

A Neuro-Symbolic Framework to Enhance Interpretability of Graph Neural Networks

Kislay Raj, M.S.c.

Supervised by Prof. Alessandra Mileo



A thesis presented for the degree of Doctor of Philosophy

(Ph.D.)

SCHOOL OF COMPUTING,
DUBLIN CITY UNIVERSITY

Aug 2026

Declaration

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original and have conformed to the regulations on the use and declaration of Generative AI, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Kislay Raj

ID no. 21267875

13 August 2026

Generative Artificial Intelligence

I hereby certify that Generative Artificial Intelligence (GenAI) tools were used in the preparation of this thesis solely for language-support purposes, specifically grammar and spelling checking and proofreading. GenAI was not used to generate the research methodology, analysis, results, discussion, conclusions, or original intellectual contributions of this thesis.

The DCU-whitelisted Google Gemini 2.5 Flash tool was used. All suggestions were reviewed and evaluated by the author, who retains full responsibility for the content and academic integrity of this thesis.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Alessandra Mileo, for giving me the opportunity to explore this fascinating subject and for guiding me through all the ups and downs of this journey. Her invaluable guidance, expertise, patience, and unwavering support over the past four years have been a constant source of encouragement. This PhD would not have been possible without her belief in me, and I am truly grateful for her support, mentorship, and encouragement throughout this journey.

I am deeply grateful to my wife, Priyanka Kumari, for her constant love and encouragement during challenging times. Her constant presence has been a source of strength throughout this journey.

I would also like to thank the Science Foundation Ireland Centre for Research Training in Artificial Intelligence for funding my PhD. This research was supported by Science Foundation Ireland (SFI) under Grant Number 18/CRT/622.3.

My heartfelt thanks go to my late grandfather, Ramdev Kunwar Singh, an inspiring mathematician who first introduced me to the subject. I would also like to express my sincere gratitude to my parents, Satish Kumar Singh and Punam Devi, my brother and sister, Harsh Raj and Kumari Sugandha, and my dear friends, Teerath Kumar, Praveen Acharya, and Abhishek Mandal, for their unwavering support, friendship, and encouragement throughout this journey.

Finally, I wish to thank all those who have contributed directly or indirectly to this work, including my colleagues and fellow researchers. Your support and collaboration are greatly appreciated.

Contents

Abstract	13
1 Introduction	15
1.1 Introduction to Explainable AI	19
1.2 Introduction to GNN	21
1.3 Neuro-Symbolic Background for XAI	24
1.4 Hypotheses and Research Questions	26
1.5 Thesis Structure	28
2 Literature Review	30
2.1 GNN Explainability Methods	31
2.1.1 Instance level Explanation	31
2.1.2 Model Level Explanation	34
2.1.3 Explanations in Graph Neural Networks	36
2.1.4 Neuro-Symbolic Approaches for GNN Explainability	37
2.1.5 Benchmarks and Datasets	39
2.1.6 Evaluation Metrics and Protocols	40
2.1.7 Taxonomy of Neurosymbolic Methods	43
2.2 Gaps in GNN Explainability And Motivation	45
3 Global Explanations using Functional-Semantic Activation Mapping	48
3.1 Introduction	48
3.2 (RQ1): FSAM Graph: Definition and Construction	49

3.2.1	Overall Methodology: Generating the Semantic Graph	50
3.2.2	Mathematical formulation	51
3.2.3	Semantics in FSAM: Model-Internal and Human Tiers	56
3.3	Evaluation: Experimental Validation of FSAM	57
3.3.1	Datasets and Experimental setup	57
3.3.2	(RQ2): Validating the FSAM Representation (layer wise mapping to output classes)	60
3.3.3	(RQ3): Interpreting Class Confusions via Community Detection and Class Pair Similarity	68
3.4	Qualitative Analysis	70
3.5	Limitations and Design Trade-offs	71
3.6	Summary	71
4	Measuring Semantic Collapse: Depth, Oversmoothing, and Discriminative Power	74
4.1	Introduction and Motivation	74
4.2	Extended Validation of FSAM Across Layer Configurations	76
4.3	(RQ4): Depth-wise Behaviour and Discriminative Power	81
4.4	(RQ5): Community overlap and class similarity	88
4.4.1	Scalability and Computational Complexity	94
4.4.2	Reading the Figures as an Ensemble	94
4.5	Limitations	95
4.6	Summary	96
5	Local Rule-Based Explanations for GNNs: A Semantic Activation-Mining Approach (ActiMine-GNN)	98
5.1	Introduction	98
5.2	Methodology	100
5.2.1	From Node-Level FSAM to Graph-Level ActiMine	100
5.2.2	(RQ6): Mining Compact local Activation Rules (ActiMine) . .	101

5.2.3	Instantiating Local Explanations from Rules	105
5.3	Experimental Evaluation	107
5.3.1	Datasets, Backbone, and Evaluation Protocol	107
5.3.2	(RQ7): Visual Analysis of Factual Explanations and Counter- factual Edits	112
5.3.3	Instance-Level Explanations	113
5.3.4	Comparison with State-of-the-Art Explainers	114
5.3.5	Qualitative Analysis	116
5.4	Limitations and Design Trade-offs	117
5.4.1	Towards a More Systematic Evaluation of Local Explanations	119
5.5	Summary	120
6	Conclusions	122
6.1	Introduction	122
6.2	The Functional Semantic Activation Mapping Graph Representation .	123
6.3	Measuring Semantic Collapse: Depth, Oversmoothing and Discrimi- native Power	125
6.4	Generating Local Rule Based Explanations through Semantic Acti- vation Mining	126
6.5	Challenges and Opportunities Ahead	128
A	Publications on Work from this Thesis	133
A.1	Journal Publications	133
A.2	Conference Publication	133
B	Awards	135
	Bibliography	136

List of Figures

1.1	An example of a message passing architecture	23
2.1	Overview of the taxonomy of GNN explainability methods [31]	33
3.1	The FSAM pipeline. A trained GCN is run on the input graph, per-layer activations are captured, neuron–neuron Spearman correlations and neuron–class point-biserial associations are computed, and global weight thresholding produces one FSAM graph per layer, which is subsequently partitioned into communities.	52
3.2	FSAM graph for layer k of a 4-layer GCN on Cora. Blue nodes are neurons, red nodes are predicted classes, edges are co-activations retained at $\theta = 0.3$. [Layer 1: communities are distinct and moderately sparse, with each group associated with a different class.]	62
3.3	Layer 2: communities are more compact and some merge.	62
3.4	Layer 3: community boundaries blur and edge density rises, indicating loss of neuron specialisation.	63
3.5	Layer 4: communities merge extensively and the graph becomes densely connected, the pattern we term semantic collapse.	63

3.6	Class-pair misclassification count against Jaccard similarity of the corresponding FSAM communities, for Cora (left) and CiteSeer (right). Each point is a pair of classes. Pairs whose communities share more neurons tend to be confused more often (Pearson $r = 0.4312$ and $r = 0.3432$ respectively), a moderate association indicating that shared internal units are one source of class confusion.	64
3.7	Pairwise Jaccard similarity between class-selective neuron sets at layers 1–4 for Amazon Photo Dataset. Brighter cells indicate class pairs sharing more neurons. Similarity increases with depth across most class pairs, consistent with the progressive loss of class-specific structure reported for Cora in Figures 3.2–3.5.	65
3.8	Pairwise Jaccard similarity between class-selective neuron sets at layers 1–4 for CoauthorCs Dataset. Brighter cells indicate class pairs sharing more neurons. Similarity increases with depth across most class pairs, consistent with the progressive loss of class-specific structure reported for Cora in Figures 3.2–3.5.	66
3.9	Pairwise Jaccard similarity between class-selective neuron sets at layers 1–4 for the computer Dataset. Brighter cells indicate class pairs sharing more neurons. Similarity increases with depth across most class pairs, consistent with the progressive loss of class-specific structure reported for Cora in Figures 3.2–3.5.	67
4.1	Layer wise accuracy per class contribution of CoAuthorCs Dataset . .	82
4.2	Layer wise accuracy per class contribution of AmazonPhoto Dataset .	84
4.3	Layer wise accuracy per class contribution of Computers Dataset . . .	85
4.4	Per-class misclassification rates in each community between classes pair in layer 1	91
4.5	Per-class misclassification rates in each community between classes pair in layer 2	92

4.6	Per-class misclassification rates in each community between classes pair in layer 3	92
4.7	Per-class misclassification rates in each community between classes pair in layer 4	94
5.1	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 $B^{(\ell)} = \mathbb{I}\{H^{(\ell)} > \tau\}$; (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 ($\ell = 1, \dots, L$), producing per layer rule sets and masks. In $B^{(\ell)}$, rows are indexed by (graph ID, node ID) and ones mark active embedding components. Step (7) summarises the resulting insights on the model.	102
5.2	Comparison of the explanations on BA-2Motifs with GCN classifier	116

List of Tables

2.1	Comparison of GNN explainability methods by scope and supported functionality. Type (scope): instance level (local, per node/sample), model level (global), or hybrid (supports both). Classification: the method natively explains predictive tasks (e.g., node/graph classification) by attributing importance to features/subgraphs (✓ supported; ✗ not applicable/not primary). Knowledge extraction: produces explicit symbolic artefacts (e.g., rules, Boolean formulae, constraints). FSAM produces a structured model-level representation, namely class-associated neuron communities, rather than symbolic rules; the extraction step is contributed by ActiMine-GNN. Graph generation: synthesises exemplar/prototype graphs or subgraphs (beyond masking or extraction). Abbreviations (e.g., SA, CAM, LRP) follow the original papers; ticks indicate capability, not comparative quality.	40
2.2	Benchmark datasets used in this thesis. Task: NC = node classification, GC = graph classification. The example column states what an instance and its label represent.	41
2.3	Characterisation of neurosymbolic methods for GNN explainability. .	44
3.1	Distribution of different classes into communities for layer 2	69
4.1	Accuracy and Pearson Correlation (r) for depths (1–4) across three datasets.	77

4.2	Layerwise analysis with statistical validation.	78
4.3	Community structure and within-community misclassifications across layers. A within-community misclassification is a test node assigned to a different class where both classes belong to the same FSAM community at that layer.	93
5.1	Dataset characteristics and model performance metrics.	110
5.2	Per-layer rule extraction and local explanation metrics for layers L1–L3. “Avg.” denotes the mean over covered test graphs $\mathcal{S}^{(\ell)}$	112
5.3	Performance of explanation methods across different datasets	116

