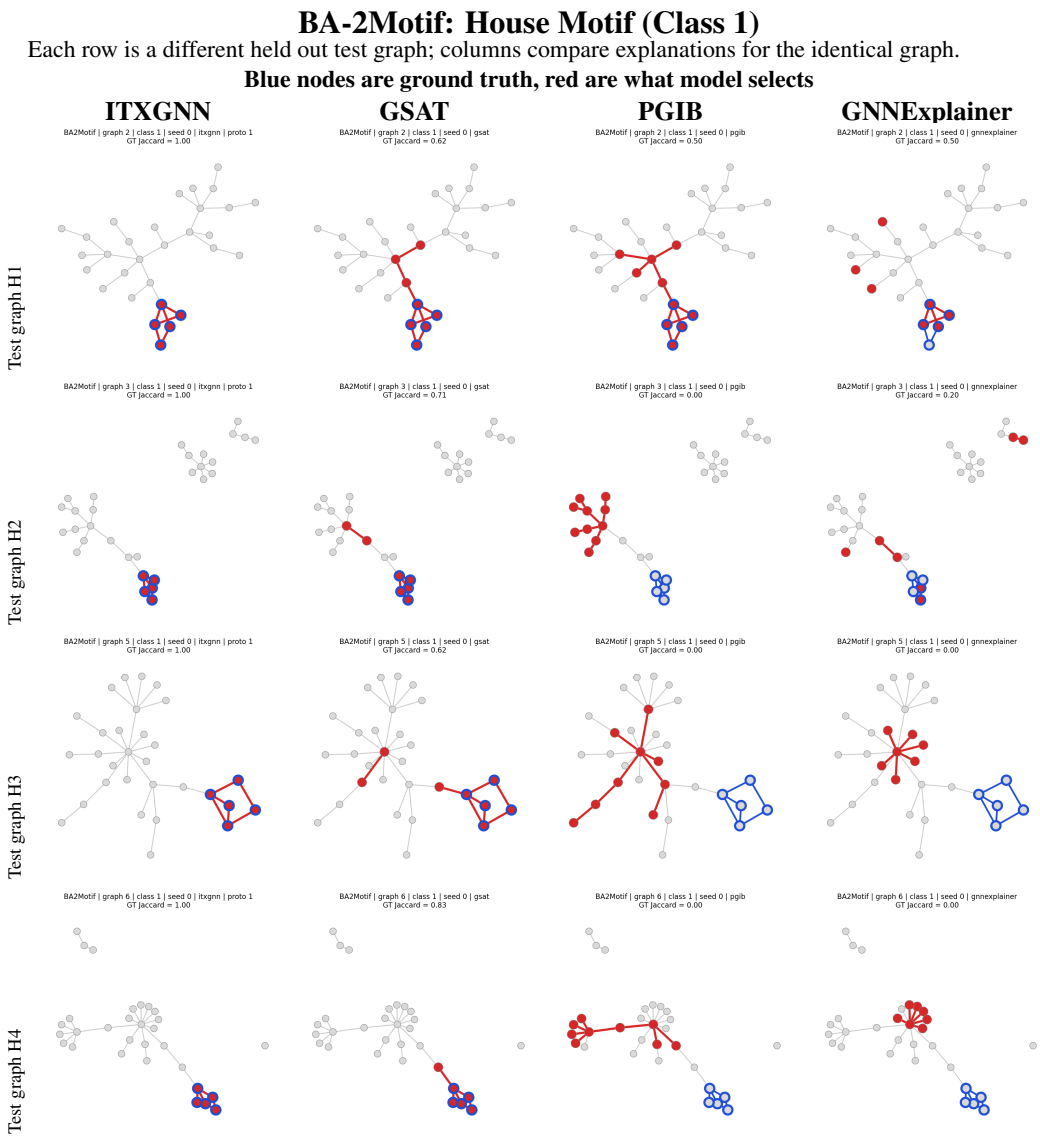

Figure 3: Qualitative comparison on four distinct held out BA2Motif graphs from the house motif class (class 1). Each row corresponds to one fixed graph evaluated by every method. Red denotes the selected explanation; blue denotes the ground truth motif.

