STATISTICAL SUBGRAPH EXPLANATIONS FOR GRAPH NEURAL NETWORKS VIA LATENT NODE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-hoc explanations of graph neural networks (GNNs) often rely on learned edge masks or combinatorial search, leaving the statistical evidence for the selected structure implicit. We recast explanatory subgraph discovery as latent node inference and investigate when departures from background connectivity support explanations of a fixed GNN. We develop two complementary methods without training an auxiliary neural explainer. HierExplainer uses variational inference under a Bernoulli block model for each specific target to estimate posterior node-membership probabilities. HypoExplainer screens prediction label pairs of GNN using edge distribution after Bonferroni correction tests against a supervised background reference, then constructs explanations within each target's computation graph. Our theory distinguishes structural recovery from predictive faithfulness: under explicit compatibility assumptions, HierExplainer's posterior objective combines prediction fidelity with structural regularization, while HypoExplainer admits an asymptotic recovery guarantee under edge independence and minimal signal conditions. Experiments assess recovery, faithfulness, stability, and resource cost on synthetic motif benchmarks and a protein-protein interaction network.

1 INTRODUCTION

Graph neural networks combine node features with local graph structure through message passing. Explaining which part of a graph supports a prediction is important in applications such as biological networks and recommender systems.

Since the full computation graph often contains redundant edges and nodes, a post-hoc GNN explainer aims to identify a compact explanatory subgraph G_S that is sufficient for the target prediction and whose removal changes the prediction. Following the post-hoc setting in Burkart & Huber (2021), the GNN Φ is fixed and the explainer uses Φ and its output to select G_S. GNNExplainer (Ying et al., 2019) maximizes the mutual information between the original prediction and the prediction obtained from a masked subgraph, and then optimizes edge masks by gradient descent. Post-hoc GNN explanation is commonly formulated as a search for a faithful, sparse subgraph that preserves a trained model's prediction. Its illustration is shown in Figure 1. GNNExplainer (Ying et al., 2019), for example, learns continuous masks over edges and features by optimizing a mutual-information objective. Later methods parameterize explainers across instances, perturb local variables, estimate Shapley-style attributions, or search for counterfactual subgraphs. Many of these approaches still require an additional neural explainer, a high-dimensional edge-mask optimization, or an expensive search. They also provide limited statistical information about whether the selected structure differs from background connectivity.

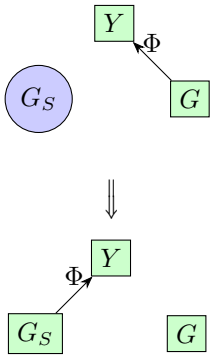


Figure 1: Post-hoc explanation: the fixed GNN predicts from the full graph (top) and a candidate explanatory subgraph (bottom).

We study post-hoc GNN explanation in two conditional regimes. When a target computation graph plausibly follows a two-state Bernoulli block model, HierExplainer uses variational inference to

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.

assign latent structural node scores. When a full local generative model is not assumed, but connectivity among GNN predictions can be counted on disjoint samples, HypoExplainer tests whether edges between predicted-label pairs differ from background edges. It then selects structures associated with rejected label pairs within each target node's computation graph. Both methods seek a local subgraph whose statistically distinctive structure is linked to the trained GNN's prediction. We state the assumptions under which the selected latent node set corresponds to the mutual-information objective commonly used for post-hoc explanation.

We evaluate HierExplainer and HypoExplainer on synthetic and real-world graphs using parameter recovery, motif-recovery AUC and F_1, fidelity-style sufficiency and necessity metrics, sparsity, and runtime. On synthetic data, HierExplainer accurately estimates latent-block parameters, and HypoExplainer recovers motif structures. The main real-data analysis uses a protein-protein interaction graph; the Football structural sanity check is reported in the appendix. Our contributions are:

- We formulate candidate subgraph generation as conditional structural inference in two explicit regimes, without training another neural explainer.
- We develop HierExplainer's latent-block scoring and HypoExplainer's test-based local label-pair screening to provide statistical evidence for explanatory subgraphs.
- We connect both statistical objectives to mutual information under explicit assumptions, placing variational estimation and hypothesis testing within the post-hoc GNN explanation framework.

2 Related Work

GNNExplainer (Ying et al., 2019) learns edge and feature masks through a mutual-information objective; PGExplainer (Luo et al., 2020) trains a shared edge explainer; and SubgraphX (Yuan et al., 2021) searches subgraphs using Shapley-inspired rewards. PGM-Explainer (Vu & Thai, 2020) fits a local probabilistic graphical model, GraphSVX (Duval & Malliaros, 2021) computes Shapley-value attributions, and CF-GNNExplainer (Lucic et al., 2022) produces counterfactual subgraphs. Our methods instead use explicit structural assumptions to generate candidate node-induced explanations. The stochastic block model (Holland et al., 1983) and variational graph-mixture inference (Daudin et al., 2008) provide the probabilistic foundation for HierExplainer.

3 Problem formulation

Graph neural networks: Consider a graph $G = (V, A)$, where V is the vertex set and A is the adjacency matrix with $A_{ij} = 1$ indicating an edge between vertices i and j. Graph neural networks (GNNs) extract patterns from graph-structured data for tasks such as node classification and graph classification. This paper focuses on node-level explanation. Let $\Phi(G, X; W)$ be a trained GNN with node-feature matrix X, learned parameters W, and predicted label distribution $P_\Phi(Y \mid G, X)$. For a target node v, the relevant input is its local computation graph induced by message passing.
Node inference: Following Ying et al. (2019), the explanation algorithm aims to find G_S by maximizing the mutual information between the output of the fixed GNN on the full graph G and on the explanatory subgraph G_S.

In practice, the mutual-information objective is optimized through a cross-entropy or fidelity surrogate between the original predicted class and the prediction on the candidate subgraph. We assume that Y is determined by the nodes and edges in G_S through the trained GNN's local computation graph. For node v, let $\mathcal{N}$ be its neighborhood, let n_v be the size of $\mathcal{N}$, and let $\mathcal{N}_S$ be the node set of G_S. Define $\mathbf{Z}_S = \{Z_j\}_{j=1}^{n_v}$ to indicate whether each node in the neighborhood belongs to G_S:

$$Z_j = \begin{cases} 1 & j \in \mathcal{N}_S \\ 0 & j \notin \mathcal{N}_S \end{cases}$$

The edge set in G_S contains the edges in G connecting nodes in $\mathcal{N}_S$. Identifying G_S can then be written as the following latent-node selection problem:

$$\max_{\mathbf{Z}_S} Q_\Phi(\mathbf{z}) = \max_{\mathbf{Z}_S} \sum_{c=1}^{C} \mathbf{1}[y = c] \log P_\Phi\left(Y = y \mid A, \mathbf{Z}_S\right) \tag{1}$$

We consider two cases: A is generated by $\mathbf{Z}_S$, or no explicit generative relation between the two variables is assumed.

4 CONDITIONAL STRUCTURAL METHODS

4.1 HIEREXPLAINER

HierExplainer is applied locally and target by target. For a target node v, we first extract its GNN computation neighborhood $\mathcal{N}$ and the corresponding local adjacency matrix A. Within this local graph, $\mathbf{Z}_S$ defines a two-block partition: the block containing v is the candidate explanatory block and the remaining nodes form a background block. The target is anchored in the explanatory block, i.e., $Z_v = 1$. Thus, for this target, recovering all nodes assigned to the same latent block as v is equivalent to recovering the node set $\mathcal{N}_S$ of the candidate explanation G_S.

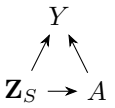


Figure 2: Relation among Y, Z, and A when A is generated by Z.

The two-block model is a one-versus-background working model rather than a global multi-community SBM. Edges whose two endpoints are both in the explanatory block are generated with probability $\pi_{S,c}$, whereas every other local node pair is governed by the background probability $\pi_{B,c}$. This target-specific construction is repeated separately for different target nodes. For a single-node post-hoc explanation, only one target-specific pass is required.

From a Bayesian perspective, Z_j is a latent Bernoulli variable. HierExplainer estimates $\mathbf{Z}_S$ from its structural posterior $P(\mathbf{Z}_S \mid A, Y = y_o)$ under the local two-block model, with the pretrained GNN output used to initialize the variational membership probabilities. We assume the following hierarchical relation between A and $\mathbf{Z}_S$:

Assumption 1. *Assume that A is generated by $\mathbf{Z}_S$, as shown in Figure 2.*

Assumption 2. *The nodes in G_S are connected with probability $\pi_{S,c}$ and background edges are connected with probability $\pi_{B,c}$:*

$$\begin{aligned} P(A_{ij} = 1 \mid Z_i Z_j = 1, y_i = c) &= \pi_{S,c} \\ P(A_{ij} = 1 \mid Z_i Z_j = 0, y_i = c) &= \pi_{B,c}. \end{aligned} \tag{2}$$

Many social network models are based on Assumption 1. To model the fact that edges inside G_S follow a different distribution from background edges, we use a simple SBM-type structure. Other explicit edge-generating models could also be used under Assumption 1. For target node i, nodes in G_S form an explanatory block within its neighborhood, while nodes outside G_S form a background block. We assume that $Z_j \mid Y$ is Bernoulli with $P(Z_j = 1 \mid y_i = c) = \alpha_c$ for $j \in \mathcal{N}$. Connections among nodes follow Assumption 2 defined.

With above assumptions, We develop Hierexplainer as following steps:
Initialize and Localize: To extract explanation subgraph of a target node v, its neighborhood $\mathcal{N}$, Adjacency matrix A, GNN output y_o, trained model Φ are input. The computation neighborhood $\mathcal{N}$ of target v is extract from its induced local adjacency matrix A.
Structure Fitting: We use variational inference to approximate $P(\mathbf{Z}_S \mid A, Y = y_o)$ to select nodes among explanation subgraph. Define the evidence lower bound as

$$\mathcal{J}(\mathbf{Z}_S \mid Y) = \mathbb{E}_{\mathbf{Z}_S|Y} \log P(A, \mathbf{Z}_S \mid Y) - \mathbb{E}_{\mathbf{Z}_S|Y} \log R(\mathbf{Z}_S \mid Y). \tag{3}$$

The full joint likelihood of $\log P(A, \mathbf{Z}_S \mid y_i = c)$ is provided in Appendix A. Under a mean-field assumption, $R(\mathbf{Z}_S \mid y_i = c) = \prod_{j \in \mathcal{N}} h(Z_j \mid y_i = c; \tau_{jc})$, where τ_{jc} is the variational probability that node j is explanatory for class c. To optimize (3), we initialize $\tau_{jc}^{(0)} = P_\Phi(Y_j = c)$, the prediction probability from the pretrained GNN. Under this setting, identifying G_S is achieved by estimating $\hat{\boldsymbol{\theta}}_c$ through:

$$\hat{\boldsymbol{\theta}}_c = (\hat{\alpha}_c, \hat{\pi}_{S,c}, \hat{\pi}_{B,c}, \hat{\tau}_{jc}) = \text{argmax} \mathcal{J}(\mathbf{Z}_S \mid y_i = c). \tag{4}$$

For the current target class c, $\hat{\tau}_{jc}$ is the estimated posterior probability that node j belongs to the explanatory block.