ABSTRACT

A Neuro-Symbolic Framework to Enhance Interpretability of Graph Neural Networks

Kislay Raj

Graph Neural Networks (GNNs) are now a standard tool for modelling graph-structured data in applications such as molecular property prediction, drug discovery, recommender systems, and citation networks. In high-stakes domains, especially computational biology and chemoinformatics, practitioners increasingly rely on GNNs for decision-making on relational data. However, despite their strong predictive performance, they still face several important challenges. It is still difficult to understand how GNNs organise information across layers, how depth affects the quality of the learned representations, and how individual decisions can be explained in human terms. The work in this thesis is driven by the central idea that combining activation-centred analysis with rule-based explanations can be the key to more transparent decision-making. The first challenge concerns the explainability of GNNs. As with other neural network-based models, they suffer from the black-box problem, making it difficult to understand the processes that drive their predictions. In this thesis, we address part of this challenge by focusing on graph convolutional network architecture on benchmark graph datasets, primarily in node classification settings. We introduce Functional Semantic Activation Mapping, a representation that acts as an intermediate layer between the knowledge encoded in a trained

GNN and a human-interpretable semantic structure. Given a trained GNN, we first show how a semantic activation graph can be constructed and used to obtain global insights into the organisation of classes, the appearance and loss of class-specific structure with depth, and the relationship between activation communities and observed patterns of class confusion. This analysis is then extended to examine depth, oversmoothing, and their effect on discriminative power. Building on the same activation extraction method, we introduce a semantic activation-mining approach that extracts compact rule-based explanations from binarised hidden activations, producing small sets of activation rules together with local subgraph rationales for individual predictions. Finally, the thesis outlines several research directions for neurosymbolic explainability in graph-based learning systems.

Chapter 1

Introduction

Explainable Artificial Intelligence (XAI) is the research area that aims to bring transparency to AI decision-making processes [21]. With the success of deep learning and the desire to deploy models in high-stakes settings, the need to account for decisions has become extremely important [2]. Providing reliable explanations is essential for trust, auditability, and informed human oversight, especially in domains such as healthcare, finance, and public services. However, the black-box nature of neural networks makes their decision making process an open challenge.

The desire to understand what happens inside neural networks is not new. Researchers have been trying to make model predictions more interpretable long before today’s deep architectures became possible. However, recent years have seen a surge of techniques that approach the problem from different angles with feature attributions, example based reasoning, counterfactuals, and surrogate models, each giving some insight into trained networks [19]. Despite this progress, many approaches derive explanations strictly within the same representational space used for training, and rarely draw upon external, human-understandable knowledge. As a result, explanations can lack semantic clarity for expert audiences and struggle to convey global model behaviour.

Graph Neural Networks (GNNs) extend deep learning to relational data and have shown strong performance in tasks such as node and graph classification, as well as link prediction [26, 56, 73]. Their ability to use topology and attributes makes them

powerful tools for analysing complex graphs such as citation networks, protein-to-protein interactions, and financial graphs. However, the reason GNNs work well is that they learn complex patterns in graph structures, which makes it difficult to see how they make decisions. Although operating on input data that explicitly models relationships might suggest improved interpretability, understanding *how* such relationships are represented within and between layers remains ambiguous [74].

In the current literature, most explanation methods for GNNs use semantics in the input graph. They are *local*, which means they extract a small subgraph or identify prominent nodes and edges in the input data that contribute to a specific prediction [75]. These methods are useful, but typically do not expose the internal information flow across layers, nor do they provide a *global* view of model behaviour. Recent surveys [76, 7] also highlighted that there is still a lack of explanations at the dataset-level and the absence of standardised quantitative protocols that jointly assess the fidelity, compactness, and stability of the explanation

In this thesis, we take a neurosymbolic view and argue that combining neural representations with symbolic structures such as rules, constraints, and concepts can support more transparent decision-making. Based on these research gaps, we follow two complementary directions. First, we develop a global, layer-wise analysis of internal representations, which is used as a diagnostic to quantify neuron–neuron dependencies, class–neuron associations, and how the behaviour of the model changes with depth and different layers. Second, we seek to derive human-readable rules that summarise a model’s behaviour at the dataset level and can also be instantiated for particular inputs; in other words, the variables in a rule are bound to the concepts or substructures present in a given instance to produce a concrete, per-instance explanation.

The first line of work leads to *Functional Semantic Activation Mapping* (FSAM) [44], which we use to characterise model behaviour across depth. For each layer, FSAM constructs a semantic activation graph whose vertices correspond to neurons and whose edges capture statistically significant co-activations. We use the term *seman-*

tic in a model-internal sense. A neuron community is semantic if its co-activation structure aligns with the model’s output classes or with recurring input substructures, as measured on held-out data. Accordingly, *semantic coherence* denotes the degree of this alignment, quantified by community class purity under the evaluation protocol of Section 2.1.6, and a *concept* denotes a recurring, class-associated activation pattern rather than a human-defined category. Throughout, evidence for semantic structure is quantitative alignment of this kind, reported alongside fidelity, stability, and coverage; Visual inspection is secondary and illustrative. By comparing these graphs across layers, FSAM can indicate where class or concept-specific structure appears, merges, or dissipates. Then, by using the FSAM semantic graph, we investigate the central question in GNN design. To what extent does increasing depth enhance meaningful internal structure, and does higher predictive accuracy necessarily correspond to more faithful or more discriminative representations? We distinguish semantic collapse from two related notions. Oversmoothing is a well documented property of embedding geometry, in which node embeddings become increasingly homogeneous with depth [62, 28, 45]. Semantic collapse, as defined in this thesis, is the loss of class-specific structure in the co-activation organisation of neurons: FSAM communities that were aligned with particular classes, merge or lose purity, measured by community class purity and a between/within-community separability index. The two are related, since homogeneous embeddings will eventually erode co-activation structure, but they are different measurements. As FSAM signals are computed from statistical dependencies among neuron activations and are localised to particular layers and communities, so they indicate where class specificity degrades rather than only that the embeddings have become similar on average. We therefore do not seek to re-establish oversmoothing; we provide a layer-wise, community-level indicator of when and where class specificity is lost. Semantic collapse is also distinct from changes in accuracy: in Chapter 4 we find cases in which accuracy plateaus or improves while FSAM class-specificity signals deteriorate, indicating that a model can be correct for reasons that are not semantically

well-grounded and accuracy alone is an insufficient measure of representation quality.

While FSAM offers a global, layer-wise description of behaviour, it does not directly produce the human-readable rules. To address this gap, in chapter 5 we introduce ActiMine-GNN, a framework that turns hidden layer activity into compact interpretable rules. It works on the same concept of activation extraction method as FSAM, but it fits shallow decision trees over these pattern features to obtain succinct rule sets that apply at the dataset level but can also be instantiated for individual graphs. In contrast to instance-level surrogate methods such as GNNExplainer [69], PGExplainer [31] and PGM-Explainer [58], which typically learn a separate interpretable model or mask for each target node or neighbourhood (so that the number of surrogates grows with the number of instances to be explained), ActiMine-GNN maintains a small collection of *global* surrogates (for example, one shallow tree per layer or a single tree over concatenated features) and caps the number of mined patterns (top- M per layer) evaluate the resulting rules in terms of fidelity at matched sparsity, rule coverage, and agreement between the surrogate and the underlying GNN, alongside standard task accuracy.

In this introductory chapter, we first set out the core background and concepts in Explainable GNN. We then motivate our study through three areas that organise the remainder of the chapter and frame our research questions: **(1) Expansion in Graph Neural Networks** rapid advances in architectures and applications increase the need to understand what is being learned at the scale; **(2) Explanations in Graph Neural Networks** current methods are predominantly local, with limited layer wise and global level explanation methods and **(3) Neuro-Symbolic AI** combining neural representations with symbolic structures (rules, constraints, concepts) offers a path to transparent, human understandable and transferable explanations. These three areas are central to our aims: the growth of GNNs creates the interpretability demand; the limitations of existing GNN explanations expose the methodological gap we address; and the neurosymbolic perspective provides the

organising principles for the rule-based and layer-wise analyses we develop.

1.1 Introduction to Explainable AI

With recent advances in Artificial Intelligence (AI), XAI seeks to make the decision making processes of AI systems transparent to human audiences. As modern AI systems are increasingly deployed in high stakes settings, explainability is essential for trust, accountability, and informed oversight. In this thesis, we focus on **opening the black box** of trained models, rather than designing inherently interpretable architectures, by analysing what a model’s *internal activations* reveal about its decision process. Although GNNs achieve strong performance, their interpretability lags behind models used in image and text domains. Local explainability methods account for deep model behaviour by identifying the input features that influence an outcome. We group these methods by how they assign importance scores: **Gradient/feature-based methods** adapt techniques from image and text, using gradients or hidden feature-map values to calculate feature significance [41]. **Perturbation methods** learn a graph mask by observing changes in predictions when input graphs are perturbed [69, 31]. **Decomposition methods** break down the prediction score into contributions from neurons in the final hidden layer, then propagate these contributions back to the input space [49, 41]. **Surrogate methods** explain an input graph by sampling its neighbourhood and training an interpretable model to approximate the predictions [24, 58]. Global explainability is less explored. **XGNN** [72] trains a graph generator to maximise the predicted probability for a class and uses the generated graphs to explain that class. However, it assumes each class can be explained by a single graph, which oversimplifies many real-world phenomena. Most methods either explain the GNN’s final decision or construct a graph for a specific decision. Focusing only on the final decision is not enough to understand model behaviour. To gain deeper insight into GNNs, we should examine both the model’s outputs and the intermediate layers where representation spaces are formed. Techniques developed for image and text are not directly applicable to

GNNs because graphs are not grid-like [75]. A large body of XAI research provides *feature-attribution* explanations (for example, saliency, masking, and local surrogates). These methods are useful, but they tend to target single predictions and the final decision surface. As a result, they offer limited visibility into the *layer-wise* construction of representations and the *dataset-level* regularities that drive behaviour. Recent surveys of GNN explainability and neurosymbolic methods point to these gaps and call for approaches that connect internal signals to human-interpretable structure, with standardised evaluation of fidelity, sparsity, stability, and coverage. We draw inspiration from studies in Convolutional Neural Networks (CNNs) that use graph analysis [23]. Our perspective is to study explanations *through neuron activations*. Prior work in computer vision shows that **co-activation graphs** [23], whose nodes are neurons and whose edges encode statistical dependencies between their activations, can act as functional representations of a network. This supports global analysis (for example, community structure and class similarity) and local, human-readable explanations when linked to external knowledge. These methods compute layer-wise activation statistics, threshold edges, and then analyse graph structure. They also show how to connect inputs and classes to the activation graph to obtain factual and counterfactual explanations. We follow this line of work in shaping our activation-centred approach. Explainability still presents open challenges that hinder trustworthy deployment [59]. These include a lack of human-understandable reasoning [68] and the absence of standardised evaluation protocols [47]. Global explainability that captures model behaviour across datasets remains less explored than local-level methods. We address two open issues highlighted by recent surveys [46, 29]: (i) the scarcity of global, dataset-level GNN explanations and (ii) the limited availability of quantitative evaluation methods. Together, these activation-based techniques align with survey calls for methods that bridge internal representations and symbolic structure, while supporting rigorous and comparable evaluation.