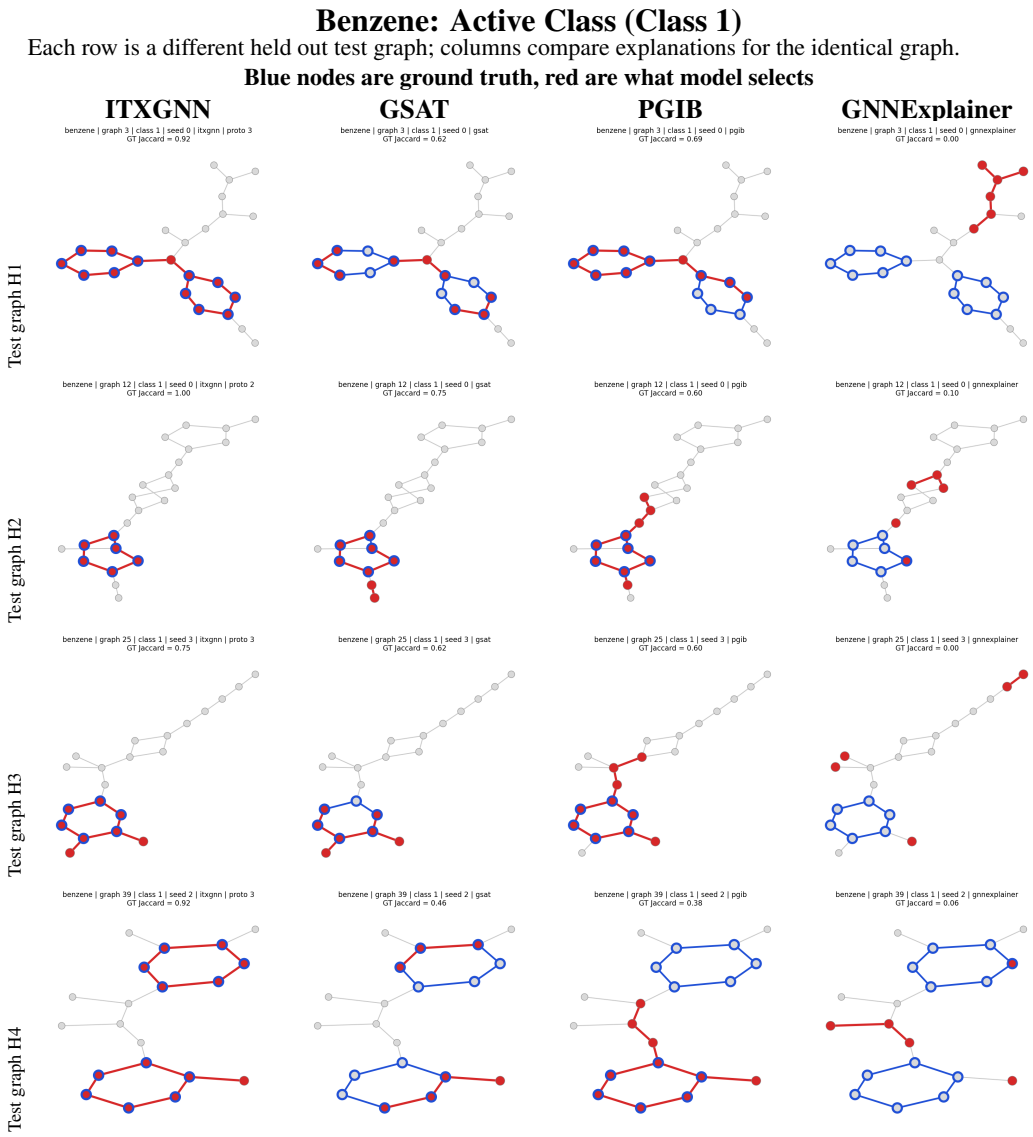

Figure 4: Qualitative comparison on four distinct held out Benzene graphs from the active class (class 1). Each row corresponds to one fixed graph evaluated by every method. Red denotes the selected explanation; blue denotes the ground truth motif.