Recovery Explanation: We recover the binary indicator by the posterior decision rule $\hat{Z}_j = \mathbf{1}[\hat{\tau}_{jc} > 1 - \hat{\tau}_{jc}]$ (equivalently, $\hat{\tau}_{jc} > 0.5$). Because v is anchored by $Z_v = 1$, the nodes with $\hat{Z}_j = 1$ are precisely the estimated block containing v and induce the target-specific explanation G_S.

Proposition 1. *Under Assumptions 1 and 2, when the predictive likelihood associated with a candidate latent subgraph agrees with the output of the frozen GNN, $P(Y \mid A, \mathbf{Z}_S) = P_\Phi(Y \mid A, \mathbf{Z}_S)$, the estimation $\hat{\boldsymbol{\theta}}_c$ minimize $\mathrm{KL}(R(\mathbf{Z}_S \mid Y) \,\|\, P(\mathbf{Z}_S \mid A, Y))$ and $\hat{\mathbf{Z}}_S$ approached by HierExplainer maximize,*

$$Q_\Phi(\mathbf{z}) + log P(\mathbf{Z}_S \mid A). \tag{5}$$

The proof is provided in Appendix D.2. The condition in this proposition states how the Bayesian posterior is linked to the post-hoc GNN prediction whose explanation is sought. It is essential compatibility condition to connect with the fixed-GNN explanation objective. Proposition 1 separates the three roles played by HierExplainer. First, the local Bernoulli block model defines a structural posterior for the latent explanation indicator. Second, variational inference provides a tractable mean-field approximation to that posterior. Third, once the Bayesian predictive likelihood is identified with the frozen GNN, posterior MAP inference has the explicit combination form of a fixed-GNN explanation likelihood $Q_\Phi(\mathbf{z})$ and a local structural regularization $\log P(\mathbf{Z}_S \mid A)$. Accordingly, the general theoretical interpretation of HierExplainer is a structure-regularized version of the (1) explanation objective. The stronger statement that the recovered latent block directly maximizes (1), and hence the mutual-information objective represented by that surrogate, is made only under the additional condition in the corollary that the structural term does not alter the optimizer. This distinction keeps the Bayesian estimation argument and the post-hoc GNN explanation objective mathematically consistent.

4.2 HypoExplainer

HierExplainer estimates $\mathbf{Z}_S$ under Assumption 1, but a full generative model may be too restrictive. When the structure of $P(A, \mathbf{Z}_S)$ is not specified, we instead use edge-distribution tests to identify statistically distinctive candidate structures. The three variables are associated as follows. The explanatory subgraph G_S should preserve the trained GNN's prediction, whereas nodes and edges outside the explanation should have little effect on that prediction. We use the following assumption to link edge-distribution contrast to the GNN explanation objective.

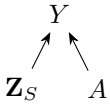


Figure 3: Relation among Y, Z, and A without a fully generative model between Z and A.

Assumption 3. *The graphical model in Figure 3 describes the relation among $\mathbf{Z}_S$, A, and Y.*

Assumption 4. *For ever label pair (k, l) among GNN prediction, edges among them are generated from Bernuolli distribution*

$$A_{ij} \mid y_i = k, y_j = \ell \sim \mathcal{B}(p_{k\ell}).$$

and for edges among background

$$A_{ij} \mid i, j \in \mathcal{N}_B \sim \mathcal{B}(p_{00}),$$

Remark: This assumption is the local independent-edge condition required by the ordinary two-sample proportion test (Agresti, 2013),(Dunn, 1961). The background nodes set $\mathcal{N}_B$ here means the nodes in it are not in the computation graphs of the evaluated target nodes sets; We introduce this Assumption based on stochastic graph generation mechansim and Bernuolli distribution is the most common one without extra information.

For an edge connecting nodes i and j, membership in G_S can be represented as $Z_i Z_j = 1$, and background membership as $Z_i Z_j = 0$. According to Assumption 3 and Assumption 4, define the edge probabilities inside and outside the explanatory subgraph as:

$$\begin{aligned} p_{00} &= P(A_{ij} = 1 \mid Y, Z_i Z_j = 0) \\ p_{kl} &= P(A_{ij} = 1 \mid Y_i = k, Y_j = l, Z_i Z_j = 1) \end{aligned}$$

With this assumption, we estimate $\mathbf{Z}_S$ by hypothesis testing and call the method HypoExplainer:
Initialize and localize: The supervised background nodes set $\mathcal{N}_B$ is given, other initialization are

the same as Hierexplainer.

Hypothesis test and statistic: Assumption 4 states that edges in G_S follow a distribution different from that of background edges. Thus p_{00} is a background parameter that does not vary with the GNN output y_o, while p_{kl} is the edge probability for predicted-label pair (k, l). We identify candidate explanatory label pairs with the following test:

$$H_0 : p_{kl} = p_{00} \quad \text{vs} \quad H_1 : p_{kl} \neq p_{00}. \tag{6}$$

Under H_0, edges between predicted-label pair (k, l) have the same distribution as background edges. When H_0 is rejected, the pair (k, l) is treated as a statistically distinctive candidate. We use a Z statistic for this two-sample test. Estimating $\hat{p}_{00}$ requires a supervised subset of background nodes $\mathcal{N}_B$. Define $n_{kl} = \#\{(i,j) \mid y_{oi} = k, y_{oi} = l\}$, $E_{kl} = \#\{(i,j) \mid A_{ij} = 1, y_{oi} = k, y_{oi} = l\}$, $n_{00} = \#\{(i,j) \mid i, j \in \mathcal{N}_B\}$, $E_{00} = \#\{(i,j) \mid A_{ij} = 1, i, j \in \mathcal{N}_B\}$, $\hat{p}_{kl} = E_{kl}/n_{kl}$, $\hat{p}_{00} = E_{00}/n_{00}$, $\hat{p}_{pool} = (E_{kl} + E_{00})/(n_{kl} + n_{00})$, then we introduce $Z-$statistic:

$$\mathcal{Z}_{stat,kl} = \frac{\hat{p}_{kl} - \hat{p}_{00}}{\sqrt{\hat{p}_{pool}(1 - \hat{p}_{pool})\left\{\frac{1}{n_{kl}} + \frac{1}{n_{00}}\right\}}}. \tag{7}$$

We apply the Bonferroni correction to the pairwise tests.

Recovery Explanation: After running the test for all $k, l \in [C]$, we build $\mathcal{R}_Z$, the set of candidate label pairs rejected by the test. By above constructon, $\mathcal{R}_Z$ contains GNN output-label pairs selected by their edge distributions. If GNN predictions and connection probabilities between classes are not independent, edge distributions can differ across predicted-label pairs. With definition of $\mathcal{C}_Z = \bigcup_{(k,\ell)\in\mathcal{R}_Z}\{k, \ell\}$ as union of labels that at least one of their pair are in $\mathcal{C}_Z$, we link these selected pairs to the node indicators: $\hat{Z}_i = \mathbf{1}(Y_i \in \mathcal{C}_Z)$, so nodes whose predicted labels belong to $\mathcal{C}_Z$ are included in G_S.

The following assumptions control the structural signal and its relationship to GNN faithfulness.

Assumption 5 (Minimum structural signal)**.** *For each true alternative $(k, \ell) \in \mathcal{R}^*$, $\mathcal{R}^*_Z = \{(k, \ell) : p_{k\ell} \neq p_{00}\}$, define*

$$\delta_{k\ell} = p_{k\ell} - p_{00}, \qquad \overline{p}_{k\ell} = \frac{n_{k\ell}p_{k\ell} + n_{00}p_{00}}{n_{k\ell} + n_{00}},$$

$$\sigma^2_{0,k\ell} = \overline{p}_{k\ell}(1 - \overline{p}_{k\ell})\left(\frac{1}{n_{k\ell}} + \frac{1}{n_{00}}\right),$$

and

$$\sigma^2_{1,k\ell} = \frac{p_{k\ell}(1 - p_{k\ell})}{n_{k\ell}} + \frac{p_{00}(1 - p_{00})}{n_{00}}.$$

for fixed signifigant level α and missing probability β, every true alternative satisfies

$$|\delta_{k\ell}| \geqslant z_{1-\alpha/(2m)}\,\sigma_{0,k\ell} + z_{1-\beta/m}\,\sigma_{1,k\ell} \tag{8}$$

where z_q is the qth standard-normal quantile.

Assumption 6. *The ground truth of nodes indicator of explanation subgraph is a maximizer of the objective in (1):*

$$\mathbf{Z}^\star_S \in \arg\max_{\mathbf{z}} Q_\Phi(\mathbf{z}). \tag{9}$$

Assumption 5 is a sufficient large-sample power condition. (Fleiss et al., 2003, Ch. 4). Since Hypoexplainer are based on Z-test, this assumption guarantees Type II error controllable and Assumption 6 is the explicit bridge from edge distribution recovery to GNN faithfulness.

Under these assumptions, Theorem 1 connects the latent node set selected by test (6) to the post-hoc mutual-information objective described in Section 3.

Theorem 1. *Under Assumptions 4–6, when GNN prediction labels C is fixed and $n_{kl} \sim n_{00} \to \infty$, for fixed signifigant level α and missing probability β, we have*

$$P\left(\widehat{\mathbf{Z}}_S \in \arg\max_{\mathbf{z}\in\mathfrak{Z}_\mathcal{N}} Q_\Phi(\mathbf{z})\right) \geqslant 1 - \alpha - \beta - o(1). \tag{10}$$

The proof is provided in Appendix D.3. This Theorem 1 is a high probability recovery theorem for the structurally distinctive label target. Under Assumptions 3–6, when the ground truth actually maximize the objective (1), by control Type I and Type II error of Z-test, Theorem 1 manage to recover the ground truth explanation subgraph in a rather high probability. The test therefore selects an explanatory subgraph that maximizes the same mutual-information objective used for post-hoc GNN explanation under the stated assumptions with a fixed controllable probability.

5 EXPERIMENTS

The main text retains three complementary evaluations: HierExplainer's synthetic structural recovery, HypoExplainer's synthetic motif recovery and faithfulness, and the PPI-Essential case study. Additional parameter diagnostics, resource measurements, and the Football structural sanity check appear after the bibliography.

5.1 HIEREXPLAINER: STRUCTURAL RECOVERY

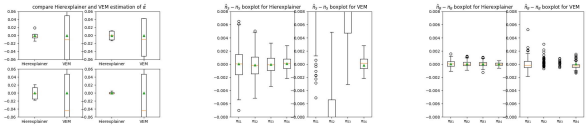


Figure 4: Estimation-bias distributions for $\boldsymbol{\alpha}$ (left) and π (right).

