ActiMine-GNN: Activation Mining and Interpretability in Graph Neural Networks

Kislay Raj[1,2,*], Alessandra Mileo[1,2]

[1]Research Ireland Centre's for Research Training in Artificial Intelligence, DCU, Dublin, Ireland
[2]Insight SFI Research Centre for Data Analytics, School of Computing, Dublin City University, Dublin, Ireland

Abstract

In this paper we introduce **ActiMine-GNN**, a novel framework that derives compact sets of activation rules from hidden layers of GNNs and trains shallow surrogate models for explanation. Unlike prior work that seeks individually discriminative patterns, ActiMine converts layer embeddings into binary activation matrices, mines nonredundant coactivation patterns with FP growth, and trains a depth limited decision tree over pattern features. ActiMine-GNN achieves higher fidelity at substantially lower sparsity than strong baselines, while maintaining task accuracy. A quantitative analysis reports the coverage of the rule set, the conciseness, and the global surrogate alignment with the frozen GNN. We also study robustness to activation thresholding and pattern caps, and report compute and scaling behaviour. These results indicate that compact activation rules can provide transparent and faithful rationales for GNN decisions across synthetic and molecular benchmarks.

Keywords

Neurosymbolic AI, Explainable AI, Graph Neural Networks, Rule Mining,

1. Introduction

GNNs are powerful tools for learning from graph-structured data, achieving state-of-the-art performance in node classification, link prediction, and graph classification [1, 2, 3]. However, explainability still presents open challenges that hinder trustworthy deployment [1]. These include a lack of human-understandable reasoning [2] and the absence of standardised evaluation protocols [3]. In particular, *global* explainability approaches that capture the behaviour of the model across datasets remain under-investigated compared to instance-level methods. In this paper, we focus on addressing two open issues highlighted by recent surveys [4, 5]: (i) the scarcity of *global*, dataset-level GNN explanations, and (ii) the limited availability of quantitative evaluation methods. We address these by mining compact activation rule sets from hidden layers (providing global coverage with local instantiations) and by evaluating faithfulness via fidelity at matched sparsity, rule coverage, and alignment between the surrogate rule model and the GNN.

Most existing explainability techniques for GNNs, such as GNNExplainer [6] and PGExplainer [7], rely on instance-level post hoc explanations. Although useful, these methods typically identify influential nodes or edges, but do not clarify the internal reasoning processes and the deeper semantic relationships captured by the GNN layers. Recent methods like DT+GNN [8], which translate layers with decision trees, provide transparency, but at the expense of reduced model flexibility and scalability. INSIDE-GNN [9] attempts to mine hidden-layer activations to reveal internal GNN reasoning; however, it requires iterative and potentially computationally expensive subgroup discovery, limiting scalability to larger graphs.

Our previous research introduced Functional Semantic Activation Mapping (FSAM) [10], providing global interpretability by tracking neuron activations across layers. FSAM identifies when and where neurons lose their class-specific activations, providing insight into the effects of network depth and

XAI-KRKG@ECAI25: First International ECAI Workshop on eXplainable AI, Knowledge Representation and Knowledge Graphs, October 25–30, 2025, Bologna, Italy

*Corresponding author.
kislay.raj2@mail.dcu.ie (K. Raj); alessandra.mileo@insight-centre.org (A. Mileo)
0000-0003-0089-6866 (K. Raj); 0000-0002-6614-6462 (A. Mileo)

helping to detect scenarios where GNNs achieve correct predictions for incorrect reasons. However, FSAM does not directly translate these activations into human-understandable decision rules or instance-level masks and does not tackle the scalability issue of activation analysis for GNN explainability.
To address this gap, we propose **ActiMine-GNN**, an explainability framework that builds on FSAM, binary activation extraction, frequent pattern mining, and shallow decision trees. ActiMine-GNN first converts hidden-layer activations into binary activation matrices, capturing significant neuron activations across r-hop ego-neighbourhoods. Then we extract frequent coactivation patterns, which we use to construct interpretable activation pattern features. These features are then passed to a shallow decision tree to produce clear, dataset-level rules with per-instance explanations. A common problem when generalising local GNN explainability approaches based on surrogate models to the entire GNN is the need to generate exponentially many possible surrogates on based on the size of the input graph. In contrast, ActiMine-GNN trains a *fixed, small* number of shallow *global* surrogates (e.g., one tree per layer or a single tree on concatenated features), caps the number of mined patterns (top-M per layer), and thus avoids surrogate proliferation while keeping the rule set compact.