1.2 Introduction to GNN

A graph has a variable number of nodes, each with a variable number of neighbours, which breaks the locality and weight sharing assumptions of standard convolutions. GNNs [26, 56] address this through message passing. At each layer, a node aggregates information from its neighbours and updates its representation using a learnable, permutation invariant function. By stacking layers, information flows over longer ranges. This allows GNNs to combine topology with node and edge attributes and to support common tasks such as node classification, link prediction, and graph classification in domains including citation networks, recommender systems, and molecular discovery [27].

GNN’s focus was largely on how to combine information from a node and its neighbours efficiently, a process that later became known as message passing. Building on these early ideas, a range of GNN architectures have been proposed, each refining the message-passing process in different ways. In what follows, we briefly review some of the most widely used architectures below:

Graph Convolutional Networks (GCNs): The GCN model [26], simplified earlier spectral-based graph convolutions to propose an efficient, layer wise propagation rule for semi supervised node classification. Its forward model takes the form:

$$\mathbf{Z} = f(\mathbf{X}, \mathbf{A}) = \text{softmax} \left(\hat{\mathbf{A}} \text{ReLU} \left(\hat{\mathbf{A}} \mathbf{X} \mathbf{W}^{(0)} \right) \mathbf{W}^{(1)} \right)$$

where $\mathbf{X}$ is the node feature matrix, $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are trainable weight matrices, and $\hat{\mathbf{A}}$ is the adjacency matrix with self-loops, suitably normalised. At each layer, features are linearly transformed and then averaged over a node’s neighbours according to $\hat{\mathbf{A}}$, followed by a nonlinearity, producing smooth, locally aggregated node representations.

GraphSAGE: GraphSAGE [22] was proposed to address a key limitation of early GCNs, namely their largely transductive setting. Rather than relying on the full

graph during training and inference, GraphSAGE learns inductive node representations by sampling a fixed-size neighbourhood and aggregating the features of these neighbours. Several aggregation functions were introduced (mean, LSTM, pooling), with the pooling aggregator in particular offering a good balance between performance and computational cost. A limitation of the original design is that neighbourhood sampling is uniform, implicitly treating all neighbours as equally informative, which may not hold in graphs where certain connections are more relevant than others.

Graph Attention Networks (GATs): The GAT model [56] introduced an attention mechanism to address the limitations of GCN and GraphSAGE. It computes hidden representations of each node by attending over its neighbours, assigning a computed importance weight to each. This allows for a more expressive aggregation and provides a degree of model interpretability. A noted limitation was its initial inability to incorporate edge features.

Graph Isomorphism Network (GIN): The GIN model [64] provided a theoretical foundation for analysing the expressive power of GNNs. It proved that GCN and GraphSAGE were incapable of distinguishing certain graph structures due to their non-injective aggregation functions. GIN proposed a simple yet powerful architecture using Multi-Layer Perceptrons (MLPs) and a sum aggregator to model injective functions over multisets, making it as powerful as the Weisfeiler-Lehman test in discriminating graph structures. Training uses familiar components such as nonlinearities, normalisation, residual connections, and regularisation. As depth increases, oversmoothing, where node representations become indistinguishable where many distant signals are compressed into a fixed size vector, can arise. These effects motivate our later analyses of layer wise signals.

As shown in Fig. 1.1, a message passing layer updates the target node A by aggregating messages from its 1-hop neighbours (B, D, E), which themselves may already encode information from their own neighbours (e.g., B from C). After T layers, A 's representation depends on its T -hop neighbourhood. This has direct consequences

for explainability: salient evidence can be *multi-hop*. In deeper GNNs, signals become mixed, a phenomenon known as oversmoothing. Consequently, instance level masks typically access only nearby edges (two to three hops), which is insufficient to capture the network’s global behaviour or to reveal its decision making process.

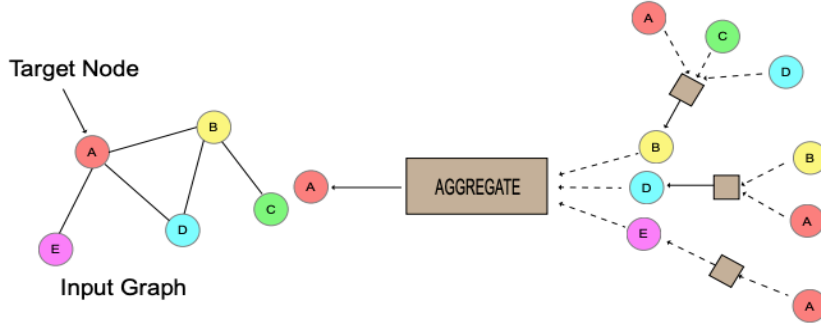


Figure 1.1: An example of a message passing architecture

This thesis develops post-hoc, activation-based explanation methods for message-passing GNNs. All experiments use GCN encoders [26] with ReLU activations and mean or sum aggregation. FSAM and the depth analysis (Chapters 3 and 4) are evaluated on node classification benchmarks, and ActiMine-GNN (Chapter 5) on graph classification benchmarks. Throughout, we analyse trained models through their hidden activations. We do not propose inherently interpretable architectures, and we do not modify training.

The methods assume only access to the hidden activations of a fixed trained model. They make no further architecture-specific assumptions, so they should in principle transfer to other message-passing encoders such as GraphSAGE [22], GAT [56], and GIN [64]. We state this as an expectation rather than a finding: all empirical claims in this thesis are confined to GCN encoders on the tasks and datasets described in Section 2.1.5, and validation on other architectures is left to future work.

Explanations for GNNs, and how our approach relates to existing work, are covered in section 2.1.3.

1.3 Neuro-Symbolic Background for XAI

Neuro-symbolic(NeSy) integration is a promising route to bring symbolic knowledge to deep models while keeping them explainable [15, 9]. The aim is to combine standard symbolic reasoning with neural learning so that accuracy is preserved and behaviour can be read in human terms. A standard design treats logical rules as constraints in a vector space. In practice, these constraints are added to the loss as regularisation terms that penalise rule violations and encourage rule-consistent predictions, without changing gradient-based training. A related line makes logic differentiable, allowing truth values and operators to be optimised with the model’s parameters. A complementary, post-hoc takes the model’s internal activations and turns them into symbolic objects such as rules, concepts, or relations. Together, these approaches keep explanations grounded in the learned behaviour of the model while expressing them in a form that people can read and reason about, without changing how the underlying network is trained. An earlier research, *explanatory graphs* [77], builds a graph from CNNs by disentangling object-part patterns per convolutional filter and encoding their spatial relations. This representation reveals which patterns tend to co-activate and which filters detect them. However, the resulting graph remains layered, acyclic, and bipartite, a dictionary of parts rather than a functional network, so it is not well-suited to standard graph theoretic analysis or metrics.

In [16], the authors introduce a graph built from vector embeddings that places data instances and convolutional filters in the same structure. Preliminary results show that community detection on this graph can recover clusters of images from topology alone. A limitation remains when restricted to hidden-layer units and their links; the graph is still layered, acyclic, directed, and bipartite, which constrains the use of standard graph-theoretic methods.

Another alternative, the *relational graph* [70], is built through a message-passing scheme. Unlike the computational graph, the resulting structure is neither layered, acyclic, directed, nor bipartite. However, its purpose is architectural analysis that

compares predictive performance between deep models rather than producing global or local post hoc explanations.

Further research on *co-activation graphs* [23] constructs a graph from CNNs by treating neurons as nodes and connecting pairs whose activations are statistically dependent across a dataset (for example, via rank correlations). We draw inspiration from this line of work, but adapt the idea to GNN instead of focusing on neurons in a convolutional layer. We analyse patterns of activation across nodes and layers in a GNN, capturing how information is shared and propagated over the underlying graph.

Another research learns a *decision tree* from a pre-trained CNN to provide semantic reasons for categorisation [78]. The pipeline first trains disentangled filters in the top convolutional layer so that each filter aligns with a single concept; activations from these filters are then fed to a tree that indicates which parts activate which filters and by how much [13]. This produces readable, part-based rationales, but introduces dependencies on disentanglement losses and layer-specific design. Our approach takes the positive lesson that symbolic surrogates add auditability while addressing these gaps in ActiMine–GNN mines recurrent activation patterns directly (no disentanglement or external concept labels), induces compact rules with dataset-level coverage, and supports per-instance instantiation without restricting explanations to a single domain.

A NeSy approach is helpful because it keeps explanations model-grounded (derived from internal behaviour) and human-centred (expressed as rules, concepts, or relations). It also supports an iterative cycle of extraction, validation against domain knowledge, explanation and feedback, as advocated in recent work on neurosymbolic explainability for high-stakes settings [33]. Activation-centred analysis shows what the network represents and where those representations form or dissipate across layers, before any task-specific tooling. This makes it natural to (i) diagnose behaviour (e.g., loss of class specificity with depth) and (ii) convert recurrent activation patterns into human-readable artefacts that experts can audit—steps we later carry

over.