	RI	F_1
Variational EM	0.887	0.802
GCN	0.903	0.832
HierExplainer	0.912	0.844

Table 1: Synthetic structural recovery: Rand Index (RI) and F_1. Higher is better.

We generate a synthetic graph from a standard SBM with class probabilities $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.1, 0.2, 0.3, 0.4)$. We compare HierExplainer with the variational EM algorithm of Daudin et al. (2008), initialized with same value as HierExplainer. Over 1000 simulations, Figure 4 shows estimation-bias distributions and Table 2 reports results for $\pi_{S,c}$ and α_i. These results support HierExplainer's estimation have bias close to zero and lower variability than those of VEM and the latent-node inference component of the explanation framework.

We also evaluate recovery of G_S through node classification. The comparison includes the original trained GCN (Kipf & Welling, 2017) and variational EM, which recovers node groups without supervision. We use the Rand Index (RI) (Rand, 1971) and F_1 score (Taha & Hanbury, 2015). Table 1 shows that HierExplainer has higher RI and F_1 scores than both baselines, suggesting that the recovered latent block provides a reasonable explanatory structure for the GNN prediction.

	Method	α_1	α_2	α_3	α_4	$\pi_{S,1}$	$\pi_{S,2}$	$\pi_{S,3}$	$\pi_{S,4}$
	TRUE	0.100	0.200	0.300	0.400	0.030	0.080	0.080	0.080
mean	HierExplainer	0.101	0.200	0.302	0.397	0.030	0.080	0.080	0.080
	Variational EM	0.262	0.183	0.158	0.397	0.039	0.059	0.094	0.079
median	HierExplainer	0.101	0.200	0.300	0.397	0.030	0.080	0.080	0.080
	Variational EM	0.294	0.190	0.156	0.397	0.038	0.068	0.095	0.080
std	HierExplainer	0.010	0.012	0.015	0.015	0.003	0.002	0.001	0.001
	Variational EM	0.072	0.075	0.074	0.015	0.011	0.018	0.015	0.005

Table 2: Mean, median, and standard deviation of $\hat{\alpha}_i$, $i = 1, 2, 3, 4$, and the diagonal parameters $\pi_{S,c}$ over 1000 experiments with HierExplainer and variational EM.

5.2 HYPOEXPLAINER: MOTIF RECOVERY AND FIXED-GNN FAITHFULNESS

We generate BA-Shapes, BA-Community, Tree-Cycles, and Tree-Grid as in Ying et al. (2019). BA-Shapes and BA-Community contain house motifs; Tree-Cycles and Tree-Grid attach cycle and grid motifs to a tree base. Dataset descriptions and statistics are retained in Appendix B.2. The comparison includes Saliency, GNNExplainer, PGExplainer, and SubgraphX, and uses a pretrained GCN backbone. Table 9 in Appendix B.2 reports results for a GIN backbone.

Evaluation metrics. For quantitative evaluation, we treat ground-truth explanatory edge membership for edges mask methods or nodes membership for nodes scoring methods as a binary label and the importance weights produced by each explainer as prediction scores. We report AUC, F_1, and

BA-Shapes					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**1.000**	0.997	0.996	0.980	0.326
F_1	**0.950**	0.022	0.426	0.600	0.007
F_{NS}	**0.974**	0.868	0.421	0.263	0.433
BA-Community					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**0.998**	0.990	0.984	0.712	0.264
F_1	**0.939**	0.016	0.650	0.177	0.003
F_{NS}	**0.788**	0.571	0.537	0.088	0.533
Tree-Grid					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**1.000**	0.951	0.627	0.397	0.469
F_1	**1.000**	0.000	0.509	0.454	0.014
F_{NS}	**0.563**	0.000	0.000	0.000	0.000
Tree-Cycles					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	0.786	0.905	0.538	**0.944**	0.496
F_1	0.000	0.022	0.436	**0.595**	0.020
F_{NS}	0.485	0.000	**0.554**	0.384	0.093

Table 3: Quantitative comparison of HypoExplainer and baseline GNN explanation methods on synthetic datasets.

the following metrics. *Probability of sufficiency (PS):* The proportion of nodes for which the explanation graph produces the same prediction as the original graph, where G_S denotes the explanation. *Probability of necessity (PN):* The proportion of nodes for which removing the explanation changes the GNN prediction, where G_{-S} is the graph after removing the explanation. F_{NS} *score:* Higher PS and PN are both desirable. Their harmonic mean is $F_{NS} = \frac{2 \cdot PN \cdot PS}{PN+PS}$. *Sparsity:* The average proportion of edges removed from the original graph G_i to obtain explanation G_{iS}. The full metric formulas are retained in Appendix B.2.

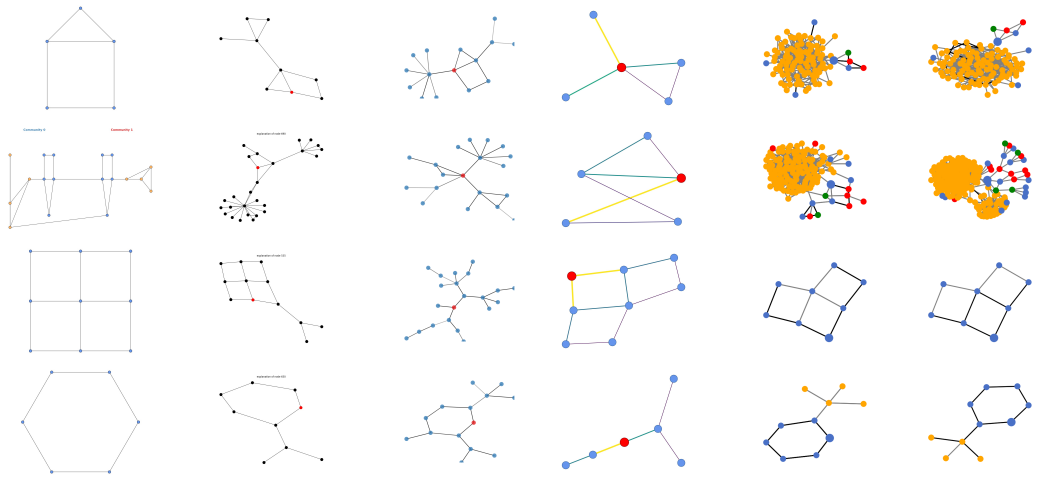
Motif Recovery: Figure 5 provides a qualitative comparison, and Table 3 reports quantitative results. Appendix B.2 reports results for a GIN backbone. On BA-Shapes, BA-Community, and Tree-Grid, HypoExplainer has the highest AUC and F_1 values in the table. It also has the highest F_{NS} on these three datasets, indicating a stronger balance between preserving the original prediction and changing it when the explanation is removed. Tree-Cycles is different: PGExplainer has the best AUC and F_1, GNNExplainer has the best F_{NS}, and HypoExplainer's F_1 is zero. The failure mode is discussed below. Figure 5 compares the topology of representative explanations. For BA-Shapes and BA-Community, the HypoExplainer examples more closely recover the house motifs, while the baseline examples omit motif structure or include less compact subgraphs. For Tree-Grid, HypoExplainer recovers the grid in the displayed example, whereas the baselines omit motif edges or add extra nodes and edges. The Tree-Cycles example should be read alongside the weaker aggregate results in Table 3; a representative visualization does not resolve the zero aggregate F_1.

We also report resource measurements to characterize the practical computational cost of the available implementations, not to establish a controlled cross-method efficiency ranking. The accounting associated with Table 7 separates shared checkpoint loading, method-specific setup, per-target inference, peak CPU memory, timeouts, OOMs, and successful-run denominators. Appendix B.2 reports CPU runtime and absolute process-RSS independently from effectiveness metrics.

Robustness and stability. Table 4 reports Jaccard stability (Jaccard, 1902) under perturbations of the computation neighborhood $\mathcal{N}$, in which eligible non-motif edges are added or removed. The reduced study used the pre-trained GCN, 12 outcome-blind targets per synthetic dataset, and three deterministic replicates for each addition or removal of 1, 2, or 4 eligible non-motif edges in the fixed three-hop context. Let R_t be the collection of perturbation operations for target t, with $r \in R_t$. Infeasible perturbations and perturbations that changed the predicted class were recorded but excluded from the stability mean. For original and perturbed selected edge sets E_t and $E'_{t,r}$,

Dataset	Hypo	PG	Sal.	GNNE
BA-Shapes	**0.995**	0.978	0.961	0.807
BA-Community	**0.996**	0.933	0.954	0.765
Tree-Grid	**0.848**	0.785	0.831	0.577
Tree-Cycles	0.636	**0.804**	0.742	0.615

Table 4: Mean edge Jaccard stability. Hypo: HypoExplainer; PG: PGExplainer; Sal.: Saliency; GNNE: GNNExplainer. Bold marks each row maximum.



(a) Ground Truth motifs. (b) HypoExplainer (c) GNNExplainer (d) Saliency (e) PGExplainer (f) SubgraphX

Figure 5: Representative motifs and explanations for the four synthetic datasets. The target node in each HypoExplainer panel is marked in red.

define Jaccard stability $\bar{J}$ by

$$J_{t,r} = \frac{|E_t \cap E'_{t,r}|}{|E_t \cup E'_{t,r}|}, \qquad \bar{J} = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{|R_t|} \sum_{r \in R_t} J_{t,r}.$$

Each eligible target therefore receives equal weight, regardless of its number of perturbation records. The metric measures stability of the selected edge set conditional on an unchanged prediction, and larger values indicate greater stability. HypoExplainer has the highest value on BA-Shapes, BA-Community, and Tree-Grid; PGExplainer has the highest value on Tree-Cycles.

Tree-Cycles limitation. A balanced tree with repeated six-node cycles also supplies relatively homogeneous predicted-label connectivity, so HypoExplainer's density-contrast screen can be weak or unstable even when the pre-trained GNN remains sensitive to the selected edge set. The Tree-Cycles result in Table 3, with AUC 0.786 and $F_1 = 0$, also reflects the fixed score threshold of 0.5 used for F_1. A ranking can have a useful AUC even when no positive score exceeds 0.5, or when thresholded positives do not overlap the true positives. Appendix B.2 gives the full diagnosis.

Reliability and limitations: HypoExplainer relies on contrasts between motif and background edge distributions, so its performance depends on both densities. Appendix B.3 reports pairwise classification accuracy, false-positive rate (FPR), and empirical familywise-error frequency ($\hat{e}_{FWER}$) (Fawcett, 2006; Holm, 1979; Benjamini & Hochberg, 1995) as motif counts and background-edge removal vary. The response in Table 10 depends sharply on the regime. For Tree-Cycles, F_1 is near one only at low motif fractions with no basis drop, but becomes zero for all motif fractions when 25% or 50% of basis edges are dropped and is already zero at 75–100% motifs with no drop. A balanced tree with repeated six-node cycles also supplies relatively homogeneous predicted-label connectivity, so HypoExplainer's density-contrast screen can be weak or unstable even when the pre-trained GNN remains sensitive to the selected edge set. The Tree-Cycles result in Table 7, with AUC 0.786 and $F_1 = 0$, also reflects the fixed score threshold of 0.5 used for F_1. A ranking can have a useful AUC even when no positive score exceeds 0.5, or when thresholded positives do not overlap the true positives. Appendix B.2 gives the full diagnosis.