The method is applicable to both the node and graph classification settings; in this paper, we focus on the classification of the graph and evaluate the approach on three standard benchmarks used in GNN explainability: **BA-2Motifs** [6] (synthetic motif detection), **AIDS** [11] (antiviral activity classification on molecular graphs), and **BBBP** [12] (blood–brain barrier permeability). These data sets are widely adopted to assess the fidelity and sparsity of subgraph-based explanations and provide a mix of synthetic ground truth (BA-2Motifs) and chemically meaningful structure (AIDS/BBBP). Across these benchmarks, ActiMine-GNN achieves higher fidelity at matched sparsity while maintaining task accuracy, providing transparent and compact global rules.

2. Proposed Methodology: ActiMine-GNN

In this paper, we consider ActiMine-GNN on Graph Convolutional Networks (GCNs) [13] for *binary* graph classification because the benchmarks we evaluate (BA-2Motifs, AIDS, BBBP) are binary tasks, and this facilitates comparative fidelity analysis. This choice is not a limitation of the approach, as the pipeline is model-agnostic and can be extended to multiclass classification by training a multiclass surrogate (or one-vs-rest surrogate) on the same pattern features and ranking mined patterns by their association with each class; evaluation then uses class-conditional fidelity at matched sparsity. The approach is built upon the activation-extraction approach of FSAM [10], which provides a *global* understanding of coactivation of a trained GNN by (i) extracting layer-wise activations and binarising them to form activation matrices, and (ii) building a *co-activation graph* over hidden units to analyse how class-specific activations emerge or dissipate across layers. FSAM produces model-level insights (e.g., loss of class specificity with depth), but it does not produce decision rules or instance-level masks. We keep the FSAM activation extraction step unchanged: namely, we derive binary activation matrices from ReLU embeddings for each hidden layer. We then replace FSAM's coactivation graph construction with a rule mining pipeline as follows: (1) for each layer, form transactions from r hop ego-neighbourhoods; (2) mine frequent coactivation patterns under support / top-M constraints to obtain activation pattern features; (3) train a depth-limited decision tree on these features to extract interpretable activation rules; and (4) induce per-instance masks from the selected rules for evaluation. Figure 1 illustrates this process: the left block corresponds to the result of the FSAM activation-extraction step which represents the input, the middle and right blocks implement our pattern mining and rule induction pipeline, respectively.

2.1. Binary Activation Matrix

We begin by converting each continuous GCN embedding into a discrete activation pattern a step motivated by the focus of FSAM on neuron activations [10], but here specialised in rule mining in graph classification. Let $\mathcal{G} = \{G_i\}_{i=1}^{N}$ with $G_i = (V_i, E_i, X_i, y_i)$, $n_i = |V_i|$, and $y_i \in \{c_0, c_1\}$. Here i is the graph index (GID) and $v \in V_i$ the node index (NID); let $n_{\text{tot}} = \sum_{i=1}^{N} n_i$.