This thesis follows that pattern. In Chapter 3, we use activation signals to obtain a global, layer-wise view of the model (FSAM). Chapter 4 then extends this analysis to study how representations evolve and collapse across layers. In Chapter 5, we distil recurrent activation patterns into compact rules with per-instance instantiation (ActiMine-GNN). The state of the art is discussed in Chapter 2, where we set out the background and motivation for this work.

1.4 Hypotheses and Research Questions

The first phase of this research investigates whether it is feasible to construct a faithful model-level representation for GNNs by capturing statistical dependencies among neural activation layers. Consequently, we state our first hypothesis in our research as follows.

Hypothesis 1 (H1). The knowledge encoded by a trained GNN can be captured by extracting statistically significant co-activations among neurons and representing them, per layer, as a Functional semantic activation graph (FSAM); the resulting communities align with output classes or coherent concepts and expose global model behaviour.

To guide the assessment of H1, we consider the following research questions:

- **RQ1: Semantic Activation Graph — Definition and Construction.**
How should FSAM nodes and edges be derived from layer activations, and what significance criteria produce a reproducible and computationally tractable construction?
- **RQ2: Validating the FSAM Representation** This question examines how closely FSAM communities at each layer align with the model’s output classes, and how this alignment changes as we move deeper into the network.
- **RQ3: Interpreting Class Confusions through Community Detection**

and Class–Pair Similarity. Here, we examine whether pairs of classes that the model tends to confuse can be understood in terms of overlap or proximity between their FSAM communities.

The second phase focuses on how increasing depth affects these semantic structures, with particular attention to oversmoothing and discriminative power. This leads to our second hypothesis:

Hypothesis 2 (H2). Increasing the depth of the GNN beyond a task dependent optimum induces semantic collapse: neuron co-activations become more homogeneous, class-specific discriminativeness diminishes, and class–pair confusions rise even when headline accuracy plateaus or improves.

Two research questions were used to assess H2:

- **RQ4: Depth-wise Behaviour and Discriminative Power.** How do accuracy, class specificity, and FSAM community coherence change as depth increases, and at what depth does discriminative power begin to deteriorate relative to accuracy trends?
- **RQ5: Community Overlap vs. Class Confusion/Similarity.** Does increased inter-class community overlap with depth track observed rises in class–pair confusion/similarity across datasets and random initialisations?

A third phase in this thesis focuses on mining compact, human-readable activation rules from hidden-layer co-activations in order to provide a quantitative and actionable account of instance-level decisions in GNNs. The third hypothesis is formulated as follows:

Hypothesis 3 (H3). Given a trained GNN, we hypothesise that it is possible to extract compact local activation rules from binarised hidden-layer activations. These rules should capture the factual evidence behind a prediction and support small, validity-preserving counterfactual changes that are sufficient to provide the model’s decision. Two research questions were used to assess H3:

- **RQ6:** How can we mine local activation rules from hidden-layer activations so that they remain compact, achieve high fidelity and align well with the underlying GNN?
- **RQ7:** How can visual inspection of instantiated rules provide clear factual explanations?

1.5 Thesis Structure

This thesis consists of the following six chapters:

(i) Introduction (Chapter 1). The present chapter outlines the motivation and context, introduces the main background on explainable AI, GNNs, and NeSy approaches, and presents the hypotheses and research questions that structure the remainder of the work.

(ii) Literature Review (Chapter 2). The chapter outlines the current state-of-the-art research in explainable GNN with a focus on: (a) global explanations; (b) local explanations; (c) graph-based explainability methods for GNNs; and (d) neuro-symbolic approaches that induce symbolic structure from neural representations. This chapter positions our contributions within the wider context and clarifies the evaluation criteria used throughout.

(iii) Global Explanations using Functional Semantic Activation Mapping (Chapter 3.) This chapter introduces the core methodology: the semantic activation graph constructed from statistically significant co-activations (FSAM). We formalise the representation and evaluate it for global and local explanations covering definition and construction, validation against output classes, interpretation of class confusions through community structure, and visual analysis across the layers.

(iv) Measuring Semantic Collapse: Depth, Oversmoothing, and Discriminative Power (Chapter 4). Building on the FSAM framework introduced in Chapter 3, this chapter investigates how increasing depth affects both the internal representation and the predictive behaviour of GNNs. We examine oversmoothing and semantic collapse in terms of FSAM signals.

(v) Local Rule-Based Explanations for GNNs: A Semantic Activation-Mining Approach (Chapter 5). This chapter introduces the ActiMine-GNN method. It is an activation mining method that works on the same FSAM framework introduced in Chapter 3 for the graph classification task. The technique turns hidden layer activity into compact, human-readable rules, and we illustrate how the resulting rules give counterfactual explanations at the instance level.

(vi) Conclusions (Chapter 6). The final chapter summarises the research by revisiting the hypotheses and research questions, reflecting on the limitations of the proposed methods, and outlining directions for future work.

Chapter 2

Literature Review

In this chapter, we review the state of the art in explainability for GNN, with a specific focus on the Graph Convolutional Network (GCN) architecture, aligning the scope with our activation-based neurosymbolic perspective. We begin with the most commonly used methods. First, we focus on instance-level explainability methods that explain individual predictions. At the instance level, we discuss the most recent state-of-the-art methods such as gradient-based methods, feature attribution methods, mask-learning approaches, surrogate models, decomposition-based models, and subgraph search techniques. We then move to global approaches. Here, we examine prototype-based and generator-based techniques, as well as methods for global rule tracing. Next, we review NeSy-based methods that operate directly on internal activations. These methods analyse how neurons behave across layers and relate them to classes or concepts. We then turn to neurosymbolic approaches for GNNs that extract rules, concepts, and relations from hidden layers. Taken together, these produce compact, human-readable summaries that describe model behaviour at the level of whole datasets. Finally, we consider methods that align internal representations with external knowledge, and we summarise the evaluation protocols used to compare explainers across the thesis.

2.1 GNN Explainability Methods

Explaining the decisions of GNNs is challenging as graphs are discrete and relational, so gradient-based visualisation heuristics from images or text can disrupt structure and mislead [62]. We organise the literature into four complementary families that aim to make behaviour readable: instance level methods (local method) 2.1.1, model level methods (global method) 2.1.2, rule-based approaches (intrinsic or post hoc), and concept-based approaches 2.1.4 for details. We added the Benchmarks and Datasets Section 2.1.5 used in this thesis and evaluated the methods using a small, consistent set of criteria in Section 2.1.6. A neurosymbolic (NeSy) perspective links internal model signals to human-readable structure and motivates both our activation-centred and rule-based focus [5]. Figure 2.1 presents a taxonomy that distinguishes between *factual* and *counterfactual* explanations (see Section 2.1.7). Factual explanations highlight features or subgraphs that support a given prediction, whereas counterfactual explanations describe small changes that would be sufficient to alter that prediction. In this chapter, we refer to a few representative methods to illustrate each category; detailed discussion and comparisons are deferred to the subsections indicated.

2.1.1 Instance level Explanation

Instance-level methods aim to explain deep models by identifying the important input features that are used for predicting the outcome. These methods focus on finding the patterns and features in the input graph that influence a given prediction. Instance-level explanations deal with individual examples. They offer detailed insight at a local level. In most cases, they highlight specific nodes or edges in the graph that are important for the prediction. For example, they may show why a particular node was assigned to a given class. They may also explain why a link is predicted between two nodes. In doing so, they reveal which local patterns influence the decision and how the model represents the neighbourhood relevant to that instance. This provides a detailed, case by case view of GNN behaviour.

We organise instance-level methods into four categories according to how they assign importance: gradients/features, perturbation-based, decomposition, and surrogate methods, as shown in Figure 2.1.

1) **Gradients/features:** Explainability in image and text tasks often relies on gradient and feature-based methods that use gradients and hidden feature maps to score input importance. In the GNN, it examines the gradients of the loss with respect to the input (or hidden maps) to indicate which features influence a specific instance. Various methods have been developed for explaining graph networks, including SA [4, 41], Guided BP [4], CAM [41] and Grad CAM [51]. These approaches differ in how they backpropagate gradients and combine hidden feature maps.

2) **Perturbations:** Perturbation-based methods monitor how the predictions of GNNs change when specific input features are modified. This allows us to identify the features that have an impact on the original prediction. By examining the impact of these perturbations, one can identify the features that were most influential in the original prediction. Various perturbation based methods are proposed, including GNNExplainer [69], PGExplainer [31], ZORRO [48], GraphMask [14], Causal Screening [60], and SubgraphX [75]. These approaches share a common template (mask generation, mask type, objective), but differ in the algorithms used and the fidelity or sparsity criteria. Choice of perturbation scale is critical as very small changes may not expose meaningful behaviour, while large changes can produce unrealistic inputs, which reduces reliability.

3) **Decomposition:** These methods decompose the GNN’s contribution to the prediction of a particular instance into contributions from different nodes, edges, or layers in the network. This decomposition can help to identify the most important components of the graph for that prediction. In recent times, various techniques have been developed to help understand deep graph neural networks, including Layerwise Relevance Propagation (LRP) [4, 50], Excitation BP [41] and GNN-LRP [49]. The core idea is to apply score decomposition rules that distribute the output back through the network. It is important to note that all the algorithms for deep graph

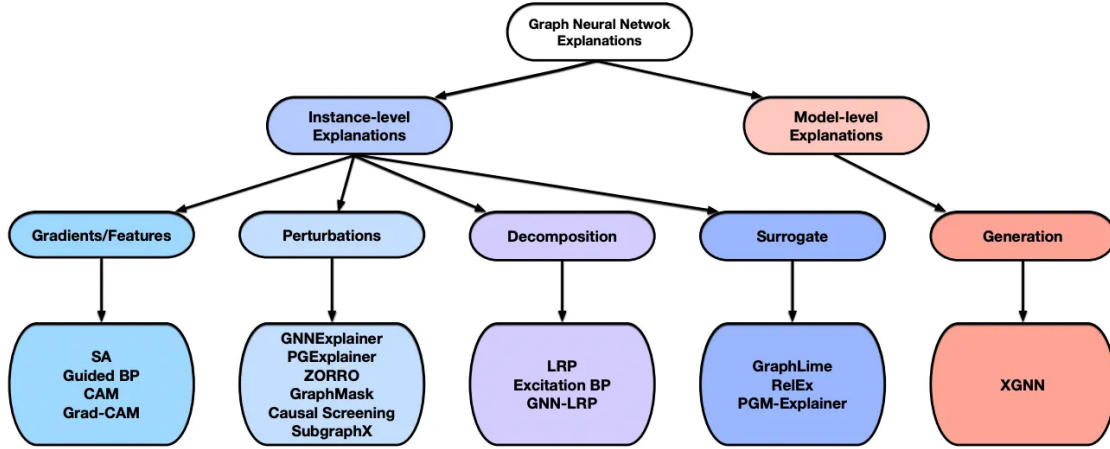


Figure 2.1: Overview of the taxonomy of GNN explainability methods [31]

models often linearise and ignore certain non-linearities; methods differ in their decomposition rules and explanation targets.