5.3 PPI-Essential Real-Data Case Study

The Protein-Protein Interaction network is a public BioSnap dataset `https://snap.stanford.edu/biodata/index.html` containing experimentally documented physical interactions among human proteins, including metabolic enzyme-coupled and signaling interactions. Nodes represent proteins, and edges represent physical interactions in human cells. The graph has 21521 nodes and 342353 edges. Proteins are mapped to genes, and binary node labels indicate gene essentiality. Essential genes are required for an organism's survival and are studied in relation to

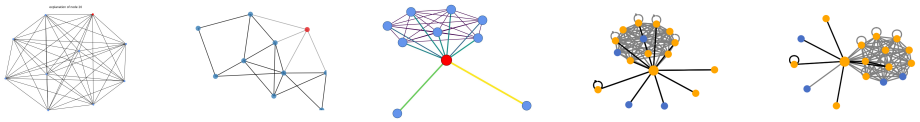


Figure 6: Subgraphs identified by HypoExplainer, GNNExplainer, Saliency, PGExplainer, and SubgraphX, from left to right.

biological-system resilience, minimal genomes or organisms, and therapeutic targets. The dataset was first analyzed in Agrawal et al. (2018).

Faithfulness and sparsity: PPI-Essential has no indexed explanatory-edge ground truth. Accordingly, Table 5 reports native faithfulness and sparsity. The full PS/PN breakdown and resource measurements appear in Appendix C.2. The qualitative example is discussed below. HypoExplainer therefore has the highest F_{NS} for both backbones in this table. Its lower sparsity reflects its retention of edges from the original input graph, which produces denser subgraphs than methods that select individual important edges.

Method	F_{NS}	Sparsity
GCN		
HypoExplainer	**0.669**	0.337
Saliency	0.204	**0.999964**
GNNExplainer	0.274	0.891
PGExplainer	0.000	0.9993
SubgraphX	0.032	0.9995
GIN		
HypoExplainer	**0.565**	0.416
Saliency	0.191	**0.999964**
GNNExplainer	0.261	0.885
PGExplainer	0.000	0.9992
SubgraphX	0.091	0.9994

Table 5: PPI-Essential faithfulness and sparsity for GCN and GIN.

Potential biology discovery, a K-10 clique: Figure 6 compares explanatory subgraphs for a PPI node corresponding to Entrez Gene ID 152485. The supervised dataset labels it as non-essential, while the trained GNN predicts it as essential. HypoExplainer identifies a dense, clique-like structure with 10 nodes, while the other displayed explanations omit edges or include extra nodes. This structure is a hypothesis-generating signal rather than biological validation. Prior work has examined cliques in PPI networks for disease-gene prediction (Yang et al., 2014) and essential-protein analysis (Lin et al., 2009). Clique-like modules can be biologically informative when supported by gene annotations, enrichment analysis, and external literature. Because PPI-Essential has no explanatory-topology ground truth for each node, its quantitative results are not directly comparable to the motif-recovery evaluation in Section 5.2. The PPI example is therefore an explanatory case study, not definitive biological evidence. The structure in Figure 6 offers one possible explanation for the trained GNN's essential prediction despite the node's non-essential label: its local neighborhood resembles dense essential or disease-associated PPI modules. Biological validation would require gene symbols for the selected nodes and GO, pathway, disease-gene, or essential-gene enrichment analysis.

Additional evidence and scope. The Football experiment evaluates structural recovery rather than target-level GNN faithfulness. Full results are in Appendix C. Resource measurements are in Appendix B.2; the timing results are descriptive because implementation scopes and setup conditions differ across methods.

6 Conclusion

We presented a statistical framework for post-hoc GNN explanation based on latent node indicators and explicit assumptions about edge distributions. The framework treats explanatory-subgraph discovery as either variational inference under an SBM-type model or hypothesis testing with local instance-level refinement. Experiments on synthetic and real-world graphs show where the methods recover explanatory subgraphs and where their assumptions limit performance; runtime and memory results are reported as descriptive implementation measurements rather than as a controlled efficiency ranking.

AI USE STATEMENT

We used generative AI tools to improve the clarity, grammar, and organization of the manuscript; draft portions of the paper; assist in generating synthetic datasets for the experiments; and develop mathematical arguments and proofs. The data-generation procedures, theoretical assumptions, and proofs are documented in the main text and appendix. The authors take full responsibility for the final submission, including the accuracy of the text and references, the validity of the datasets and experimental results, and the correctness of the mathematical claims.

REPRODUCIBILITY STATEMENT

We provide our code and data in the supplementary material to support reproduction of the reported experiments.

REFERENCES

Monica Agrawal, Marinka Zitnik, and Jure Leskovec. Large-scale analysis of disease pathways in the human interactome. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, volume 23, pp. 111, 2018.

Alan Agresti. *Categorical Data Analysis*. Wiley, Hoboken, NJ, 3 edition, 2013. ISBN 9780470463635.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

J-J Daudin, Franck Picard, and Stéphane Robin. A mixture model for random graphs. *Statistics and computing*, 18(2):173–183, 2008.

Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961. doi: 10.1080/01621459.1961.10482090.

Alexandre Duval and Fragkiskos D. Malliaros. GraphSVX: Shapley value explanations for graph neural networks. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2021.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Joseph L. Fleiss, Bruce Levin, and Myunghee Cho Paik. *Statistical Methods for Rates and Proportions*. John Wiley & Sons, Hoboken, NJ, 3 edition, 2003. ISBN 9780471526292. doi: 10.1002/0471445428.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Paul Jaccard. Distribution comparée de la flore alpine dans quelques régions des alpes occidentales et orientales. *Bulletin de la Murithienne*, 31:81–92, 1902.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Chen-Ching Lin, Hsueh-Fen Juan, Jen-Tsung Hsiang, Yih-Chii Hwang, Hirotada Mori, and Hsuan-Cheng Huang. Essential core of protein- protein interaction network in escherichia coli. *Journal of proteome research*, 8(4):1925–1931, 2009.

Ana Lucic, Maartje 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 (AISTATS)*, 2022.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971. doi: 10.1080/01621459.1971.10482356. URL `https://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356`.

Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, 15:1–28, 2015.

Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, Cambridge, 1998.

Minh N. Vu and My T. Thai. PGM-Explainer: Probabilistic graphical model explanations for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Lei Yang, Xudong Zhao, and Xianglong Tang. Predicting disease-related proteins based on clique backbone in protein-protein interaction network. *International journal of biological sciences*, 10 (7):677, 2014.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pp. 9240–9251, 2019.

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 (ICML)*, volume 139, pp. 12241–12252, 2021.

APPENDIX A SUPPLEMENTARY METHOD DETAILS

A.1 TWO-SAMPLE Z TEST

Two-sample Z test: Following Chapter 2 of Agresti (2013), let $X_{1,1}, X_{1,2}, \ldots, X_{1,n_1} \sim \mathcal{B}(p_1)$ and $X_{2,1}, X_{2,2}, \ldots, X_{2,n_2} \sim \mathcal{B}(p_2)$ be two independent samples, where $X_{g,i}$ follows a Bernoulli distribution $\mathcal{B}(p_g)$ for $g \in \{1, 2\}$. Estimate p_g by $\hat{p}_g = S_g/n_g$, $S_g = \sum_{i=1}^{n_g} X_{g,i}$. To test whether the two samples are from the same distribution, the hypotheses and Z statistic are:

$$H_0 : \; p_1 = p_2 \quad \text{vs.} \quad H_1 : \; p_1 \neq p_2$$

and

$$Z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\hat{p}(1-\hat{p})\left(\dfrac{1}{n_1} + \dfrac{1}{n_2}\right)}} \tag{11}$$

with $\hat{p} = (S_1 + S_2)/(n_1 + n_2)$. For significance level α, let $z_{1-\alpha/2}$ denote the standard normal quantile. Reject H_0 if and only if $|Z| > z_{1-\alpha/2}$, and the two-sided p-value is $p_{\text{value}} = 2\big(1-\Phi(|Z|)\big)$, where Φ denotes the standard normal cumulative distribution function.

A.2 HIEREXPLAINER LIKELIHOOD AND ALGORITHM

$$\begin{aligned} \log P(A, \mathbf{Z}_S \mid y_i = c) \quad = \quad & \sum_{j \in \mathcal{N}} \{ Z_j \log \alpha_c + (1 - Z_j) \log(1 - \alpha_c) \\ & + \frac{1}{2} \sum_{i \neq j} Z_j Z_i \log \mathcal{B}(A_{ij}, \pi_{S,c}) + (1 - Z_j Z_i) \log \mathcal{B}(A_{ij}, \pi_{B,c}) \Big\} \end{aligned} \tag{12}$$

APPENDIX B SYNTHETIC DATASET ANALYSIS

The evaluation addresses structural and motif recovery, fixed-GNN faithfulness under keep and remove masks, stability under perturbations that preserve the prediction, and resource cost.

B.1 DATA GENERATION IN SECTION 5.1

We generate a synthetic graph from a standard SBM with class probabilities $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (0.1, 0.2, 0.3, 0.4)$. The transition matrix is

$$\pi = \begin{bmatrix} 0.03 & 0.002 & 0.003 & 0.001 \\ 0.002 & 0.08 & 0.0015 & 0.002 \\ 0.003 & 0.0015 & 0.08 & 0.001 \\ 0.001 & 0.002 & 0.001 & 0.08 \end{bmatrix}.$$

Under the model in (2), the corresponding parameters are $(\pi_{S,1}, \pi_{B,1}) = (0.03, 0.006)$, $(\pi_{S,2}, \pi_{B,2}) = (0.08, 0.0055)$, $(\pi_{S,3}, \pi_{B,3}) = (0.08, 0.0055)$, $(\pi_{S,4}, \pi_{B,4}) = (0.08, 0.004)$. Because G_S is the object of interest, we evaluate estimation of $(\pi_{S,1}, \pi_{S,2}, \pi_{S,3}, \pi_{S,4})$.

B.2 COMPLETE RESULTS IN SECTION 5.2

We generate BA-Shapes, BA-Community, Tree-Cycles, and Tree-Grid as in Ying et al. (2019). Table 6 summarizes the datasets. BA-Shapes consists of a 300-node Barabasi-Albert (BA) base graph with 80 house motifs attached to randomly selected base nodes, followed by random edge perturbations. It has no node features. Base-graph nodes have label 0, while nodes at the top, middle, and bottom of each house have labels 1, 2, and 3, respectively. BA-Community combines two BA-Shapes graphs. Its node features are sampled from two Gaussian distributions, one for each component, and its labels encode both structural role and community membership, giving 8 classes. Tree-Cycles uses an 8-level balanced binary tree as the base graph and attaches 80 six-node cycle motifs to randomly selected base nodes. Tree-Grid follows the same construction but replaces each cycle with a 3-by-3 grid motif.