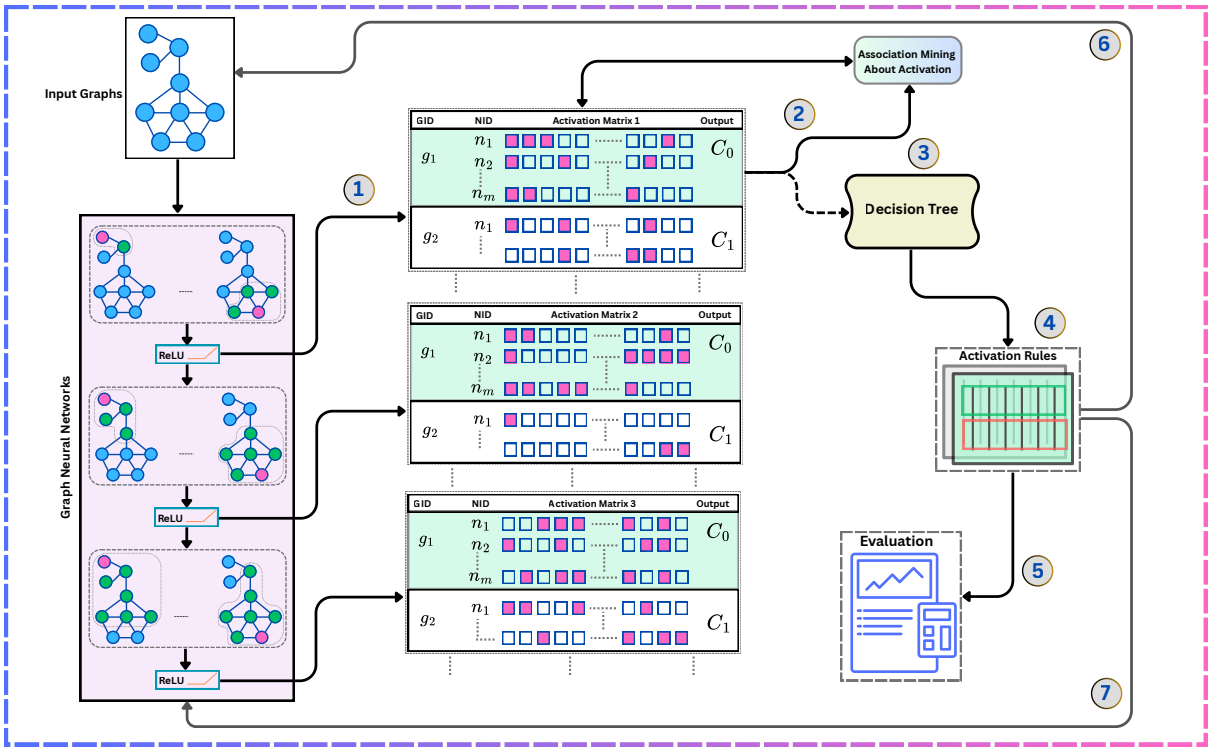

Figure 1: Overview of ActiMine-GNN. For each hidden layer ℓ: (1) compute node embeddings $H^{(\ell)}$ from the input graph (node colours indicate input categories only and are not explanations); (2) binarise activations to obtain $B^{(\ell)} = \mathbb{1}\{H^{(\ell)} > \tau\}$; (3) mine frequent coactivation patterns from $B^{(\ell)}$ using FP-growth, optionally filtered by a maximum entropy background model; (4) fit a depth limited decision tree on pattern features to obtain activation rules; (5) instantiate instance level node masks from the selected rules; (6) evaluate at the same sparsity budget α (fidelity, 1−fidelity, and sparsity). Dotted stacks indicate repetition across layers ($\ell = 1, \ldots, L$), producing per layer rule sets and masks. In $B^{(\ell)}$, rows are indexed by (graph ID, node ID) and ones mark active embedding components or (7) to provide insights on the model

A trained L-layer GCN produces continuous activations

$$H_i^{(\ell)} = \left[h_{i,v}^{(\ell)}\right]_{v \in V_i} \in \mathbb{R}^{n_i \times K}, \qquad h_{i,v}^{(\ell)} = \mathrm{ReLU}\left(W^{(\ell)} \sum_{u \in N_i(v) \cup \{v\}} \frac{h_{i,u}^{(\ell-1)}}{\sqrt{d_{i,u}\, d_{i,v}}}\right),$$

with $h_{i,v}^{(0)} = X_{i,v}$, $W^{(1)} \in \mathbb{R}^{K \times d}$ and $W^{(\ell)} \in \mathbb{R}^{K \times K}$ for $\ell > 1$. FSAM continues with continuous correlations on $H_i^{(\ell)}$ to build a global coactivation graph; instead, we discretise $H_i^{(\ell)}$ channel-wise to obtain binary indicators for pattern mining:

$$B_i^{(\ell)}(v,k) = \mathbb{1}\{h_{i,v,k}^{(\ell)} > \tau_k\} \in \{0,1\}, \qquad B_i^{(\ell)} \in \{0,1\}^{n_i \times K},$$

where thresholds $\{\tau_k\}$ are chosen in validation (ReLU's > 0 is a special case; layer-specific $\tau_k^{(\ell)}$ are also admissible). Binarisation (i) produces transactions for scalable itemset mining, (ii) normalises away arbitrary activation scales across layers/graphs, and (iii) produces crisp rule conditions for shallow surrogates. This choice is orthogonal to the number of classes and works unchanged for multiclass (one-vs-rest or a multiclass surrogate).

2.2. Frequent Coactivation Mining

Transactions and background. For layer ℓ, define the transaction of each node $T_{i,v}^{(\ell)} = \{k : B_i^{(\ell)}(v,k) = 1\}$. The marginal activation probability (independent Bernoulli / MaxEnt under fixed

marginals) is

$$p_k^{(\ell)} = \frac{1}{n_{\text{tot}}} \sum_{i=1}^{N} \sum_{v \in V_i} B_i^{(\ell)}(v,k).$$

Itemset mining algorithm We extract frequent itemsets $I \subseteq \{1,\dots,K\}$ with *FP-growth*, which compresses transactions into an FP tree and avoids the candidate explosion of Apriori; it is robust on sparse binary activations and fast at moderate K. We considered Eclat as an alternative, but empirically determined that FP growth was consistently faster in our setting.

Support and its interpretation Global support is the fraction of nodes (across all graphs) whose transaction contains I:

$$\text{supp}(I) = \frac{1}{n_{\text{tot}}} \sum_{i=1}^{N} \left| \{\, v \in V_i \;:\; I \subseteq T_{i,v}^{(\ell)} \,\} \right|.$$

In an independence context, the expected support for I is $\prod_{k \in I} p_k^{(\ell)}$; deviations from $\prod p_k^{(\ell)}$ indicate association beyond chance.