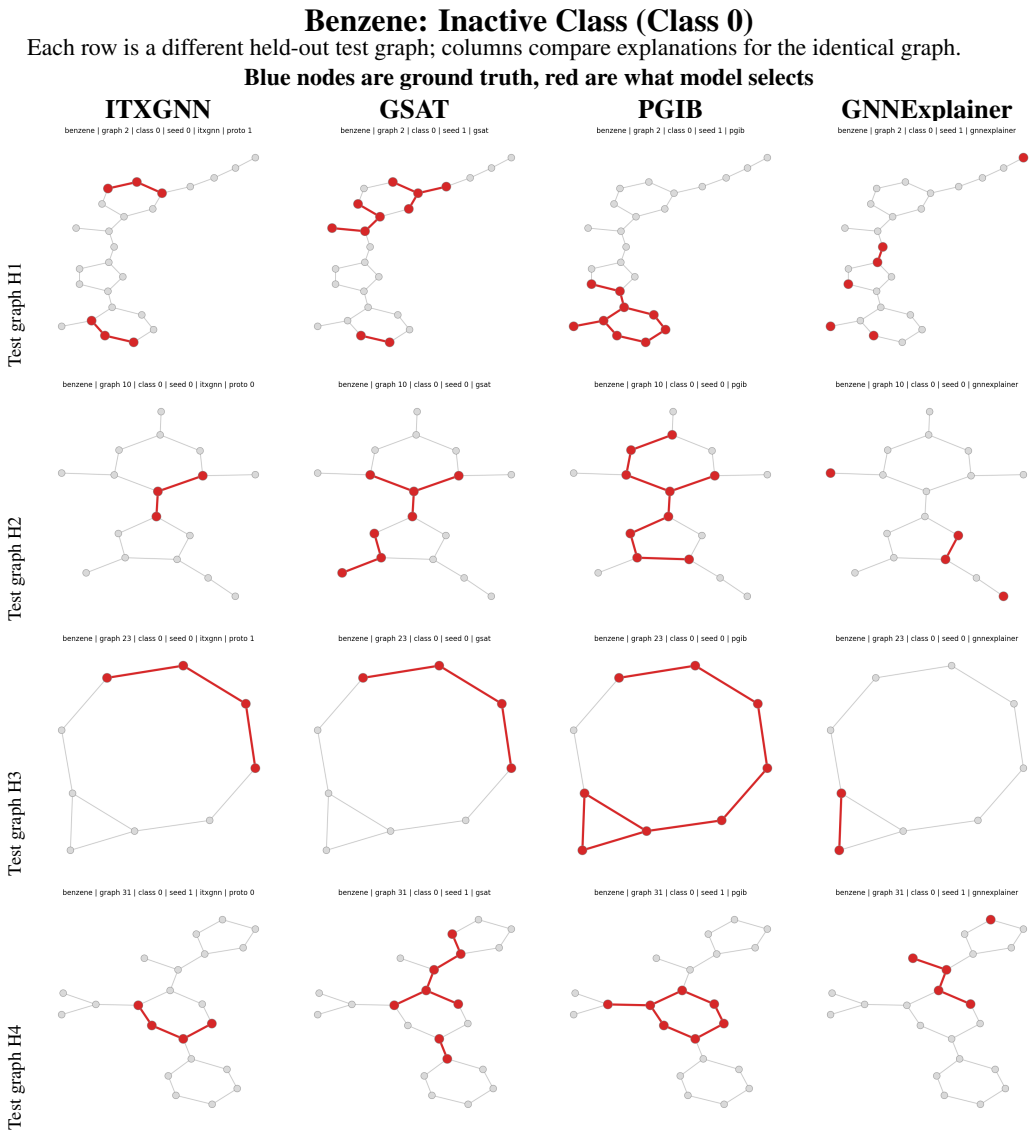

Figure 5: Qualitative comparison on four distinct held out Benzene graphs from the inactive class (class 0). Each row corresponds to one fixed graph evaluated by every method. Red denotes the selected explanation; blue denotes the ground truth motif; in this case there is no ground truth.

C TECHNICAL DERIVATION

C.1 THE FRAMEWORK

C.1.1 SUBGRAPH EXTRACTOR

Our subgraph extractor scores nodes using attention scores. In general, attention scores are computed by forming a query from graph-level context, denoted $\mathbf{g}_{b_u}$, and a key from the node-level score representation $\mathbf{h}_u^{\text{score}}$ (this is given by an MLP):

$$\mathbf{q}_u = W_Q \mathbf{g}_{b_u} + \mathbf{b}_Q,$$
$$\mathbf{k}_u = W_K \mathbf{h}_u^{\text{score}} + \mathbf{b}_K.$$

The attention score is then computed through a scaled dot product:

$$a_u = \frac{\mathbf{k}_u^\top \mathbf{q}_u}{\sqrt{d_h}}, \tag{9}$$

Since raw attention scores can vary in scale across graphs we conduct graph wise normalization independently within each graph. We then pass each normalized attention scalar score $\hat{a}_u$ through a gate MLP to obtain logits, $\mathbf{l}_u = [l_u^{(0)}, l_u^{(1)}]$, corresponding to dropping or keeping a node respectively. Now we apply a Gumbel-Softmax operator on the logits to obtain a differential approximation to a binary node selection mask:

$$\mathbf{m}_i = \text{GumbelSoftmax}(\boldsymbol{\ell}_i; \tau),$$

where τ is the temperature parameter. Note that we use a linear τ schedule.