	BA-Shapes	BA-Community	Tree-Cycles	Tree-Grid
#nodes	700	1,400	871	1,231
#edges	4,110	8,920	1,950	3,410
#labels	4	8	2	2

Table 6: Summary of the synthetic datasets.

Figure 5 provides a qualitative comparison, and Table 7 reports quantitative results. The baselines include GNNExplainer Ying et al. (2019), PGExplainer Luo et al. (2020), and SubgraphX Yuan et al. (2021). Each generates factual, post-hoc subgraph explanations through a different mechanism. GNNExplainer and PGExplainer optimize mutual-information objectives to obtain edge weights, while SubgraphX uses a Shapley-inspired search for influential subgraphs.

For quantitative evaluation, we treat ground-truth explanatory edge membership for edges mask methods or nodes membership for nodes scoring methods as a binary label and the importance weights produced by each explainer as prediction scores. We report AUC, F_1, and the following metrics.

Probability of sufficiency (PS): The proportion of nodes for which the explanation graph produces the same prediction as the original graph,

$$\frac{1}{N}\mathbf{1}\left\{\underset{c\in[C]}{\operatorname{argmax}} P_\Phi(Y=c \mid G) = \underset{c\in[C]}{\operatorname{argmax}} P_\Phi(Y=c \mid G_S)\right\}$$

where G_S denotes the explanation.

Probability of necessity (PN): The proportion of nodes for which removing the explanation changes the GNN prediction,

$$\frac{1}{N}\mathbf{1}\left\{\underset{c\in[C]}{\operatorname{argmax}} P_\Phi(Y=c \mid G) \neq \underset{c\in[C]}{\operatorname{argmax}} P_\Phi(Y=c \mid G_{-S})\right\}$$

where G_{-S} is the graph after removing the explanation.

F_{NS} **score:** Higher PS and PN are both desirable. Their harmonic mean is $F_{NS} = \frac{2 \cdot PN \cdot PS}{PN+PS}$.

Sparsity: The average proportion of edges removed from the original graph G_i to obtain explanation G_{iS},

$$1 - \frac{1}{n}\sum_{i=1}^{n}\frac{|G_{iS}|}{|G_i|}.$$

Table 7 reports these metrics for a pretrained GCN backbone. It reports results for a GIN backbone. On BA-Shapes, BA-Community, and Tree-Grid, HypoExplainer has the highest AUC and F_1 values in the table. It also has the highest F_{NS} on these three datasets, indicating a stronger balance between preserving the original prediction and changing it when the explanation is removed. HypoExplainer remains sparse on BA-Shapes and BA-Community, although SubgraphX has the highest sparsity. Tree-Cycles is different: PGExplainer has the best AUC and F_1, GNNExplainer has the best F_{NS}, and HypoExplainer's F_1 is zero. Section B.3 examines this failure mode.

BA-Shapes					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**1.000**	0.997	0.996	0.980	0.326
F_1	**0.950**	0.022	0.426	0.600	0.007
PS	**0.950**	0.767	0.267	0.217	0.433
PN	**1.000**	**1.000**	**1.000**	0.333	0.433
F_{NS}	**0.974**	0.868	0.421	0.263	0.433
Sparsity	0.982	0.931	0.954	0.927	**0.988**
BA-Community					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**0.998**	0.990	0.984	0.712	0.264
F_1	**0.939**	0.016	0.650	0.177	0.003
PS	**0.650**	0.400	0.367	0.050	0.533
PN	**1.000**	**1.000**	**1.000**	0.350	0.533
F_{NS}	**0.788**	0.571	0.537	0.088	0.533
Sparsity	0.978	0.956	0.955	0.947	**0.995**
Tree-Grid					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**1.000**	0.951	0.627	0.397	0.469
F_1	**1.000**	0.000	0.509	0.454	0.014
PS	0.400	**1.000**	**1.000**	**1.000**	**1.000**
PN	**0.950**	0.000	0.000	0.000	0.000
F_{NS}	**0.563**	0.000	0.000	0.000	0.000
Sparsity	0.678	0.148	0.202	0.515	**0.993**
Tree-Cycles					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	0.786	0.905	0.538	**0.944**	0.496
F_1	0.000	0.022	0.436	**0.595**	0.020
PS	0.717	**1.000**	**1.000**	0.683	0.650
PN	0.367	0.000	**0.383**	0.267	0.050
F_{NS}	0.485	0.000	**0.554**	0.384	0.093
Sparsity	0.501	0.002	0.005	0.469	**0.993**

Table 7: Quantitative comparison of HypoExplainer and baseline GNN explanation methods on synthetic datasets.

The representative motif explanations and their interpretation are now presented in Figure 5 in the main text.

F_1 **Diagnosis** In experiments on Hypoexplainer, for the GCN backbone, the Tree-Cycles AUC of 0.786 and $F_1 = 0$ come from the same pooled continuous latent node-membership scores $s_{v,u}$. AUC measures their ranking, while F_1 uses hard predictions from the fixed threshold

$$\widehat{y}_{v,u} = \mathbf{1}[s_{v,u} > 0.5], \qquad F_1 = \frac{2TP}{2TP + FP + FN}.$$

$F_1 = 0$ whenever $TP = 0$: either no score crosses 0.5, or every score above the threshold belongs to a non-cycle node. AUC is

$$\mathrm{AUC} = \Pr(s^+ > s^-) + \tfrac{1}{2}\Pr(s^+ = s^-),$$

and can remain above chance when cycle nodes tend to rank above other nodes even if every score falls below 0.5. The implementation skipped cases with one predicted class or an empty selected label set and did not retain their identities or per-target scores. The available records cannot separate threshold calibration from a selected-class mismatch as the source of the zero true-positive counts. This calculation evaluates node membership; the cycle-edge evaluation arrays were disabled.

Efficiency and scaling: Table 8 reports resource use separately from recovery and faithfulness. For successful target runtimes $t_1, \ldots, t_m$, we report

$$\mathrm{median}(t), \qquad \mathrm{IQR}(t) = Q_{0.75}(t) - Q_{0.25}(t).$$

We apply the same median/IQR definition to peak process resident-set size (RSS). Runtime begins after each method's setup step, whose scope varies across implementations. All measurements use CPU. RSS records total process memory, including memory used outside the explainer itself. A missing measurement appears as "–". Comparisons are descriptive because methods differ in their timed regions, setup, warm-up records, target filters, and sometimes target identities.

We report resource measurements to characterize the practical computational cost of the available implementations, not to establish a controlled cross-method efficiency ranking. The accounting associated with Table 7 separates shared checkpoint loading, method-specific setup, per-target inference, peak CPU memory, timeouts, OOMs, and successful-run denominators. Table 7 reports CPU runtime and absolute process-RSS independently from effectiveness metrics.

These measurements should be interpreted descriptively because the historical implementations do not use identical timed regions, setup boundaries, warm-up procedures, target filters, or, in every case, exactly the same target identities. In the synthetic experiments, HypoExplainer is substantially faster per target than GNNExplainer in the recorded runs, but Saliency and PGExplainer have lower recorded per-target runtimes; SubgraphX varies considerably by dataset. Accordingly, we do not claim uniform runtime or memory superiority. The efficiency result is instead that the proposed statistical procedures have practical target-level costs under the reported implementations, with a controlled benchmark left for future work.

Dataset	Method	Runtime (s), median [IQR]	Peak RSS (MiB), median [IQR]
BA-Shapes	HypoExplainer	0.0417 [0.0014]	558.77 [85.91]
	Saliency	**0.001066 [0.000305]**	**350.02 [1.51]**
	GNNExplainer	2.2313 [0.0491]	1175.94 [77.84]
	PGExplainer	0.0040 [0.0006]	467.45 [17.58]
	SubgraphX	16.5434 [15.1114]	373.16 [18.21]
BA-Community	HypoExplainer	0.0586 [0.0033]	811.49 [77.30]
	Saliency	**0.001555 [0.000842]**	**445.12 [1.93]**
	GNNExplainer	2.2794 [0.2908]	1318.38 [265.55]
	PGExplainer	0.0045 [0.0010]	542.24 [19.71]
	SubgraphX	6.3623 [21.3530]	517.67 [42.24]
Tree-Grid	HypoExplainer	0.0493 [0.0031]	637.80 [64.95]
	Saliency	**0.000616 [0.000229]**	391.18 [0.77]
	GNNExplainer	2.1033 [0.0901]	1131.15 [74.16]
	PGExplainer	0.0026 [0.0005]	477.95 [19.14]
	SubgraphX	0.0329 [0.0760]	**298.54 [29.92]**
Tree-Cycles	HypoExplainer	0.0459 [0.0054]	589.95 [35.20]
	Saliency	**0.000633 [0.000155]**	**373.61 [0.14]**
	GNNExplainer	2.0385 [0.0729]	1095.67 [31.88]
	PGExplainer	0.0025 [0.0002]	463.27 [12.80]
	SubgraphX	0.0093 [0.0071]	381.98 [22.61]

Table 8: CPU target-level runtime and process resident-set size (RSS) for five explainers on four synthetic datasets. Entries are medians with interquartile ranges in brackets, and bold marks the smallest value within each dataset. Differences in setup boundaries and target cohorts limit the table to a descriptive comparison.

HypoExplainer's median target-level runtime ranges from 0.0417 to 0.0586 seconds, and its peak RSS ranges from 558.77 to 811.49 MiB. Its runtime is lower than GNNExplainer's on all four datasets; Saliency and PGExplainer are faster. These comparisons reflect the timing boundaries and target cohorts used by each method.

GIN backbone: Table 9 gives the synthetic results for a GIN backbone. Each method comes from a separate execution, the target cohorts differ, and F_1 follows each method's native definition. HypoExplainer retained 24 of 60 Tree-Grid targets and 22 of 60 Tree-Cycles targets for continuous-score recovery and resource summaries. Its PS and PN arrays use the full 60-target denominator. The scope of "Total time" also differs across methods, so the target-runtime row gives a more direct account of per-target cost. Efficiency comparisons remain descriptive.

HypoExplainer has the largest displayed AUC and F_1 on BA-Shapes, BA-Community, and Tree-Grid. It also has the largest F_{NS} on BA-Shapes and Tree-Grid; Saliency leads on BA-Community and Tree-Cycles. On Tree-Cycles, HypoExplainer has an AUC of 0.731 and a thresholded F_1 of zero. The scores retain some ranking information, yet the fixed threshold recovers no true positives. Its lower sparsity on Tree-Grid and Tree-Cycles shows a separate tradeoff in explanation size.