Threshold selection and redundancy control We set a minimum support per layer σ by validation on a small grid (e.g. $\{0.01, 0.02, \dots, 0.10\}$), choosing the *largest* σ that still produce at least M patterns after pruning. We prune through *closed* itemsets (removing subsumed patterns) to avoid near duplicates.

Ranking: support, confidence, and lift For graph-level ranking, define a presence indicator $\psi_I(G_i) = \mathbb{1}\{\Phi_I(G_i) > 0\}$ with

$$\Phi_I(G_i) = \frac{1}{n_i} \sum_{v \in V_i} \mathbb{1}\{I \subseteq T_{i,v}^{(\ell)}\}.$$

Given class c, the *confidence* and *lift* are

$$\text{conf}(I \to c) = \Pr(y_i = c \mid \psi_I(G_i) = 1), \qquad \text{lift}(I \to c) = \frac{\Pr(y_i = c \mid \psi_I(G_i) = 1)}{\Pr(y_i = c)}.$$

We first filter by $\text{supp}(I) \geq \sigma$ (unsupervised frequency), then rank by conf (class association) and report lift > 1 as evidence over background prevalence. We keep the top-M non-redundant itemsets per layer as *salient* patterns, that is, frequent and class-associated (lift > 1). Unless otherwise stated: $r = 1$ (ego-neighbourhood for transactions), $M = 100$ per layer, σ selected as above; Multiple testing control through Holm adjustment when screening many I.

2.3. Activation Rule Induction via Decision Trees

For each selected itemset I, define node- and graph-level features

$$\phi_I(i,v) = \mathbb{1}\{I \subseteq T_{i,v}^{(\ell)}\}, \qquad \Phi_I(G_i) = \tfrac{1}{n_i} \sum_{v \in V_i} \phi_I(i,v).$$

We train a shallow decision tree surrogate on graph features $\left([\Phi_I(G_i)]_{I \in \mathcal{F}}, y_i\right)_{i=1}^{N}$. We limit the depth at 4 (min leaf = 3) as the length of the rule (at most four feature tests per rule) to reduce overfitting and keep the set of rules compact; deeper trees yielded only marginal gains in fidelity against substantially longer rules in validation.

Each root-to-leaf path produces a human-readable activation rule.

$$\bigwedge_{j=1}^{m} \left(\Phi_{I_j}(G_i) \bowtie_j t_j\right) \Rightarrow \hat{y}_i = c, \qquad \bowtie_j \in \{\geq, <\},$$

which can be visualised as per-instance masks by highlighting nodes with $\phi_{I_j}(i,v) = 1$ for the selected rule conditions.

2.4. what do the mined rules look like?

Rule form: From each layer ℓ, ActiMine-GNN mines frequent coactivation patterns p_i (sets of activated embedding components) and trains a depth limited decision tree over *pattern-count* features. A rule therefore has the form

$$\textbf{Rule } k \text{ (Layer } \ell,\ \text{class } c) :\quad \big[\text{count}(p_{i_1}) \geq t_1\big] \wedge \cdots \wedge \big[\text{count}(p_{i_m}) \geq t_m\big] \ \Rightarrow\ c,$$

where count(p_i) is the number of nodes in the graph whose ℓ-layer binary activations include pattern p_i, and t_j are small integer thresholds (typically 1–3). We annotate each rule with support (coverage on test graphs), precision (class-conditional accuracy on covered graphs) and lift (increase over class prior).

BA-2Motifs example (Layer 2, class c_0) Consider a 5-cycle graph from BA-2Motifs (class c_0). One high-ranking Layer 2 rule for c_0 is:

$$\textbf{Rule 4 (L2, } c_0\textbf{)} :\quad \big[\text{count}(p_{13}) \geq 1\big] \wedge \big[\text{count}(p_{27}) \geq 1\big] \ \Rightarrow\ c_0.$$

Here p_{13} and p_{27} are Layer 2 coactivation patterns mined by FP-growth; intuitively, they correspond to embedding components that tend to co-occur on cycle nodes. On a graph where Rule 4 fires, *instance level mask* is formed by nodes that support the antecedent patterns (optionally expanded by the chosen mask variant, e.g., ego neighbourhood or top-k edges). In the same sparsity budget α, retaining the masked subgraph reproduces the original class on most covered graphs (Layer 2 fidelity $= 0.94$ in BA-2Motifs; Table 3); deleting it substantially reduces confidence in c_0 (infidelity $= 1 - 0.94 = 0.06$), illustrating necessity.