4) **Surrogate methods:** Surrogate approaches fit an interpretable model to approximate the GNN in a local neighbourhood of the instance. Typical pipelines collect nearby samples and their predictions, train a simple model, and use its parameters as the explanation. It is worth noting that these techniques assume that the associations within areas of the input example are less intricate and can be accurately represented by a surrogate model. Several recent state-of-the-art methods have been proposed to explain deep graph models using surrogate methods, including GraphLime [24], RelEx [80], and PGM Explainer [31], exemplifying this idea, differing in how they construct the local dataset and which surrogate they employ. Finally, the explanations derived from the model are considered as explanations for the input graph. However, these methods differ in two aspects: how they collect the dataset and which interpretable surrogate model they utilise.

Current methods for explaining GNNs usually work one instance at a time. They

highlight a small set of nodes, edges, or subgraphs that are said to justify a single prediction. While this is useful, it gives little insight into how internal representations are formed across layers or how the model behaves over an entire dataset. Mask-learning approaches are also sensitive to how the perturbation is defined and at what scale it is applied, which can make the resulting explanations unstable. Surrogate models trained locally around each instance may drift away from the behaviour of the base GNN, and their number tends to grow with the number of explanations required. Search procedures over large, sparse graphs can become computationally expensive. Global methods based on prototypes or generators make stronger assumptions about the class structure and may fail to capture multiple distinct decision pathways. In this thesis, we follow a different route by working directly with internal activations and extracting compact rule sets. FSAM provides a global view of the class specificity in layers and reveals where deeper models may be “right for the wrong reasons”, for example, when accuracy improves while the semantic structure deteriorates.

2.1.2 Model Level Explanation

Model-level explanations describe the overall decision logic of a GNN across a dataset. They offer a high-level view of behaviour rather than case-by-case evidence. Prototype-based methods, such as XGNN [72], synthesise graphs that trigger high confidence predictions and reveal class-specific structural motifs. Activation rule mining extracts symbolic rules from hidden layer activations and produces global, logic-style summaries of how the model makes decisions [44]. Such rules improve interpretability and can be checked against domain knowledge. Global views are useful, but they often rely on a small set of representative graphs and may miss rare decision pathways [37]. Figure 2.1 distinguishes factual from counterfactual explanations. The figure focuses on the nature of the explanation, while the rest of the chapter categorises methods by scope: instance level, model level, and rule based approaches. Hybrid methods combine intrinsic transparency with symbolic

reasoning, for example by pairing a transparent predictor with rule extraction.

Table 2.1 summarises representative GNN explainability methods by approach, focus, and use of rule based reasoning. Rule based reasoning helps to produce human readable explanations and to align predictions with domain knowledge. Some concept based methods also employ graph generation to provide global insight. Knowledge extraction methods aim to make the learned knowledge explicit and are closely related to rule and concept based techniques, since they produce explanations that experts can validate.

The derived graph patterns may not correspond directly to real-world entities, which makes explanations harder to interpret. Input optimisation has been used to explain image classifiers at the model level [38]. However, directly optimising graph structure is not straightforward, because the topology is discrete and there are additional validity constraints. Model-level views can also lack the granularity of instance-level methods. In addition, their apparent generality may not transfer well across different types of graphs. Further follow-up studies show that multiple valid substructures can drive the same label and may be overlooked if the explainer favours a single motif [3]. In chemistry, ring patterns illustrate the issue: building graphs edge by edge can struggle to capture motifs that require adding a set of edges together, which affects faithfulness for certain classes [71]. As a result, explanations may be biased towards one pattern and lack multimodal coverage. GLGExplainer [3] takes a different route. It aggregates local explanations into learned graphical concepts and then combines these into Boolean formulas. This helps to make class-level behaviour more readable. However, the method still has some limitations. It depends on local explanations and discrete clustering. It also lacks explicit semantic grounding. Together, these factors can reduce alignment with the model’s internal reasoning. These approaches focus on the final decision. They are also not designed to link internal structure to prior knowledge in a systematic way. In the remainder of this thesis, we address these gaps. We do so by combining activation-based, layer-wise analysis with compact rule extraction. This provides global coverage with clear ties

to internal signals and, when available, to domain knowledge. Despite their limitations, model-level explanations remain important. Model-level explanations are essential for understanding overall behaviour and for detecting biases or systematic errors. Most recent approaches rely on visual or interactive analysis and largely summarise what the final layer predicts. In addition, our approach extracts the structure from within the model. FSAM [43] builds class relationships from neuron co-activations and class associations, layer by layer, which allows a quantitative assessment of the semantic structure learned by the GNN. This makes it possible to locate where increasing depth improves or degrades class specificity.

2.1.3 Explanations in Graph Neural Networks

GNNs have become the principal approach for learning on relational data across domains such as social networks, molecular structures, and citation graphs, among others [67, 17, 63]. Building on operations such as graph convolution, attention, and pooling, they aggregate information from neighbourhoods and combine topology with node and edge attributes [26, 56, 73]. Despite these advances, explainability for graph models remains comparatively under-explored, which hinders adoption in settings that require transparency and explainability.

Explaining GNNs is challenging because of how information is organised in graphs. In images or text, data lie on a regular grid, and locality is easy to define; pixels are positioned next to other pixels, and words appear next to other words. Graphs are different as they are based on non-Euclidean space and are typically represented by a feature matrix and an adjacency matrix. In this setting, predictive signals rarely arise from a single node or a single feature in isolation. Instead, they emerge from patterns that combine both structure (who is connected to whom) and attributes (what nodes and edges represent). On top of this, message passing layers propagate information over several hops. As a result, nodes that are far apart in the original graph can still influence the prediction for a given target node, making the internal reasoning of GNNs harder to disentangle and explain. Methods effective in computer

vision, such as input optimisation or soft masking, do not transfer directly to graphs. Directly optimising the adjacency matrix would mean taking gradient-based steps on its entries. In many graph problems, the adjacency matrix is discrete: entries state whether an edge is present or not. For undirected graphs, this matrix must be symmetric, and in many cases, self-loops are disallowed, so not every pattern of entries defines a valid graph. When explainers use continuous relaxations (soft masks), they often introduce fractional edge weights. This can break those validity constraints, meaning the “explained” object is no longer a proper graph [38, 8]. A further difficulty is that nodes and edges act together. Outcomes depend on their joint contribution, so attributing importance to single features alone rarely captures the interaction effects that matter.

As with other deep neural networks, GNNs inherit the black-box problem; they can achieve high accuracy, but their internal decision-making processes are difficult to understand. Early work in GNN explainability adapted gradient- and activation-based techniques from convolutional networks to the message-passing setting, producing saliency maps over nodes, edges, and features (see Section 1.1). Structure-aware approaches then learned instance-specific masks to identify the smallest subgraphs supporting model predictions, with GNNExplainer being an important early work in this direction [69]. These limitations motivate the perspective taken in this thesis: to complement global, instance-level attributions with methods that characterise layer-wise behaviour and produce compact, readable summaries at the dataset level. Subsequent sections formalise this view, set out the evaluation criteria, and our hypotheses and research questions are stated in Section 1.4.

2.1.4 Neuro-Symbolic Approaches for GNN Explainability

NeSy approaches combine GNNs with symbolic reasoning to improve explainability. The aim is to produce logical, human readable artefacts while keeping the model’s predictive behaviour in view. Graph relations and subgraph motifs align naturally with rule preconditions, which motivates a NeSy focus for GNN explainability.

Rule based methods analyse trained models to derive sets of if-then rules that summarise decision logic. RelEx extracts relational clauses from observed behaviour [80]. GLGExplainer induces global logical formulas over learned concepts [3]. GraphTrail derives model level rule traces for a global view [1]. We assess rule quality by fidelity to the underlying GNN and by interpretability, measured through rule length, coverage, and expert judgement [55]. These pipelines are informative, yet scalability is one of the limitations as without pruning or targeted visualisation, complexity grows quickly [10].

Concept-based methods do not work directly on raw features. Instead, they link model predictions to concepts. These concepts can be defined by humans or discovered automatically. In graphs, concept bottleneck models first predict intermediate concepts, such as functional groups in molecules [5]. Graph CAV extends concept activation vectors to graph data. It identifies subgraph patterns whose presence or absence affects the model’s output [37]. By grounding explanations in domain concepts, experts can more easily check and validate model behaviour. These concept-level explanations can also guide the construction of downstream rules.

Other NeSy variants inject structure during training or operate on activations. Logic guided training and knowledge graph integration impose constraints that steer what the model learns [69]. Frequent activation patterns can be mined and summarised as short clauses [49]. Layer-wise diagnostic mapping FSAM is one the method that reveal semantic structure and informs later rule extraction. Intrinsic masking frameworks, for example INSIDE-GNN, improve traceability even when they do not output explicit rules [57].

In this thesis, we use both lines. FSAM provides a layer-wise, activation centred characterisation. ActiMine-GNN extracts compact rule sets from hidden activity. Together, they offer global coverage with per instance detail and allow alignment with domain knowledge when available. These are the following NeSy approaches in brief

1. Rule extraction: Deriving human-readable if-then rules from a trained GNN,

either by analysing its behaviour (RelEx [80]) or by distilling its logic (neuro-symbolic distillation [15]).

2. Knowledge graph integration: Injecting constraints or relations from an external knowledge graph during training to shape the model’s representations in a semantically meaningful way (Logic Guided GNNs [20]).
3. Concept-based explanations: Linking predictions to human-defined or automatically discovered concepts, such as functional groups in molecules, rather than low-level features, to support expert validation [12, 40].