B.3 Reliability and limitations

B.3.1 Generated data and trained models

The design crosses four datasets with motif fractions of 15%, 50%, 75%, and 100% and background-edge drops of 0%, 25%, and 50%. This gives 48 cells, each with one deterministic graph and one three-layer GCN. BA-Shapes and Tree-Grid use 12, 40, 60, and 80 motifs; BA-Community uses the same counts per component; Tree-Cycles uses 9, 30, 45, and 60. Counts are rounded down within each basis component. Removal applies only to edges with basis provenance. Motif, attachment, original 1% perturbation, cycle-extra, and BA-Community join edges remain in the graph. For Tree-Cycles, removal applies to balanced-tree basis edges instead of BA edges.

BA-Shapes					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**1.000**	0.996	0.996	0.998	0.326
F_1	**0.952**	0.027	0.803	0.832	0.011
PS	**0.950**	0.933	0.700	0.867	0.433
PN	**1.000**	0.933	**1.000**	**1.000**	0.433
F_{NS}	**0.974**	0.933	0.824	0.929	0.433
Sparsity	0.982	0.944	0.941	0.943	**0.988**
Total time (s)	3.003	0.330	137.460	**0.252**	998.528
Runtime (s), median [IQR]	0.0429 [0.0047]	0.0050 [0.0013]	2.2314 [0.0503]	**0.0040 [0.0007]**	15.3295 [13.4920]
Peak RSS (MiB), median [IQR]	607.36 [42.50]	**372.67 [12.83]**	1174.34 [96.77]	437.38 [16.41]	383.26 [10.89]
BA-Community					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**0.998**	0.988	0.977	0.947	0.272
F_1	**0.943**	0.052	0.618	0.265	0.002
PS	0.750	**0.967**	0.383	0.100	0.433
PN	**1.000**	0.983	**1.000**	0.550	0.500
F_{NS}	0.857	**0.975**	0.554	0.169	0.464
Sparsity	0.975	0.959	0.955	0.947	**0.994**
Total time (s)	3.948	0.450	149.825	**0.280**	406.995
Runtime (s), median [IQR]	0.0582 [0.0049]	0.0057 [0.0022]	2.3072 [0.2956]	**0.0044 [0.0011]**	0.2335 [0.1822]
Peak RSS (MiB), median [IQR]	828.38 [98.59]	**455.27 [0.30]**	1231.02 [36.93]	553.21 [20.16]	499.86 [14.32]
Tree-Grid					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	**1.000**	0.935	0.607	0.574	0.464
F_1	**1.000**	0.000	0.504	0.441	0.012
PS	0.400	**1.000**	**1.000**	**1.000**	**1.000**
PN	**0.833**	0.000	0.000	0.000	0.000
F_{NS}	**0.541**	0.000	0.000	0.000	0.000
Sparsity	0.678	0.148	0.215	0.522	**0.993**
Total time (s)	1.820	0.190	131.177	**0.172**	168.414
Runtime (s), median [IQR]	0.0484 [0.0039]	0.0028 [0.0007]	2.1123 [0.0607]	**0.0027 [0.0003]**	0.1371 [0.2287]
Peak RSS (MiB), median [IQR]	694.05 [97.01]	479.64 [0.11]	1140.08 [76.20]	478.26 [18.50]	**433.55 [36.73]**
Tree-Cycles					
Metric	HypoExplainer	Saliency	GNNExplainer	PGExplainer	SubgraphX
AUC	0.731	0.492	0.543	**0.926**	0.493
F_1	0.000	0.000	0.457	**0.591**	0.015
PS	0.683	**1.000**	0.983	0.900	0.633
PN	0.650	**1.000**	0.650	0.617	0.650
F_{NS}	0.666	**1.000**	0.783	0.732	0.642
Sparsity	0.509	0.001	0.005	0.455	**0.993**
Total time (s)	1.602	**0.160**	124.457	0.167	5.613
Runtime (s), median [IQR]	0.0482 [0.0031]	**0.0025 [0.0004]**	2.0225 [0.0759]	0.0026 [0.0004]	0.0355 [0.1074]
Peak RSS (MiB), median [IQR]	618.63 [50.47]	514.62 [6.20]	1219.92 [66.53]	466.53 [8.95]	**390.00 [24.51]**

Table 9: Synthetic explanation quality, faithfulness, sparsity, and computational cost with a GIN backbone. Bold marks the largest quality or sparsity value and the smallest time or RSS value within each dataset.

Each cell uses a recorded graph, split, and training seed, with the checkpoint taken after 1,000 epochs. The classifier is a three-layer concat GCN with hidden and output width 20. It uses Adam with a learning rate of 0.001, weight decay of 0.005, gradient clipping at 2, and an 80/20 random node split. Features are constant 10-dimensional vectors except for the Gaussian BA-Community features. All 48 training runs completed. A cell contains only one graph and checkpoint, so cross-cell differences have no sampling interval or causal interpretation.

B.3.2 Pair-decision and target-level error metrics

For target v and tested pair $h = (k, \ell)$, For target v and tested predicted-label pair $h = (k, \ell)$, let $T_{vh} = \mathbf{1}[h \in \mathcal{R}_v^\star]$, $\mathcal{R}_v^\star = \left\{h : p_{k\ell}^{(v)} \neq p_{00}\right\}$ denote ground truth of explanation subgraph label pairs and let $D_{vh} = \mathbf{1}[h \in \widehat{\mathcal{R}}_v]$ denote label pairs rejected H_0 in (6). Over all tested target–pair cases $\mathcal{H}$, the confusion counts are

$$TP = \sum_{(v,h)\in\mathcal{H}} T_{vh}D_{vh}, \qquad TN = \sum_{(v,h)\in\mathcal{H}} (1 - T_{vh})(1 - D_{vh}),$$
$$FP = \sum_{(v,h)\in\mathcal{H}} (1 - T_{vh})D_{vh}, \quad FN = \sum_{(v,h)\in\mathcal{H}} T_{vh}(1 - D_{vh}).$$

Thus

$$\text{Accuracy}_{pair} = \frac{TP + TN}{TP + TN + FP + FN}, \qquad \text{FPR}_{pair} = \frac{FP}{FP + TN}.$$

These pairwise quantities follow the standard confusion-matrix definitions of classification accuracy and false-positive rate (Fawcett, 2006). FPR is unavailable when $FP + TN = 0$, because no oracle-negative case was tested. For each target define

$$V_v = |\widehat{\mathcal{R}}_v \backslash \mathcal{R}^\star|, \quad R_v = |\widehat{\mathcal{R}}_v|, \quad I_v^{\text{FWER}} = \mathbf{1}[V_v \geqslant 1], \quad \text{FDP}_v = \frac{V_v}{\max(R_v, 1)}.$$

Following the conventional familywise-error and false-discovery-rate definitions (Holm, 1979; Benjamini & Hochberg, 1995), the corresponding repeated-sampling quantities are

$$\text{FWER} = \Pr(V_v \geqslant 1), \qquad \text{FDR} = \mathbb{E}\left[\frac{V_v}{\max(R_v, 1)}\right].$$

Each grid cell contains one graph and a fixed cohort of $M = 60$ targets. We report the descriptive summaries

$$\widehat{e}_{\text{FWER}} = \frac{1}{M}\sum_{v=1}^{M} I_v^{\text{FWER}}, \qquad \widehat{e}_{\text{FDR}} = \frac{1}{M}\sum_{v=1}^{M} \text{FDP}_v, \qquad M = 60.$$

The fixed oracle is a declared evaluation rule rather than an estimate of latent population edge probabilities. Target neighborhoods overlap, so $\widehat{e}_{\text{FWER}}$ and $\widehat{e}_{\text{FDR}}$ summarize the observed target-level violation frequency and mean FDP. They do not estimate population FWER or FDR under independent target sampling. Smaller values indicate fewer false discoveries, although a procedure that makes no discoveries also produces zero; pairwise accuracy and recovery are needed to interpret that case. Tests use Bonferroni correction above. Under valid individual p-values, its nominal level controls familywise error within each target's local contrast family; it does not require the pairwise FPR or the descriptive, fixed-oracle $\widehat{e}_{\text{FWER}}$ to equal 0.05. At the 15% Tree-Cycles setting, the graph contains only 54 motif nodes, so the fixed 60-target cohort is completed with the six lowest-index background nodes.

Metric	Background drop (%)	BA-shape				BA-community				Tree-Grid				Tree-Cycle			
		15%	50%	75%	100%	15%	50%	75%	100%	15%	50%	75%	100%	15%	50%	75%	100%
Pairwise accuracy	0	0.419	0.383	0.376	0.314	0.252	0.213	0.223	0.197	0.386	0.404	0.330	0.366	0.312	0.300	0.342	0.395
	25	0.431	0.444	0.347	0.381	0.274	0.216	0.221	0.225	0.290	0.390	0.404	0.386	0.408	0.434	0.337	0.380
	50	0.437	0.417	**0.454**	0.390	**0.285**	0.219	0.232	0.251	0.337	0.352	0.383	**0.415**	0.411	0.337	**0.436**	0.395
FPR	0	0.183	0.223	0.228	0.348	0.525	0.403	0.401	0.364	0.568	0.596	0.568	**0.512**	0.393	0.418	0.333	0.281
	25	0.086	0.120	0.266	0.200	0.477	0.274	0.264	0.259	0.680	**0.512**	0.638	0.568	0.183	0.132	0.327	0.241
	50	**0.006**	0.093	0.057	0.123	0.199	**0.146**	0.178	0.156	0.571	0.545	0.532	**0.512**	0.179	0.326	**0.128**	0.209
$\widehat{e}_{\mathrm{FWER}}$	0	0.317	0.350	0.333	0.533	0.900	0.733	0.767	0.667	0.417	0.467	0.417	**0.350**	0.367	0.383	0.317	0.267
	25	0.167	0.200	0.450	0.267	0.867	0.750	0.717	0.617	0.567	**0.350**	0.500	0.417	0.183	0.117	0.283	0.217
	50	**0.017**	0.233	0.150	0.233	0.733	0.533	0.583	**0.483**	0.467	0.400	0.417	**0.350**	0.167	0.233	**0.100**	0.150
$\widehat{e}_{\mathrm{FDR}}$	0	0.175	0.249	0.211	0.449	0.451	0.407	0.425	0.450	0.292	0.308	0.333	0.267	0.367	0.375	0.317	0.233
	25	0.079	0.110	0.345	0.218	0.471	0.482	0.447	0.365	0.458	0.250	0.325	0.292	0.183	0.117	0.283	0.217
	50	**0.004**	0.143	0.057	0.172	0.370	0.364	0.360	**0.304**	0.367	0.308	0.300	**0.233**	0.167	0.233	**0.100**	0.150

Table 10: Pairwise decision and target-level error metrics across synthetic motif fractions and background-edge-drop settings. Rows give the three background-drop levels, while dataset column blocks give the four motif fractions. Bold marks the best value for each metric within a dataset across all 12 settings: accuracy is maximized, whereas FPR, $\widehat{e}_{\mathrm{FWER}}$, and $\widehat{e}_{\mathrm{FDR}}$ are minimized. All exact ties are bolded.