How the rule produce a mask. Given a firing rule, we score nodes by their antecedent support (number of satisfied patterns) and select the highest scoring nodes until the budget α is met; the chosen mask variant (node-removal, ego, distance-weighted, or top-k edges) is then applied. This produces a compact, human interpretable subgraph that links the global rule to a local rationale. Figure 2 shows an example mask induced by Rule 4 on a 5-cycle: removing the masked node breaks the cycle into a path and often changes the prediction, while retaining the mask alone preserves the class.

Molecular example. In BBBP, Layer 2 rules typically combine a small number of patterns that occur on substructures linked to permeability (e.g., ring and heteroatom contexts). The local mask highlights the few atoms whose Layer 2 activations satisfy the antecedent, providing concise explanations at $\alpha \approx 0.17$ with fidelity 0.88 and infidelity 0.12 (Table 3).

3. Experiments and Results

To evaluate the effectiveness of ActiMine-GNN, we performed experiments on benchmark graph classification datasets.

3.1. Dataset Summary

We evaluated GNN interpretability using a pipeline comprising activation mining, decision tree rule extraction, and XGBoost-based rule validation. Experiments were performed on three different binary graph classification benchmark datasets:

- BA2 [6]: A synthetic dataset containing 1,000 graphs with balanced classes, each featuring either a 5-cycle or a house motif as ground truth
- AIDS [11]: 2,000 molecular graphs (atoms as nodes, bonds as edges) with a binary label for anti-HIV activity; node and edge types provided; classes are imbalanced.

- BBBP [12]: blood–brain barrier permeability (permeable vs non-permeable); 1,640 molecular graphs retained after standard sanitisation and deduplication.

We quantify interpretability with (i) *fidelity* (agreement between the surrogate and the frozen GNN) and (ii) *sparsity*, defined as the budget of the instance-level mask, i.e. the fraction of nodes (or edges) retained $\alpha = \frac{|M_V|}{|V|}$ (lower is better). We also report *rule conciseness* as the average number of rule conditions per leaf (equivalently, the average path length in the decision tree). XGBoost's feature importance is used only to validate that the features of the mined patterns carry a predictive signal; it is *not* our sparsity metric. These metrics are especially relevant for the prediction of molecular properties, where concise and interpretable explanations are critical for domain experts. We quantify interpretability with (i) fidelity (sufficiency of the masked subgraph to reproduce the frozen GNN's prediction) and (ii) sparsity, defined as the mask budget α (fraction of nodes retained; lower is better).

Table 1 summarises the principal properties of each dataset as *Neg. Ex./Pos. Ex.* denotes the number of graphs with negative/positive labels in the full data set after preprocessing (before splitting). *Avg. Nodes/Avg. Edges* are means per graph. *Train/Val/Test Acc.* are the accuracies of the final checkpoint in the corresponding groups.

Table 1: Dataset characteristics and model performance metrics.

Dataset	Neg. Ex.	Pos. Ex.	Avg. Nodes	Avg. Edges	Train Acc.	Test Acc.	Val. Acc.
BA-2Motifs [6]	500	500	25.00	50.92	0.995	0.970	1.000
AIDS [11]	400	1600	15.69	32.39	0.989	0.990	0.975
BBBP [12]	389	1251	24.08	51.96	0.855	0.787	0.848

The GNN used is a 3-layer GCN (20 hidden units per layer, ReLU, global mean pooling), trained with an 80/10/10 train/validation/test split; hyperparameters selected by grid search;

Evaluation metrics. Let $f(G) \in \mathbb{R}^C$ be the frozen softmax scores of GNN for graph G, and $c^\star = \arg\max_c f_c(G)$ its predicted class. Given a rule set at layer ℓ, let $M^{(\ell)}(G) \subseteq V(G)$ be the node mask it selects (a union of all applicable rule bodies), and define

$$\alpha^{(\ell)}(G) = \frac{|M^{(\ell)}(G)|}{|V(G)|} \quad \text{(node sparsity; lower is better).}$$

Write $G^{(\ell)}_{\text{keep}} = G\big[M^{(\ell)}(G)\big]$ and $G^{(\ell)}_{\text{del}} = G\big[V(G) \setminus M^{(\ell)}(G)\big]$. We evaluated on the subset of *covered* test graphs $\mathcal{S}^{(\ell)} = \{\, G \in \mathcal{D}_{\text{test}} \;:\; |M^{(\ell)}(G)| > 0 \,\}$ (coverage $= |\mathcal{S}^{(\ell)}|/|\mathcal{D}_{\text{test}}|$).