C.1.2 PROTOTYPE LAYER

The prototype update is similar to an EM style in the following sense: the current cluster responsibilities q_{ik} act as soft assignments, and the prototype is recomputed as the weighted mean of the current subgraph embeddings in that cluster. This is not a full generative EM algorithm, but it is an alternating mean-update procedure analogous to the M-step for soft clustering. This allows for the clustering and the prototypes to be flexible and learnable; by exploiting the underlying patterns in the data to guide the overall learning objective towards a more favorable choice of clustering and set of prototypes. Given a graph embedding $\mathbf{z}_i$ and a prototype $\boldsymbol{\mu}_k$ the similarity is defined as $s_{i,k} = sim(\mathbf{z}_i, \boldsymbol{\mu}_k)$ where $sim(\cdot)$ denotes a similarity measure between two embeddings. For the Euclidean configurations, we use negative squared Euclidean distance as the score; the dataset specific distance choices are listed in Appendix D.

C.2 DETAILED DERIVATION OF OBJECTIVE FUNCTION

We first assume the dataset is empirical and each graph is equally likely we then have for a dataset of N graphs:

$$\mathbb{P}(G_i) = \frac{1}{N}.$$

Recall that for each graph G_i, the model provides empirical conditional distributions of selecting prototype k and selecting cluster l given the input graph respectively as:

$$r_{ik} = \mathbb{P}(p = k \mid G_i) \qquad q_{i\ell} = \mathbb{P}(c = \ell \mid G_i)$$

By the law of total probability, we can obtain a marginal prototype distribution and cluster distribution respectively:

$$\mathbb{P}(p = k) = \sum_{i=1}^{N} \mathbb{P}(p = k \mid G_i)\mathbb{P}(G_i) = \frac{1}{N}\sum_{i=1}^{N} r_{ik}.$$
$$\mathbb{P}(c = \ell) = \sum_{i=1}^{N} \mathbb{P}(c = \ell \mid G_i)\mathbb{P}(G_i) = \frac{1}{N}\sum_{i=1}^{N} q_{i\ell}.$$

Using the graph identity as the source of randomness, the empirical joint distribution between prototype and cluster is:

$$\mathbb{P}(p = k, c = \ell) = \sum_{i=1}^{N} \mathbb{P}(p = k, c = \ell \mid G_i)\mathbb{P}(G_i).$$

Where a natural factorization is:

$$\mathbb{P}(p = k, c = \ell \mid G_i) = \mathbb{P}(p = k \mid G_i)\mathbb{P}(c = \ell \mid G_i) = r_{ik}q_{i\ell}.$$

Therefore:

$$P(p = k, c = \ell) = \frac{1}{N}\sum_{i=1}^{N} r_{ik}q_{i\ell}.$$

D DATASETS AND IMPLEMENTATION DETAILS

Dataset	Graphs	Avg. $\lvert V\rvert$	Avg. $\lvert E\rvert$	Classes	Node / edge dim	Train / val / test
BA-2Motif	1,000	30.0	28.9	2	10 / —	800 / 100 / 100
MUTAG	188	17.9	19.8	2	7 / 4	150 / 19 / 19
SPMotif-0.5	12,000	23.3	33.3	3	4 / —	9,000 / 300 / 2,700
Benzene	12,000	20.6	21.8	2	14 / 5	9,600 / 1,200 / 1,200
Alkane-Carbonyl	1,125	21.3	22.6	2	14 / 5	899 / 113 / 113
Fluoride-Carbonyl	8,671	21.4	22.7	2	14 / 5	6,935 / 868 / 868