For this table, the large values measure disagreement between the realized pair decisions and the fixed oracle $\mathcal{R}^\star$. Normally large value of $\widehat{e}_{\mathrm{FWER}}$ and $\widehat{e}_{\mathrm{FDR}}$ represents a badly calibrated multiple-testing procedure but that does not represent Hypoexplainer is unreliable. These values can be large for several reasons: First of all, the $\widehat{e}_{\mathrm{FWER}}$, $\widehat{e}_{\mathrm{FDR}}$, and FPR in Table 10 do not directly estimate the theoretical FWER or FDR of a multiple-testing procedure. Because the 60 target neighborhoods overlap, their outcomes are dependent observations from one graph. Also, candidate edges within a neighborhood share nodes and may exhibit degree heterogeneity, clustering, motif dependence, or other correlations. Treating them as independent Bernoulli observations can underestimate the variance of $\widehat{p}_{k\ell}$, which inflates the test statistics and reduces the raw p-values. Third, the synthetic datasets are generated as description in subsection B.2, these motifs are fixed rather than generated from a simple Bernuolli in Assumption 4. That caused model misspecified and then generate the rejections. For settings of four synthetic datasets, Z-test may not the most effective but we still apply it because its simple for computation and valid for theoretical results. Since we construct estimated label set based on union of all label pairs, construction of the explanation subgraph from the label set can tolerate high false discover rate and recover the ground truth subgraph when the model mispecfication not extremely serious.

B.3.3 Explanation behavior across the same grid

Table 11 summarizes HypoExplainer's explanations. All metric definition are the same as those definied in Section B.2. HypoExplainer has high AUC and thresholded F_1 in most BA-Shapes, BA-Community, and Tree-Grid conditions. Tree-Cycles varies more: AUC ranges from 0.391 to 1.000, and F_1 is zero whenever background edges are removed, as well as for the two largest motif counts without removal. PN generally exceeds PS. Removing the explanation therefore changes the prediction more often than the selected subgraph alone preserves it. Sparsity also varies with graph structure and background density.

		BA-shape (M)				BA-community (M/community)				Tree-Grid (M)				Tree-Cycle (M)			
Metric	Background drop (%)	12	40	60	80	12	40	60	80	12	40	60	80	9	30	45	60
AUC	0	.999	.998	**1.000**	.999	.996	**.998**	.998	.997	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	.842	.909
	25	.999	.999	.995	.999	.984	.993	.986	.991	**1.000**	**1.000**	**1.000**	**1.000**	.477	.542	.391	.480
	50	.998	.982	.994	.991	.972	.959	.953	.954	**1.000**	**1.000**	**1.000**	**1.000**	.785	.699	.672	.716
F_1	0	.928	.889	**.960**	.920	.848	**.925**	.917	.919	**1.000**	**1.000**	**1.000**	.986	**1.000**	.993	.000	.000
	25	.952	.914	.877	.938	.787	.849	.832	.875	.993	.981	**1.000**	.992	.000	.000	.000	.000
	50	.925	.731	.871	.853	.567	.656	.695	.633	.998	.996	.988	.941	.000	.000	.000	.000
PS	0	.433	.483	.500	**.583**	**.683**	.383	.633	.450	.417	.467	.417	.350	.400	**.417**	.400	.317
	25	.367	.483	.433	**.583**	.533	.333	.333	.367	**.567**	.350	.500	.417	.400	.150	.333	.267
	50	.283	.267	.333	.383	.250	.167	.267	.333	.450	.400	.417	.350	.250	.250	.100	.150
PN	0	.800	.817	.833	.833	.950	**.983**	.950	**.983**	.933	.917	.950	**1.000**	.700	.767	.733	.817
	25	.883	.800	.900	.833	.917	.933	.933	.967	.750	.983	.900	.950	.600	.967	.950	.950
	50	.967	.933	**1.000**	.983	.867	.950	.933	.933	.867	.833	.950	.983	.917	.983	**1.000**	**1.000**
F_{NS}	0	.562	.607	.625	**.686**	**.795**	.552	.760	.617	.576	.618	.579	.519	.509	**.540**	.518	.456
	25	.518	.603	.585	**.686**	.674	.491	.491	.532	**.646**	.516	.643	.579	.480	.260	.494	.416
	50	.438	.415	.500	.552	.388	.284	.415	.491	.592	.541	.579	.516	.393	.399	.182	.261
Sparsity	0	.703	.757	.748	.822	**.971**	.968	.914	.920	.743	.804	.817	**.883**	.508	.527	.541	.564
	25	.743	.724	**.827**	.773	.967	.928	.922	.930	.622	.753	.748	.799	**.610**	.445	.415	.449
	50	.655	.815	.752	.759	.957	.933	.910	.874	.588	.646	.646	.716	.300	.237	.294	.263

Table 11: HypoExplainer explanation quality, faithfulness, and sparsity across motif counts and background-edge removal. Rows give the three background-drop levels, and column blocks give motif counts for each dataset. Values are shown to three decimal places. Bold marks the maximum in each dataset's 3×4 block. For sparsity, the maximum is the smallest explanation by edge count, which alone does not establish explanation quality. Extrema are computed before rounding, and ties are bolded.

APPENDIX C REAL DATA ANALYSIS

We evaluate the methods on two real networks with different structures and with different validation goals. The American College Football network has known community labels and is used as a structural sanity check for HierExplainer's two-block latent recovery. The PPI-Essential network is much larger and is used as a case study for HypoExplainer without explanatory-edge ground truth. The Football experiment therefore evaluates whether the assumed latent structural model can recover known organization; it is not, by itself, evidence that HierExplainer is a faithful post-hoc explanation of every GNN decision.

C.1 AMERICAN COLLEGE FOOTBALL NETWORK

The American College Football network represents games played among U.S. Division I college football teams during one season. Each node is a team, and an edge connects two teams that played each other. The graph contains 115 nodes in 12 known communities and 613 edges. Because the community assignments are known, they provide an external reference for evaluating structural block recovery.

For a target team v, HierExplainer treats the target's community as the explanatory block in a one-versus-background local two-block model and estimates the corresponding latent membership from the graph structure, initialized by the trained GNN output. Table 12 compares the reference class proportions with the corresponding HierExplainer estimates of α_c. Table 13 reports RI and F_1 for the original GCN labels and the recovered HierExplainer assignments relative to the known communities. Figure 7 shows the resulting assignments for all 115 teams; the concentration of the confusion matrix along the diagonal indicates that the latent-block procedure recovers much of the known community organization.

These results should be interpreted as a structural recovery diagnostic for HierExplainer. They show that the local latent-block mechanism can align with known graph communities in this dataset, but they do not by themselves establish target-level GNN faithfulness; that property must be assessed separately with keep or remove perturbation or other fixed-model fidelity metrics.

α_i^0	0.078	0.070	0.096	0.104	0.087	0.043
$\hat{\alpha}_i$	0.080	0.078	0.096	0.105	0.086	0.014
α^0	0.113	0.070	0.087	0.104	0.061	0.087
$\hat{\alpha}$	0.124	0.061	0.088	0.112	0.040	0.116

Table 12: HierExplainer estimates $\hat{\alpha}_i$ and their true values α_i^0 for the 12 groups in the Football dataset.

method	RI	F1
GCN	0.886	0.850
HierExplainer	0.922	0.898

Table 13: Node-classification performance on the Football dataset.

C.2 PROTEIN-PROTEIN INTERACTION NETWORK

Table 14 reports the quantitative PPI results. PPI-Essential has no indexed explanatory-edge ground truth. Accordingly, Table 14 reports only native faithfulness, sparsity, and resource measurements. Across these runs, HypoExplainer's FNS is 0.669 for GCN and 0.565 for GIN; the corresponding pairs are 0.204/0.191 for Saliency, 0.274/0.261 for GNNExplainer, 0/0 for PGExplainer, and 0.032/0.091 for SubgraphX. HypoExplainer therefore has the highest F_{NS} for both backbones in this table. Its lower sparsity reflects its retention of edges from the original input graph, which produces denser subgraphs than methods that select individual important edges. HypoExplainer has the lowest available median runtime for both backbones in this table; however, as discussed in Section B.2, these implementation-specific measurements are descriptive and should not be interpreted as a controlled cross-method efficiency ranking.

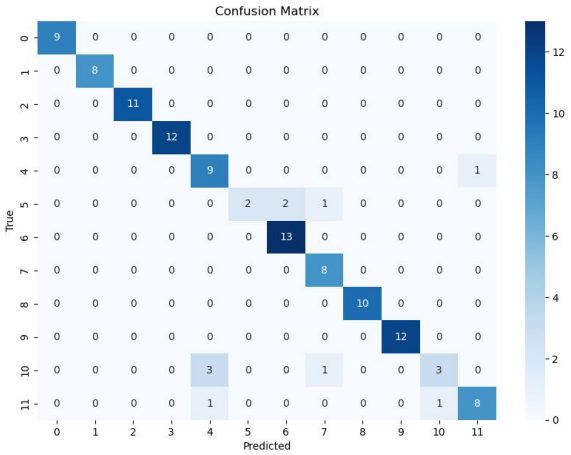


Figure 7: Node-classification confusion matrix for the Football dataset. Diagonal cells count teams assigned to their true labels; off-diagonal cells count misclassifications.

Backbone	Method	PS	PN	F_{NS}	Sparsity	Runtime median [IQR] (s)	RSS median [IQR] (MiB)
GCN	HypoExplainer	**0.817**	**0.567**	**0.669**	0.337	**0.3074 [0.0672]**	4201.34 [41.61]
	Saliency	0.433	0.133	0.204	**0.999964**	0.421659 [0.057944]	**817.23 [198.88]**
	GNNExplainer	0.433	0.200	0.274	0.891	–	–
	PGExplainer	0.433	0.000	0.000	0.9993	0.769257 [0.046450]	2933.930 [802.504]
	SubgraphX	0.600	0.017	0.032	0.9995	2.6069 [3.0550]	2064.3516 [45.8125]
GIN-sum	HypoExplainer	**0.717**	**0.467**	**0.565**	0.416	**0.2953 [0.0547]**	4308.05 [11.75]
	Saliency	0.533	0.117	0.191	**0.999964**	0.508788 [0.154227]	**1872.20 [485.57]**
	GNNExplainer	0.600	0.167	0.261	0.885	2.1512 [0.4197]	2914.00 [0.00]
	PGExplainer	0.533	0.000	0.000	0.9992	1.210757 [0.039779]	3452.820 [507.766]
	SubgraphX	0.533	0.050	0.091	0.9994	18.4248 [6.8844]	1914.3750 [370.9648]

Table 14: PPI-Essential outputs under GCN and GIN-sum backbones. The GCN GNNExplainer source has only total wall time and no target-level runtime/RSS, so those entries remain unavailable. Bold marks the numerical best within each backbone and metric among available cells (maximum for faithfulness and sparsity; minimum for runtime and RSS).