Fidelity (sufficiency, hard):

$$\text{Fid}^{+}_{\text{hard}}(\ell) = \frac{1}{|\mathcal{S}^{(\ell)}|} \sum_{G \in \mathcal{S}^{(\ell)}} \mathbb{1}\Big[\arg\max_c f_c\big(G^{(\ell)}_{\text{keep}}\big) = c^\star\Big].$$

Infidelity (lower is better): We report infidelity as the complement of fidelity in the same sparsity budget α and in the same covered set: $\text{Infidelity}^{(\ell)}(\alpha) = 1 - \text{Fidelity}^{(\ell)}(\alpha)$.

Sparsity (mask budget):

$$\overline{\alpha}^{(\ell)} = \frac{1}{|\mathcal{S}^{(\ell)}|} \sum_{G \in \mathcal{S}^{(\ell)}} \alpha^{(\ell)}(G).$$

Unless stated otherwise, we report *means over the covered test graphs* $\mathcal{S}^{(\ell)}$; fidelity is calculated at the (matched) sparsity reported in Table 2, and infidelity is the complement of fidelity in the same sparsity budget. Soft variants of fidelity can be obtained by replacing the indicator with $f_{c^\star}(G^{(\ell)}_{\text{keep}})$.

3.2. Rule Extraction Robustness

Among all datasets, ActiMine-GNN extracted between 8 and 15 high-fidelity activation rules per layer. Although we extracted and evaluated the rules at each layer, we focus on the results of layer 2 (L2) in Table 2. This is because across Layers 1⟶3, fidelity only fluctuates by ±0.08, and the fraction of nodes retained by the mask (sparsity) by at most 0.13 relative to the L2 value, confirming the stability of our experiment. We can see that the highest fidelity is achieved in the BA-2Motifs dataset, where Layer 2 produced 12 rules that apply to more than 95 % of the test graphs, with an average fidelity of 0.94.

Table 2
Per-layer rule extraction and local explanation metrics (L1–L3). L2 entries are recorded point estimates (fidelity/infidelity/sparsity from Table 3); L1/L3 entries are the logged per-layer values. Infidelity is defined as $1 - \text{Fidelity}$ at the same budget α as fidelity. "Avg." denotes the mean over covered test graphs $\mathcal{S}^{(\ell)}$.

Dataset	Layer	# Rules	Avg. Fidelity ↑	Avg. Infidelity ↓	Avg. Sparsity (α) ↓
BA-2Motifs	L1	14	0.91	0.09	0.25
	L2	12	0.94	0.06	0.18
	L3	8	0.89	0.11	0.22
AIDS	L1	8	0.83	0.17	0.28
	L2	10	0.91	0.09	0.15
	L3	9	0.88	0.12	0.20
BBBP	L1	9	0.85	0.14	0.27
	L2	13	0.88	0.12	0.17
	L3	15	0.87	0.13	0.22

Notes. Variability across layers is modest: $|\text{Fid}_\ell - \text{Fid}_{L2}| \leq 0.08$ and $|\alpha_\ell - \alpha_{L2}| \leq 0.13$ across datasets. Infidelity is reported as $1 - \text{Fidelity}$ at the same budget α as fidelity.

Key observations.

- **Fidelity** ≥ 0.88 across BA-2Motifs (0.94), AIDS (0.91), and BBBP (0.88) indicates that, at the stated budgets, the masked subgraph alone reproduces the original prediction on most test graphs.
- **Infidelity** ≤ 0.12 in the same budget α (0.06, 0.09, 0.12 respectively) shows that the removal of the highlighted subgraph markedly reduces the confidence of the model in the original class, signaling the necessity of the selected nodes.
- **Sparsity** 0.15–0.18 (mean ≈ 0.17) produces concise explanations, with about 17% nodes retained.

3.3. Instance-Level Explanations

ActiMine-GNN discovers twelve high-fidelity activation rules on the BA-2Motifs dataset at layer 2. BA-2Motifs comprises 1000 graphs: 500 hiding a 5-cycle (class c_0) and 500 hiding a 5-node 'house' motif (class c_1). ActiMine-GNN exactly recovers the correct explanatory substructure for each graph, outperforming all baseline methods. In Figure 2, we display a single test graph (a 5-cycle) and highlight its Rule 4 support set

$$V_{\text{mask}} = \{v_1\}$$

in red. Removing v_1 (and its incident edges) breaks the cycle into a simple path, causing the prediction of GNN to change from '5 cycle' to 'house'. This flip occurs in 94 % of the graphs where Rule 4 applies, producing an instance-level fidelity of 0.94. For each class c and layer ℓ we pre-rank rules by *confidence*, breaking the links by *lift* (see definitions in Section2.2), and require a minimum test-coverage threshold. The example in Fig. 2 shows the layer 2 rule ranked top for class c_0 among those that meet a predefined coverage threshold. We rank rules by validation set fidelity, breaking ties by higher support and greater conciseness (shorter path length). This fixed procedure, applied uniformly across datasets, avoids selective reporting and ensures comparability across datasets. Although the node removal mask is the simplest, **ActiMine-GNN** also supports more expressive strategies. The *node-removal (keep) mask*