Table 5: Implementation Details and Hyperparameters

	BA-2Motif	MUTAG	SPMotif	Benzene	Alkane	Fluoride
Prototypes K	2	4	6	4	4	4
Prototype distance	Euc.	cos	Euc.	Euc.	cos	cos
Parameters	46.3k	48.3k	99.7k	112.4k	85.3k	101.7k
Epochs	500	500	500	500	500	500
Batch size	64	150†	64	64	64	64
Gradient clip	1	1	5	5	5	5
λ_{MI}	1	1	1	1	1	1
Class-weighted CE	no	no	no	no	yes	yes
Gumbel τ_{start}	2	3	2	1.5	1.5	1.5
Gumbel τ_{end}	0.1	0.3	0.1	0.05	0.05	0.05
Hidden dim	64					
RWPE dim	20					
Optimizer	Adam, weight decay 10^{-5}					
Learning rate	10^{-3} (scorer, validator), 10^{-1} (clustering), 10^{-2} (prototypes)					
Gumbel schedule	linear anneal of τ over training; soft relaxation					
Node selection	Gumbel-softmax					
Hardware / software	NVIDIA RTX 3060, PyTorch 2.4.0, PyG 2.6.1					

† Full batch.

E MISCELLANEOUS DATA

Table 6. Chemical stability under ECFP embeddings. Values are means. Lower intra-ChemFID and $\rho = \mathrm{Intra}/\mathrm{Inter}$ are better; higher inter-ChemFID is better.

Dataset	Method	Intra ↓	Inter ↑	ρ ↓
Alkane	ITXGNN (ours)	**1.04**	11.04	**0.095**
	PGExplainer	4.47	8.29	0.539
	GNNExplainer	3.69	6.73	0.548
	PGIB	17.33	27.25	0.636
	GSAT	31.33	**49.14**	0.638
	ProtGNN	30.98	40.72	0.761
	SubgraphX	30.09	33.52	0.898
Benzene	ITXGNN (ours)	**3.16**	**119.43**	**0.026**
	GSAT	9.29	48.43	0.192
	PGExplainer	4.59	22.77	0.201
	GNNExplainer	3.58	7.66	0.468
	SubgraphX	17.95	35.65	0.504
	ProtGNN	23.17	44.55	0.520
	PGIB	8.42	12.94	0.651
MUTAG	ITXGNN (ours)	64.17	**175.13**	**0.366**
	ProtGNN	43.80	78.86	0.555
	GNNExplainer	**6.16**	10.33	0.596
	SubgraphX	33.62	54.35	0.619
	GSAT	36.11	56.10	0.644
	PGIB	28.46	39.95	0.712
	PGExplainer	36.74	42.98	0.855
Fluoride Carbonyl	ITXGNN (ours)	**0.1005**	0.3564	**0.282**
	PGExplainer	0.1914	0.5160	0.371
	GNNExplainer	0.2249	0.6062	0.371
	PGIB	0.3757	0.3842	0.978
	GSAT	0.2451	0.3088	0.794
	ProtGNN	0.3394	**0.6577**	0.516
	SubgraphX	0.2175	0.3983	0.546

Table 7. Chemical stability under GIN embeddings. Values are means. Lower intra-ChemFID and $\rho = \mathrm{Intra}/\mathrm{Inter}$ are better; higher inter-ChemFID is better.

Dataset	Method	Intra ↓	Inter ↑	ρ ↓
Alkane	ITXGNN (ours)	**6.60**	38.58	**0.171**
	PGExplainer	29.73	90.22	0.330
	ProtGNN	40.30	**118.87**	0.339
	GNNExplainer	20.33	53.16	0.382
	GSAT	35.66	81.20	0.439
	PGIB	49.93	105.39	0.474
	SubgraphX	39.23	71.59	0.548
Benzene	ITXGNN (ours)	**5.07**	**692.92**	**0.007**
	PGExplainer	6.20	232.73	0.027
	GSAT	6.07	210.67	0.029
	ProtGNN	21.83	242.04	0.090
	PGIB	11.65	77.97	0.149
	GNNExplainer	14.03	74.00	0.190
	SubgraphX	24.40	119.10	0.205
MUTAG	ITXGNN (ours)	65.69	**615.59**	**0.107**
	GSAT	62.17	224.39	0.277
	ProtGNN	**49.13**	176.21	0.279
	GNNExplainer	53.51	129.70	0.413
	PGExplainer	80.79	162.61	0.497
	PGIB	57.31	73.24	0.783
	SubgraphX	225.56	282.23	0.799
Fluoride Carbonyl	ITXGNN (ours)	0.0167	**0.3701**	**0.045**
	PGExplainer	0.0216	0.1997	0.108
	GNNExplainer	0.0279	0.1682	0.166
	PGIB	0.0126	0.0141	0.894
	GSAT	0.0083	0.0182	0.454
	ProtGNN	**0.0058**	0.0195	0.297
	SubgraphX	0.0149	0.1075	0.139