The PPI qualitative example and its interpretation are now presented in Figure 6 in the main text.

APPENDIX D PROOF OF THEORETICAL RESULTS

D.1 NOTATION ALIGNED WITH THE MAIN MANUSCRIPT

Let $\mathcal{P} = \{(k,\ell) : 1 \leqslant k \leqslant \ell \leqslant C, n_{k\ell} > 0\}$ denote the set of the all label pairs of GNN prediction and $m = |\mathcal{P}|$. Define the population set of label pairs whose connection probabilities differ from the background probability by

$$\mathcal{R}^* = \{(k,\ell) \in \mathcal{P} : p_{k\ell} \neq p_{00}\}, \tag{13}$$

and define the Bonferroni-rejected pair set by

$$\widehat{\mathcal{R}}_Z = \left\{(k,\ell) \in \mathcal{P} : p_{\text{value},k\ell} \leqslant \frac{\alpha}{m}\right\}. \tag{14}$$

The corresponding population and estimated label sets are

$$\mathcal{C}_Z^\star = \bigcup_{(k,\ell)\in\mathcal{R}*} \{k,\ell\}, \qquad \widehat{\mathcal{C}}_Z = \bigcup_{(k,\ell)\in\widehat{\mathcal{R}}_Z} \{k,\ell\}. \tag{15}$$

The structural target and the HypoExplainer estimate are then

$$Z_i^\star = \mathbf{1}\{y_{oi} \in \mathcal{C}_Z^\star\}, \qquad \widehat{Z}_i = \mathbf{1}\{y_{oi} \in \widehat{\mathcal{C}}_Z\}, \qquad i \in \mathcal{N}, \tag{16}$$

with

$$\mathbf{Z}_S^\star = \{Z_i^\star : i \in \mathcal{N}\}, \qquad \widehat{\mathbf{Z}}_S = \{\widehat{Z}_i : i \in \mathcal{N}\}.$$

D.2 PROOF OF PROPOSITION 1

Proof. The KL divergence between the variational distribution and the exact posterior is

$$\begin{aligned}&\mathrm{KL}(R(\mathbf{Z}_S \mid Y) \,\|\, P(\mathbf{Z}_S \mid A, Y))\\ &\quad = \mathbb{E}_R[\log R(\mathbf{Z}_S \mid Y) - \log P(\mathbf{Z}_S \mid A, Y)]\,.\end{aligned}$$

Since

$$P(\mathbf{Z}_S \mid A, Y) = \frac{P(A, \mathbf{Z}_S \mid Y)}{P(A \mid Y)}.$$

Substitution gives

$$\begin{aligned}&\mathrm{KL}(R(\mathbf{Z}_S \mid Y) \,\|\, P(\mathbf{Z}_S \mid A, Y))\\ &\quad = \mathbb{E}_R \log R(\mathbf{Z}_S \mid Y) - \mathbb{E}_R \log P(A, \mathbf{Z}_S \mid Y) + \log P(A \mid Y)\\ &\quad = -\mathcal{J}(\mathbf{Z}_S \mid Y) + \log P(A \mid Y).\end{aligned}$$

For fixed A, Y, and structural parameters, $\log P(A \mid Y)$ does not depend on the variational distribution; hence maximizing (5) over the mean-field family is equivalent to minimizing the KL divergence. When $\hat{Z}_j$ is optimized jointly over the structural parameters and $\hat{\tau}_{jc}$, the same variational statement applies at the fitted structural parameters.

To connect this posterior objective in (1), according to

$$P(\mathbf{Z}_S \mid A, Y) = \frac{P(Y \mid A, \mathbf{Z}_S) P(\mathbf{Z}_S \mid A)}{P(Y \mid A)}, \tag{17}$$

under conditions in Proposition 1, for the observed class y,

$$\begin{aligned}\log P(\mathbf{Z}_S \mid A, Y) &= \log P_\Phi(Y = y \mid A, \mathbf{Z}_S) + \log P(\mathbf{Z}_S \mid A) - \log P(Y = y \mid A)\\ &= \mathcal{L}_\Phi(\mathbf{Z}_S) + \log P(\mathbf{Z}_S \mid A) + \text{const}.\end{aligned}$$

The last term is independent of $\mathbf{Z}_S$, which proves (5) in the Proposition.

For the target class c, $\hat{Z}_j$ is approached by

$$\hat{Z}_j = \mathbf{1}[\hat{\tau}_{jc} > 1 - \hat{\tau}_{jc}] = \mathbf{1}\big[\hat{\tau}_{jc} > \tfrac{1}{2}\big]\,. \tag{18}$$

If the mean-field family contains the exact posterior, then the KL minimum is zero and the recovered mode equals a posterior MAP configuration. More generally, since when the variational approximation converges to the exact posterior and the posterior MAP is unique and separated from the remaining configurations, the recovered variational mode converges to maximum of (5). □

D.3 Proof of Theorem 1

Step 1: asymptotic null distribution and validity of the pairwise p-value. Fix a true null pair $(k, \ell) \notin \mathcal{R}^*$, so that $p_{k\ell} = p_{00}$. By the central limit theorem for Bernoulli sample proportions and Slutsky's theorem,

$$\mathcal{Z}_{\mathrm{stat},k\ell} \xrightarrow{d} N(0,1). \tag{19}$$

This is the standard large-sample justification of the two-sample proportion Z test (Agresti, 2013; van der Vaart, 1998). Consequently,

$$P(p_{\mathrm{value},k\ell} \leqslant t \mid H_{0,k\ell}) = t + o(1), \qquad 0 < t < 1. \tag{20}$$

Step 2: family-wise control of false pair inclusions. Let $m_0 \leqslant m$ denote the number of true null pairs. By the Bonferroni union bound (Dunn, 1961) and (20),

$$P\Big(\widehat{\mathcal{R}}_Z \nsubseteq \mathcal{R}^*\Big) \leqslant m_0 \frac{\alpha}{m} + o(1) \leqslant \alpha + o(1). \tag{21}$$

No independence among the different pairwise tests is required for this union-bound step; in particular, the common estimate $\widehat{p}_{00}$ may be reused across tests.

Step 3: asymptotic distribution under a true alternative. Fix $(k, \ell) \in \mathcal{R}^*$, so $\delta_{k\ell} = p_{k\ell} - p_{00} \neq 0$. The independent candidate and background samples, together with the same large-sample regime as above, give

$$\mathcal{Z}_{\mathrm{stat},k\ell} = \frac{\delta_{k\ell}}{\sigma_{0,k\ell}} + \frac{\sigma_{1,k\ell}}{\sigma_{0,k\ell}} \xi_{k\ell} + o_p(1), \qquad \xi_{k\ell} \xrightarrow{d} N(0,1). \tag{22}$$

Equivalently,

$$\mathcal{Z}_{\mathrm{stat},k\ell} \approx N\left(\frac{\delta_{k\ell}}{\sigma_{0,k\ell}}, \frac{\sigma_{1,k\ell}^2}{\sigma_{0,k\ell}^2} \right),$$

which is the standard large-sample approximation used in two-proportion power calculations (Fleiss et al., 2003, Ch. 4).

Step 4: pairwise and family-wise power bounds. The Bonferroni two-sided rejection threshold is

$$c_{\alpha,m} = z_{1-\alpha/(2m)}.$$

For $\delta_{k\ell} > 0$, (22) gives

$$P(\mathcal{Z}_{\mathrm{stat},k\ell} > c_{\alpha,m}) = 1 - \Phi\left(\frac{c_{\alpha,m} - \delta_{k\ell}/\sigma_{0,k\ell}}{\sigma_{1,k\ell}/\sigma_{0,k\ell}} \right) + o(1).$$

If every true alternative satisfies the minimum-signal condition

$$|\delta_{k\ell}| \geqslant z_{1-\alpha/(2m)} \sigma_{0,k\ell} + z_{1-\beta/m} \sigma_{1,k\ell}, \qquad (k, \ell) \in \mathcal{R}^*,$$

then the miss probability of each true pair obeys

$$P\Big((k, \ell) \notin \widehat{\mathcal{R}}_Z\Big) \leqslant \frac{\beta}{m} + o(1). \tag{23}$$

The same conclusion follows for $\delta_{k\ell} < 0$ by applying the argument to $-\mathcal{Z}_{\mathrm{stat},k\ell}$. This critical-value-plus-power-quantile condition is the standard normal-approximation power calculation for comparing two proportions (Fleiss et al., 2003, Ch. 4). Hence, with $s = |\mathcal{R}^*| \leqslant m$, another union bound gives

$$P\Big(\mathcal{R}^* \nsubseteq \widehat{\mathcal{R}}_Z\Big) \leqslant s \frac{\beta}{m} + o(1) \leqslant \beta + o(1). \tag{24}$$

Step 5: exact recovery, node recovery, and the fixed-GNN objective. Combining (21) and (24) yields

$$P(\widehat{\mathcal{R}}_Z \neq \mathcal{R}^*) \leqslant \alpha + \beta + o(1), \qquad P(\widehat{\mathcal{R}}_Z = \mathcal{R}^*) \geqslant 1 - \alpha - \beta - o(1).$$

By (15) and (16), the event $\{\widehat{\mathcal{R}}_Z = \mathcal{R}^*\}$ implies $\{\widehat{\mathcal{C}}_Z = \mathcal{C}_Z^\star\}$ and therefore $\{\widehat{\mathbf{Z}}_S = \mathbf{Z}_S^\star\}$. Thus

$$P\Big(\widehat{\mathbf{Z}}_S = \mathbf{Z}_S^\star\Big) \geqslant P(\widehat{\mathcal{R}}_Z = \mathcal{R}^*) \geqslant 1 - \alpha - \beta - o(1).$$

To connect this structural recovery result to the fixed-GNN prediction objective, define

$$Q_\Phi(\mathbf{z}) = \sum_{c=1}^{C} \mathbf{1}[y_o = c] \log P_\Phi(Y = y_o \mid A, \mathbf{z}), \qquad \mathbf{z} \in \mathfrak{Z}_\mathcal{N},$$

where $\mathfrak{Z}_\mathcal{N} \subseteq \{0,1\}^{n_v}$ is the admissible collection of node-indicator vectors. If the population structural target satisfies

$$\mathbf{Z}_S^\star \in \arg\max_{\mathbf{z} \in \mathfrak{Z}_\mathcal{N}} Q_\Phi(\mathbf{z}),$$

then, on the event $\{\widehat{\mathbf{Z}}_S = \mathbf{Z}_S^\star\}$,

$$\widehat{\mathbf{Z}}_S \in \arg\max_{\mathbf{z} \in \mathfrak{Z}_\mathcal{N}} Q_\Phi(\mathbf{z}).$$

Consequently,

$$P\left(\widehat{\mathbf{Z}}_S \in \arg\max_{\mathbf{z} \in \mathfrak{Z}_\mathcal{N}} Q_\Phi(\mathbf{z})\right) \geqslant 1 - \alpha - \beta - o(1).$$