retains only the nodes in V_{mask} and the edges they induce, removing everything else. The *ego-graph mask* retains the full radius-ℓ neighbourhood around each $v \in V_{\text{mask}}$, capturing the multi-hop context. The *distance-weighted mask* assigns continuous weights that decay with the shortest path distance from V_{mask} (measured in hops for unweighted graphs and by path length for weighted graphs), producing a soft subgraph. Finally, the *top-k edge mask* selects the k highest-weight edges under the distance-weighted scheme to produce a compact, human-readable subgraph. Unless stated otherwise, for each instance, we choose the mask variant that maximises fidelity in the same sparsity budget α, and we report the results obtained α.

3.4. Comparison with State-of-the-Art Methods

Table 3 reports a comparison of **ActiMine-GNN** with three widely used instance level GNN explanation methods: GNNExplainer, PGExplainer and PGM-Explainer. **Fidelity** measures the degree to which the masked subgraph is sufficient for the model to reproduce its original prediction: for each method and each test graph, we apply the mask of that method, retain only the highlighted subgraph, and check if the predicted class is unchanged; fidelity is the average proportion of the covered test graphs. **Infidelity** is reported as the complement of fidelity at the same sparsity budget α and on the same set covered. **Sparsity** quantifies conciseness and is the fraction of nodes retained by the mask (lower is better). As shown in Table 3, across BA-2Motifs, AIDS, and BBBP, ActiMine-GNN achieves the highest fidelity and the lowest infidelity while producing the most concise masks. These results indicate that our rule-driven approach aligns closely with the model decision process and yields human-interpretable focused rationales. For completeness, the table reports each baseline in its achieved sparsity, and ActiMine-GNN uses the Layer 2 configuration.

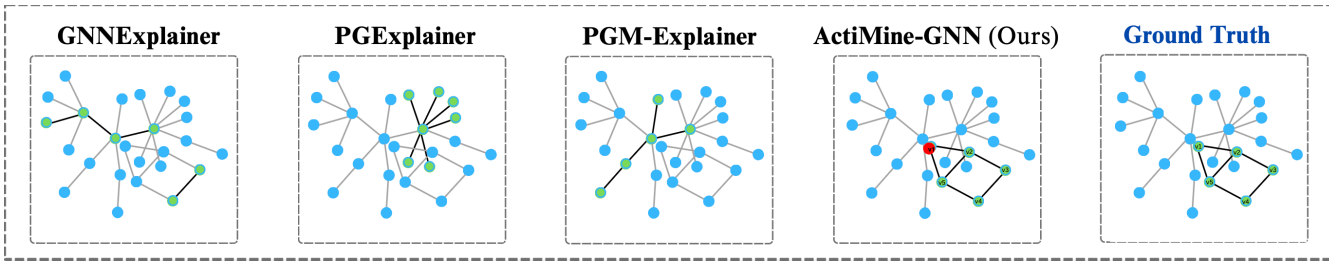

Figure 2: Comparison of the explanations on BA-2Motifs with GCN classifier

4. Conclusion and Future Work

In this paper we presented ActiMine-GNN, a rule driven framework for understanding GNN decision processes. The method mines compact activation rules from binarised hidden layer activations using decision tree induction and applies these rules to generate instance level masks for individual graphs. We quantitatively validated ActiMine-GNN on benchmark datasets, focussing on both the intrinsic quality of the rules discovered and their ability to faithfully explain the model's decisions. To assess whether the extracted rules characterise the overall behaviour of the GNN, we used rule-derived characteristics to train XGBoost surrogates and measured their agreement with the frozen GNN. For each input graph, the feature vector encodes the number of nodes that support each activation rule. The high alignment between surrogate predictions and the original GNN output indicates that the rules capture the core predictive logic of the model. We chose decision trees for rule extraction due to their clarity and interpretability, and XGBoost for quantitative validation because it reliably assesses the predictive strength of the rule derived features. Our pipeline binarises activations and relies on support thresholds and top M caps; while this confers scalability and interpretability, performance can depend on these hyperparameters. Experiments focus on graph level binary classification with a GCN backbone; although the approach is model agnostic and extends to multiclass settings (for example, one

Table 3
Performance of explanation methods across datasets. Metrics: Fidelity (higher is better), Infidelity (1−Fidelity at that method's achieved budget α; lower is better), and Sparsity (fraction of nodes retained; lower is better). Values are means over covered test graphs $\mathcal{S}$. *Note:* Baseline methods are reported at their achieved sparsities; ActiMine-GNN uses the Layer−2 configuration from Table 2.