Table 2.1 situates representative methods by scope and functionality and aligns with the neurosymbolic lines outlined in Section 2.1.4. Rule based approaches appear with explicit knowledge extraction; concept oriented methods surface where learned concepts are composed into logic. Generator based global methods are marked in the graph generation column.

2.1.5 Benchmarks and Datasets

The development and fair comparison of GNN explainability methods rely on standardised benchmarks and datasets. These are often synthetic or real-world datasets where the ground truth reasons for a prediction are known or can be plausibly inferred.

Table 2.2 summarises the benchmarks used in this thesis. For the synthetic benchmarks, the attached motif (a house, 6-node cycle, or 3×3 grid) is the known ground-truth explanation, which allows explanation quality to be checked against a known cause.

For the synthetic datasets we report motif recovery in addition to fidelity at matched sparsity. Node features are bag-of-words or TF-IDF vectors for the citation and e-commerce datasets, keyword vectors for Coauthor CS, and atom or bond types for the molecular datasets; BA-Shapes has no node features. AIDS is class-imbalanced. Across all benchmarks we use the standard splits and preprocessing.

Table 2.1: Comparison of GNN explainability methods by scope and supported functionality. Type (scope): instance level (local, per node/sample), model level (global), or hybrid (supports both). Classification: the method natively explains predictive tasks (e.g., node/graph classification) by attributing importance to features/-subgraphs (✓ supported; ✗ not applicable/not primary). Knowledge extraction: produces explicit symbolic artefacts (e.g., rules, Boolean formulae, constraints). FSAM produces a structured model-level representation, namely class-associated neuron communities, rather than symbolic rules; the extraction step is contributed by ActiMine-GNN. Graph generation: synthesises exemplar/prototype graphs or subgraphs (beyond masking or extraction). Abbreviations (e.g., SA, CAM, LRP) follow the original papers; ticks indicate capability, not comparative quality.

Method	Type	Classification	Knowledge ex- traction	Graph gen- eration
SA [4, 41]	Instance	✓	✗	✗
Guided BP [4]	Instance	✓	✗	✗
CAM [41]	Instance	✓	✗	✗
Grad-CAM [51]	Instance	✓	✗	✗
GNNExplainer [69]	Instance	✓	✗	✗
PGExplainer [31]	Instance	✓	✗	✗
GraphMask [48]	Instance	✓	✗	✗
ZORRO [14]	Instance	✓	✗	✗
Causal Screening [60]	Instance	✓	✗	✗
SubgraphX [75]	Instance	✓	✗	✗
LRP [50]	Instance	✓	✗	✗
Excitation BP [41]	Instance	✓	✗	✗
GNN-LRP [49]	Instance	✓	✗	✗
GraphLIME [24]	Instance	✓	✗	✗
RelEx [80]	Instance	✓	✓	✗
PGM-Explainer [58]	Instance	✓	✗	✗
INSIDE-GNN [57]	Hybrid	✓	✗	✗
FSAM (ours) [44]	Model	✓	✗	✗
Logic-Guided [20]	Hybrid	✓	✓	✗
GLGExplainer [3]	Model	✓	✓	✗
GraphTrail [1]	Model	✓	✓	✗
XGNN [72]	Model	✓	✗	✓
ActiMine-GNN (ours) [42]	Hybrid	✓	✓	✗

2.1.6 Evaluation Metrics and Protocols

To compare methods fairly, we use a small, consistent set of metrics. These reflect both effectiveness and interpretability for neurosymbolic explainability in GNNs. Below, we give concise definitions and explain how they are used. Fidelity measures the agreement between an explanation (or rule set) and the original GNN. For instance-level methods, this is often the accuracy of a surrogate or masked graph

Table 2.2: Benchmark datasets used in this thesis. Task: NC = node classification, GC = graph classification. The example column states what an instance and its label represent.

Dataset	Domain	Task	Classes	Example (nodes / edges / label)
Cora [52]	Citation	NC	7	Papers linked by citations; label is the paper’s topic, e.g. <i>Neural Networks</i>
CiteSeer [52]	Citation	NC	6	Papers linked by citations; label is the research area
PubMed [36]	Biomedical citation	NC	3	Diabetes publications; label is the diabetes type discussed
Amazon Computers [32]	E-commerce	NC	10	Products linked by co-purchase; label is the product category
Amazon Photos [32]	E-commerce	NC	8	Products linked by co-purchase; label is the product category
Coauthor CS [53]	Co-authorship	NC	15	Authors linked by joint papers; label is the primary research field
BA-Shapes [69]	Synthetic	NC	4	BA base graph with attached house motifs; label encodes a node’s position in or outside a motif
BA-Community [69]	Synthetic	NC	8	Two BA-Shapes communities with node features; label combines community and motif position
Tree-Cycle [54]	Synthetic	NC	2	Binary tree with attached 6-node cycles; label is tree vs cycle membership
Tree-Grids [54]	Synthetic	NC	2	Binary tree with attached 9-node grids; label is tree vs grid membership
BA-2Motifs [31]	Synthetic	GC	2	Each graph contains a house or a 5-cycle; label is which motif is present
MUTAG [34]	Molecular	GC	2	Atoms and bonds; label is mutagenic effect on a bacterium
AIDS [34]	Molecular	GC	2	Atoms and bonds; label is anti-HIV activity
BBBP [61]	Molecular	GC	2	Atoms and bonds; label is blood-brain-barrier permeability

relative to the base model [69, 31]. For rule extraction, it is the fraction of GNN predictions reproduced by the distilled rules [18].

Sparsity is the compactness of an explanation. In local masks, this is the number of nodes, edges, or features retained; in rule sets, it is the total rule count (and related size measures). Sparse explanations aid human reading but must be balanced against fidelity [69, 48].

The complexity of the rules is a proxy for the interpretability of logic-based outputs. Typical indicators include the number of predicates, the average length of the rule, the depth of the tree, and the total number of rules. Lower complexity is generally easier to audit [55].

Concept completeness and purity apply to concept-based methods: completeness measures how well discovered concepts span the model’s decision space, and purity measures the semantic coherence within each concept [37]. High completeness and purity indicate that concepts capture the underlying factors cleanly. Prototype faithfulness applies to generator-based global methods and measures how representative the synthesised graphs are for a target class. A standard test is the confidence drop when real inputs are replaced by prototypes in the inference pipeline [72]. Stability is the robustness of explanations under small input perturbations. Stable methods are expected to return similar explanations for similar inputs, which is an important requirement for trust [4]. In our work, we report these metrics at matched sparsity where this is meaningful, and we present both instance-level and model-level results. When a method produces rules or concepts, we also report their complexity and coverage, alongside fidelity. However, these evaluation criteria have limitations. At present, there is no single benchmark that brings together, in a unified setting, local and global fidelity, rule complexity, concept coherence, prototype quality, stability, and coverage. As a result, comparing different approaches across studies remains difficult and often requires case-by-case interpretation.

We apply these metrics to the datasets described in Section 2.1.5. Our evaluation covers several types of graphs:

- **Citation networks:** Cora, CiteSeer, PubMed.
- **Coauthorship graphs:** Coauthor CS.
- **E-commerce graphs:** Amazon Computers, Amazon Photos.
- **Synthetic motif benchmarks:** BA-Shapes, BA-Community, Tree-Cycle, Tree-Grids, BA-2Motifs.

- **Molecular graphs:** MUTAG, AIDS, BBBP.

For the synthetic datasets, we report not only fidelity at a matched sparsity level, but also how well ground truth motifs are recovered. For the citation, coauthorship, and e-commerce graphs, we focus on fidelity, sparsity, stability, and coverage. For the molecular datasets, we additionally consider rule complexity and, where relevant, simple concept checks. Preprocessing follows the library defaults in all cases; the data splits used for each dataset are specified in Section 3.3.1.

2.1.7 Taxonomy of Neurosymbolic Methods

We classify NeSy approaches by how symbolic information enters the pipeline, the explanatory objective, and the graph settings they target. The focus is on methods that produce, consume, or constrain symbolic structure; purely neural attributions are treated as baselines and are not part of this taxonomy.

In the Integration mechanisms we consider rule activation mining, rule extraction, knowledge graph integration, and hybrid reasoning with differentiable logic (for example, Logic Tensor Networks) [57]. Emerging lines include privacy-preserving extraction and temporal rule induction. As explanatory objectives, common goals are fidelity, robustness, sparsity, and user trust. These objectives guide metric choice and are evaluated under the protocol in Section 2.1.6. As applicability differs in the graph types they address as homogeneous, heterogeneous, attributed, and temporal graphs [25]. Temporal graphs introduce additional challenges, since explanations must track change over time.

Two trade-offs organise the ratings in Table 2.3. The first concerns integration depth. Methods that inject symbolic structure during training (knowledge graph integration, hybrid reasoning) tend towards higher robustness and rule-consistent behaviour, but they require external knowledge to exist for the domain and they alter the training procedure, which limits where they can be applied. Post-hoc extraction methods apply to any trained model without modification, but their fidelity is bounded by how well a compact surrogate can reproduce the model’s behaviour.

Table 2.3: Characterisation of neurosymbolic methods for GNN explainability.

Method Category	Example	Mechanism	Metrics	Use Cases	Graph Types
Rule Activation Mining	Inside [57]	FORSIED activation mining	Fidelity: High; Sparsity: Moderate	Biomedicine; Fraud Detection	Homogeneous; Attributed
Rule Extraction	GNNExp [69] + Distil [40]	Subgraph saliency + clauses	Fidelity: High; User Trust: High	Recommenders; Social Networks	Homogeneous; Attributed
KG Integration	Logic-Guided GNN [20]	Datalog regularisers	Fidelity: High; Robustness: Moderate	Biomedicine; KGs	Heterogeneous
Hybrid Reasoning	Logic Tensor Networks [57]	Differentiable constraints	Fidelity: High; User Trust: High	Relational reasoning	Heterogeneous
Attention-Based Rule Selection	RuleFormer-GNN [65]	Attention-based rule selection	User Trust: High; Robustness: Moderate	Social Networks	Homogeneous; Temporal
Privacy-Preserving Extraction	Differential Privacy Rules [39]	Privacy-guaranteed rule mining	Fidelity: Moderate; User Trust: High	Healthcare	Attributed
Temporal Rule Induction	Temporal Rule Mining [30]	Rules over evolving graphs	Robustness: Low; Sparsity: Moderate	Traffic; Finance	Temporal

The second trade-off is between fidelity and compactness within extraction: larger rule sets track the model more closely, while smaller ones are easier to audit, which is why the evaluation protocol of Section 2.1.6 compares fidelity at matched sparsity. The specialised categories trade along the same axes under deployment constraints: privacy-preserving extraction sacrifices some fidelity for formal guarantees, and temporal rule induction sacrifices robustness for the ability to track evolving graphs. The ratings in the table are qualitative tendencies reported in the cited works rather than measured values on a common benchmark; no such benchmark currently exists, as noted in Section 2.1.6. Within this space, FSAM and ActiMine-GNN occupy the post-hoc, compact-rule position: model-agnostic extraction, with the fidelity compactness balance controlled explicitly.

In this thesis, we concentrate on two such integration points. First, FSAM performs activation mining. It helps us to understand the activation behaviour of the hidden layers. Secondly, we extend the same method to the graph classification task, which gives us ActiMine-GNN. ActiMine-GNN performs rule extraction and converts these patterns into compact logical clauses or small global surrogate models.

The metrics column follows the criteria introduced in Section 2.1.6, and the datasets are described in Section 2.1.5.

2.2 Gaps in GNN Explainability And Motivation

Before setting out the specific gaps, it is worth stating why the problem addressed in this thesis is difficult, and why it remains open. The difficulty has three sources. First, graphs are discrete, relational, non-Euclidean objects: predictive signal arises from combinations of structure and attributes rather than from individual features, and any perturbation-based analysis must respect validity constraints that continuous relaxations break. Second, message passing entangles evidence: after several layers, a node’s representation mixes signals from its multi-hop neighbourhood, so the evidence for a prediction is distributed across hops and layers and cannot be localised to a small input region in the way saliency methods assume. Third, the object we seek is different from that of most explainability research. The dominant paradigm asks which input elements mattered for one prediction; we ask what internal structure the model has learned across a dataset, layer by layer, and whether that structure can be expressed as compact symbolic rules. This changes the unit of analysis from input features to hidden activations, and the output from importance scores to auditable artefacts.

The problem has remained open for reasons that follow from these difficulties. Techniques for inspecting internal representations were developed for convolutional networks, where filters have spatial semantics; they do not transfer directly, because neurons in a GNN respond to structural patterns that vary with the input graph, and earlier graph representations of network internals were layered, acyclic, and bipartite, which precludes standard graph-theoretic analysis. Extracting rules from activations additionally requires symbolic machinery that neural explainability pipelines do not provide. Finally, progress is hard to measure: as discussed in Section 2.1.6, no unified benchmark jointly assesses fidelity, compactness, stability, and coverage, so global, activation-level methods have lacked both the tools and the evaluation standards

that instance-level attribution enjoys. The contributions of this thesis are shaped by precisely these obstacles: FSAM supplies a graph representation of hidden activity that is amenable to standard analysis, and ActiMine-GNN supplies the extraction step from activations to rules, evaluated under the protocol of Section 2.1.6.

Given these constraints, explanations for GNNs must respect the graph structure, account for multi-hop effects, and remain compact and easy to read. Recent works generate counterfactuals and class-level prototypes to illustrate model behaviour [72]. These lines improve readability, but most operate at the input or output interface and do not use the information carried by hidden activations. As a result, they struggle to show where, across layers, class specificity emerges, and how internal signals can be turned into compact, verifiable rules.

Recent surveys, together with the reviews in Sections 2.1.3 and 2.1.4, indicate that current methods leave several aspects of GNN XAI unresolved [62, 11, 76, 79]. These works map the outlook and highlight a number of open opportunities. In this thesis, we focus on three main research gaps. The first gap concerns global explainability across layers and depth, as there is very limited research that has been done, and the alignment of learned representations with symbols. Most existing pipelines attach knowledge at the input or output interface, leaving the hidden layers largely unexamined. As discussed in Chapter 3 and Chapter 4, FSAM addresses this by showing where layer-wise activation dependencies support or weaken class decisions. In doing so, it allows us to link errors to a loss of class specificity and to formulate plausible hypotheses about the causes of misclassification.

The second gap concerns the form of explanations. Many methods return importance scores or prototypical examples, but do not offer human-readable rules that can be audited, compared, or discussed. Taking inspiration from this research gap, we developed the method called ActiMine-GNN, as we mentioned in Chapter 5. It mines recurrent activation patterns and distils them into rule sets that can be instantiated for individual cases when needed. Because the explanations are derived from interactions inside the network, they can point to the parts of the model that

support particular behaviours and can be compared with domain knowledge where such knowledge is available. We also identified a third important gap in GNN explainability, as there is a lack of standard evaluation practice. At present, coverage across graph types and temporal settings is uneven, and key metrics such as fidelity, sparsity, stability, and rule complexity are not reported in a consistent way. Closely related is the difficulty of capturing meaning in a genuinely structured form, causal or relational, rather than leaving it as unstructured, ungrounded scores, as further discussed in Chapter 5.

These gaps map directly onto the hypotheses and research questions of Section 1.4. The limited study on global activation-level explanations motivates H1 (RQ1–RQ3, Chapter 3) and, in its depth-wise dimension, H2 (RQ4–RQ5, Chapter 4). The absence of auditable, rule-based explanation forms motivates H3 (RQ6–RQ7, Chapter 5). The lack of standardised evaluation is not a gap this thesis can close alone; we address it internally through the protocol of Section 2.1.6, applied consistently when answering all seven research questions.

Chapter 3

Global Explanations using Functional–Semantic Activation Mapping

3.1 Introduction

A central challenge in explaining GNNs is to move beyond instance–wise saliency and understand how representations are organised across layers and classes. Most of the current and recent methods tend to focus on local subgraphs around a single prediction, providing limited information on how hidden layers cooperate to form a class-specific structure or where this structure begins to break down. To address this gap, this chapter introduces a method to extract and represent the knowledge from trained models to understand how hidden layers work. Inspired by functional connectivity analyses, we define a layerwise semantic activation graph that connects pairs of units based on statistical dependencies in their activities. This addresses the gap, identified in Chapter 2, between test-accuracy-based evaluation and an understanding of how internal representations are organised across layers.

This chapter introduces Functional Semantic Activation Mapping (FSAM) [44, 43]. FSAM represents each hidden layer of a trained GNN as a semantic activation graph

whose nodes correspond to neurons (or embedding dimensions) and whose edges encode strong co-activations. This graph-based view allows us to study how groups of neurons support particular output classes, how these groups relate to one another, and how such structure changes with depth. In this chapter, we consolidate and extend the FSAM formulation for the thesis: we present the mathematical definition of the activation graph, describe a step-by-step procedure for its construction, and apply graph-theoretic analysis to relate internal communities to observable model behaviour.

We can summarise the key contributions of this chapter as follows: **(RQ1)** specifies the formal definition of FSAM, how FSAM nodes and edges should be derived from layer activations, and what thresholding criteria produce a reproducible and computationally tractable construction? The second objective **(RQ2)** is to validate the representation by asking to what extent per-layer communities correspond to output classes and how this correspondence varies with depth. The third objective **(RQ3)** is to interpret class pair confusions as explained by overlap or proximity between FSAM communities for detecting groups of neurons that work together to predict similar classes on models and datasets at different levels of complexity.

3.2 (RQ1): FSAM Graph: Definition and Construction

An FSAM graph is a layerwise functional representation that connects pairs of neurons within a given layer of a trained network. This section provides a formal definition of the FSAM graph and recalls the general idea of its construction. Nodes of the FSAM graph correspond to units in a specific hidden layer, and weighted edges represent statistical dependence between their activation series measured over a dataset. We refer to this as a functional semantic graph because edges encode how unit activities vary together. The key idea is to obtain, for each depth, a sparse, threshold-filtered network that summarises which units tend to co-activate,

and to relate these layerwise structures to output classes through per-node association scores when needed. In this way, FSAM establishes a link between internal activation patterns at each layer and class-level behaviour. This approach not only clarifies the internal dynamics of the network but also evaluates how additional layers influence model performance.

3.2.1 Overall Methodology: Generating the Semantic Graph

The aim of this chapter is to improve the interpretability of GNNs by representing their internal mechanisms as semantic graphs, and to assess what these graphs reveal about model behaviour at different layers. FSAM identifies neuron groups involved in decision-making, termed *activation neurons*, and constructs a semantic graph to visualise their relationships. FSAM tracks neuron activations and visualises activation relationships across layers, providing valuable information on the network’s decision-making process and semantic coherence. FSAM explains how the entire network behaves across its layers by producing, for each layer, a semantic graph whose vertices correspond to neurons and whose edges represent strong co-activation patterns. For example, in the CiteSeer citation network one community groups the Database Systems and Information Retrieval classes, and the connections between communities reflect the classes the model treats as related (Table 3.1). Figure 3.1 shows the proposed system architecture. It highlights the steps used to choose an appropriate layer depth and to improve GNN performance. Section 3.2.2 presents the mathematical formulation for generating the semantic graph. This evaluation assesses how well semantic graphs capture the knowledge embedded in trained GNNs by highlighting the influential neurons within each layer. Let $G^{(\ell)} = (N^{(\ell)}, E^{(\ell)})$ be the FSAM graph of layer ℓ , whose vertices are the neurons of that layer and whose edges are the neuron pairs whose Spearman correlation survives the significance threshold of Step 4, and let $\mathcal{K}^{(\ell)} = \{K_1, \dots, K_m\}$ be the communities obtained by Louvain partitioning of $G^{(\ell)}$. For a community K and a class $c \in \{1, \dots, C\}$, the

class selectivity of K for c is

$$s^{(\ell)}(K, c) = \frac{1}{|K|} \sum_{j \in K} r_{pb}^{(\ell)}(j, c),$$

the mean point–biserial correlation of Step 3 between the activations of the neurons of K and the one–vs–rest indicator of class c . A community is *semantic* if $\max_c s^{(\ell)}(K, c) \geq \tau$, in which case it is aligned with the class $c^*(K) = \arg \max_c s^{(\ell)}(K, c)$. The semantic coherence of layer ℓ is the extent to which its communities are class-aligned in this sense.