Dataset	Method	Fidelity	Infidelity	Sparsity
BA-2Motifs	**ActiMine-GNN (L2)**	**0.94**	**0.06**	**0.18**
	GNNExplainer [6]	0.38	0.62	0.61
	PGExplainer [7]	0.35	0.65	0.95
	PGM-Explainer [14]	0.90	0.10	0.65
AIDS	**ActiMine-GNN (L2)**	**0.91**	**0.09**	**0.15**
	GNNExplainer [6]	0.49	0.51	0.50
	PGExplainer [7]	0.46	0.54	0.54
	PGM-Explainer [14]	0.47	0.53	0.85
BBBP	**ActiMine-GNN (L2)**	**0.88**	**0.12**	**0.17**
	GNNExplainer [6]	0.50	0.50	0.50
	PGExplainer [7]	0.51	0.49	0.53
	PGM-Explainer [14]	0.48	0.52	0.88

versus rest or multiclass surrogates), we have not exhaustively evaluated alternative architectures and leave this for future work. Although this work emphasised local masks, examining the sets of rules global would help to assess how well the method characterises the GNN as a whole. This includes: (a) measuring the coverage of the rule set and the conditional accuracy of the class on the highlighted graphs; (b) quantifying the agreement between a global surrogate trained on the features derived from the rules and the frozen GNN; (c) performing rule adjustments (dropping or adding rules) to estimate marginal contributions; and (d) checking the consistency between global rules and local masks, that is, that the masks instantiated from a given rule are predominantly produced for the class of that rule and tend to increase local fidelity. Finally, we anticipate that models capable of generating prototype subgraphs linked to concise global rules (with per class exemplars and counterfactual contrasts) will offer case level transparency and auditable rule sets, supporting adoption in regulated domains such as healthcare, finance, and autonomous systems.

Acknowledgment

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18CRT6223 and Research Ireland INSIGHT Centre for Data Analytics, Grant no. 12/RC/2289 P2.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] L. Waikhom, R. Patgiri, A survey of graph neural networks in various learning paradigms: methods, applications, and challenges, Artificial Intelligence Review 56 (2022) 6295–6364. doi:10.1007/s10462-022-10321-2.

[2] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, J. Wang, A comprehensive survey of graph neural networks for knowledge graphs, IEEE Access 10 (2022) 75729–75741. doi:10.1109/access.2022.3191784.

[3] E. Şahin, N. N. Arslan, D. Özdemir, Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning, Neural Computing and Applications (2024). doi:10.1007/s00521-024-10437-2.

[4] M. Saarela, V. Podgorelec, Recent applications of explainable ai (xai): A systematic literature review, Applied Sciences 14 (2024) 8884. doi:10.3390/app14198884.

[5] A. Longa, S. Azzolin, G. Santin, G. Cencetti, P. Lio, B. Lepri, A. Passerini, Explaining the explainers in graph neural networks: a comparative study, ACM Computing Surveys (2024). doi:10.1145/3696444.

[6] Z. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, Gnnexplainer: Generating explanations for graph neural networks, in: NeurIPS, 2019. URL: https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html.

[7] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, X. Zhang, Parameterized explainer for graph neural network, arXiv preprint arXiv:2011.04573, 2020. doi:10.48550/arxiv.2011.04573.

[8] P. Müller, L. Faber, K. Martinkus, R. Wattenhofer, Dt+gnn: A fully explainable graph neural network using decision trees, arXiv preprint arXiv:2205.13234, 2022. doi:10.48550/arxiv.2205.13234.

[9] L. Veyrin-Forrer, A. Kamal, S. Duffner, M. Plantevit, C. Robardet, On gnn explainability with activation rules, Data Mining and Knowledge Discovery 38 (2022) 3227–3261. doi:10.1007/s10618-022-00870-z.

[10] K. Raj, A. Mileo, Towards understanding graph neural networks: Functional-semantic activation mapping, in: International Conference on Neural-Symbolic Learning and Reasoning, Springer, 2024, pp. 98–106.

[11] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, M. Neumann, Tudataset: A collection of benchmark datasets for learning with graphs, arXiv preprint arXiv:2007.08663 (2020).

[12] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, V. Pande, Moleculenet: a benchmark for molecular machine learning, Chemical Science 9 (2017) 513–530. doi:10.1039/c7sc02664a.

[13] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[14] M. N. Vu, M. T. Thai, Pgm-explainer: Probabilistic graphical model explanations for graph neural networks, in: Advances in Neural Information Processing Systems (NeurIPS), volume 33, 2020, pp. 12225–12235. URL: https://proceedings.neurips.cc/paper/2020/file/8fb134f258b1f7865a6ab2d935a897c9-Paper.pdf.