Semantics in this thesis is therefore a property of the trained model rather than of the application domain. A community is semantic when its neurons are jointly selective for one of the model’s output classes; no correspondence with human-interpretable meaning is claimed, in keeping with the model-internal definition given in Chapter 1.

The definition does not change across datasets, since it refers only to neurons, their co-activations, and the model’s own classes. What changes is the label vocabulary that communities align to and the difficulty of achieving alignment. In Cora, CiteSeer and PubMed the aligned classes are research topics or disease categories; in Amazon Computers and Amazon Photos they are product categories; in Coauthor CS they are research fields. The number of classes also varies, from three in PubMed to fifteen in Coauthor CS, and alignment is harder to achieve as C grows, because a fixed number of neurons must support more one-vs-rest distinctions. Values of $s^{(\ell)}(K, c)$ are therefore comparable across layers within a dataset, but comparisons across datasets should be read with the class count in view.

3.2.2 Mathematical formulation

Step 1: Compute activation values and matrix: We begin by computing the activation values of all neurons in the GNN in a transductive node classification

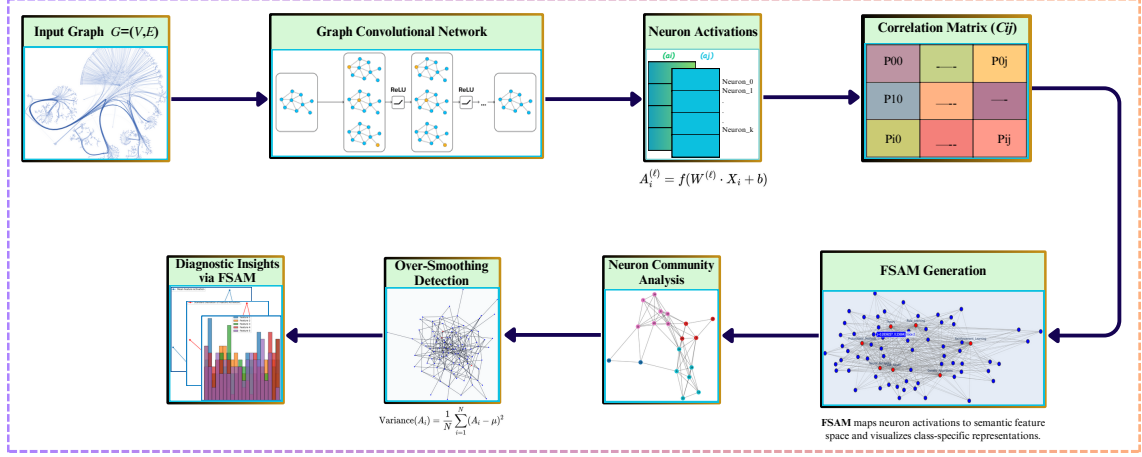


Figure 3.1: The FSAM pipeline. A trained GCN is run on the input graph, per-layer activations are captured, neuron–neuron Spearman correlations and neuron–class point-biserial associations are computed, and global weight thresholding produces one FSAM graph per layer, which is subsequently partitioned into communities.

setting. We assume a single fixed attributed graph.

$$G = (V, E, X), \quad |V| = n,$$

where V is the set of nodes, $E \subseteq V \times V$ is the set of undirected edges (optionally weighted with $e_{v,w} \in \mathbb{R}$), and $X \in \mathbb{R}^{n \times d}$ is the node feature matrix. Node features $x_v \in \mathbb{R}^d$ may be binary (e.g., bag-of-words encodings in citation networks) or real-valued (e.g., normalised numerical attributes in product/coauthor graphs). All node features are known during training, but class labels $y_v \in \{1, \dots, C\}$ are provided only for a subset $V_{\text{train}} \subset V$; the objective is to predict labels for $V \setminus V_{\text{train}}$. **Inputs:** the fixed graph (V, E) , node features X , and partial labels for V_{train} . **Outputs:** per-node probability distributions over C classes, final predicted labels $\hat{\ell}_v$, and intermediate activations for FSAM analysis.

The GNN processes (G, X) to produce, at each layer i , an *activation matrix*

$$A^{(i)} = [a_1^{(i)}, a_2^{(i)}, \dots, a_{h_i}^{(i)}] \in \mathbb{R}^{n \times h_i},$$

where h_i is the number of neurons (feature dimensions) in layer i . Each column

$a_k^{(i)} \in \mathbb{R}^n$ contains the activation values of neuron k across all n nodes. In our extended analysis, we observe that beyond a certain depth, additional neurons contribute diminishing amounts of class-discriminative information due to oversmoothing, which can degrade performance.

Step 2: Graph embedding: To capture the behaviour of *neurons*, we use Graph Convolutional Networks (GCNs) [26], which transform each node’s *ego-graph* (the node and its immediate neighbours) to a latent Euclidean space that encodes structural and feature information. GCN was selected because its aggregation is fixed rather than learned, so activation profiles vary only with the learned weights and the input, whereas learned attention (GAT) or neighbourhood sampling (GraphSAGE) would confound feature transformation with per-instance aggregation. Full-batch transductive training also makes activations deterministic, which a correlation-based method requires, and oversmoothing is best characterised for GCN. This choice carries a cost: we cannot separate general message-passing properties from GCN-specific ones, since GCN’s homophily bias and its expressiveness bound mean that structural distinctions it cannot represent are invisible to FSAM. The embedding for node v in layer ℓ is given by:

$$h_v^{(\ell)} = \text{ReLU} \left(W^{(\ell)} \sum_{w \in N(v) \cup \{v\}} \frac{e_{v,w}}{\sqrt{d_v d_w}} h_w^{(\ell-1)} \right).$$

Notation: let $H^{(\ell)} \in \mathbb{R}^{|V| \times d_\ell}$ be the node embedding matrix whose row $h_v^{(\ell)}$ is the embedding of node v ; we set $A^{(\ell)} := H^{(\ell)}$. Thus, the activation of neuron/feature i at node v is $a_{v,i}^{(\ell)} = h_{v,i}^{(\ell)}$, and the neuron i activation profile across nodes is the *column* $a_{\cdot,i}^{(\ell)}$ of $A^{(\ell)}$. where d_v is the degree of v (including self-loops if present), $W^{(\ell)}$ are trainable weight matrices and ReLU is non-linear activation. At $\ell = 0$, we set $h_v^{(0)} = x_v$. The final layer outputs $\{h_v^{(L)}\}_{v \in V}$ are passed through a softmax classifier to produce per-node class probabilities; intermediate activations $\{A^{(\ell)}\}_{\ell=1}^L$

are retained for FSAM analysis.

Step 3: Compute edge weights: We compute edge weights within the co-activation matrix to analyse the relationships between neurons using Spearman’s rank correlation coefficient, an appropriate metric for capturing monotonic and non-linear associations among activation patterns. The Spearman coefficient ρ_{ij} for neurons i and j is:

$$\rho_{ij} = \frac{\text{cov}(\text{rank}(a_i), \text{rank}(a_j))}{\sigma_{\text{rank}(a_i)} \sigma_{\text{rank}(a_j)}},$$

where cov denotes covariance, $\text{rank}(\cdot)$ returns the rank transformed activation vector across nodes, and $\sigma_{\text{rank}(a_i)}$ is the standard deviation of the ranked values. Although neuron activations can be non-monotonic functions of each other due to signed weights and non-linearities, Spearman’s ρ remains appropriate for our setting, as it measures the strength and direction of any consistent rank order relationship (increasing or decreasing) between nodes, without requiring strict monotonicity.

This measurement quantifies neuron–neuron relationships independently of the classifier head and will be used in subsequent sections to analyse representation structure across layers.

We correlate intermediate layer activations with the predicted class of the final layer to characterise the model’s decision boundary. Let $c \in \{1, \dots, C\}$, $\hat{\ell}_v = \arg \max_c p^{(L)}(y = c \mid v)$ and define the one-vs-rest indicator $\hat{y}_v^{(c)} = \mathbb{I}[\hat{\ell}_v = c]$. Set $G_c = \{v : \hat{y}_v^{(c)} = 1\}$ and $G_{-c} = \{v : \hat{y}_v^{(c)} = 0\}$ with sizes $n_c = |G_c|$, $n_{-c} = |G_{-c}|$, and $n = n_c + n_{-c}$. For layer ℓ and neuron (feature) $j \in \{1, \dots, d_\ell\}$, let $a_{\cdot j}^{(\ell)} \in \mathbb{R}^n$ be the activation column of neuron j across nodes. We compute the point-biserial correlation *per layer, per neuron, per class* as

$$r_{pb}^{(\ell)}(j, c) = \frac{\bar{a}_{j|c}^{(\ell)} - \bar{a}_{j|-c}^{(\ell)}}{s_{j,\text{pooled}}^{(\ell)}} \sqrt{\frac{n_c n_{-c}}{n^2}},$$

where

$$\bar{a}_{j|c}^{(\ell)} = \frac{1}{n_c} \sum_{v \in G_c} a_{v,j}^{(\ell)}, \quad \bar{a}_{j|\neg c}^{(\ell)} = \frac{1}{n_{\neg c}} \sum_{v \in G_{\neg c}} a_{v,j}^{(\ell)},$$

and the pooled standard deviation is

$$s_{j,\text{pooled}}^{(\ell)} = \sqrt{\frac{(n_c - 1) (s_{j|c}^{(\ell)})^2 + (n_{\neg c} - 1) (s_{j|\neg c}^{(\ell)})^2}{n_c + n_{\neg c} - 2}},$$

with $(s_{j|c}^{(\ell)})^2$ and $(s_{j|\neg c}^{(\ell)})^2$ the sample variances of $a_{v,j}^{(\ell)}$ over G_c and $G_{\neg c}$, respectively.

This quantity is computed for every (ℓ, j, c) triple.

Step 4: Thresholding and visualising: To visualise this behaviour, we constructed co-activations graphs for each layer and applied global weight thresholding [66], retaining edges whose correlation is at least $\theta = 0.3$. These visualisations show how adding more layers changes the semantic structure of the learned representations. As the network gets deeper, the later layers add very little new discriminative information. At the same time, the classes begin to overlap more, so they become less separable. FSAM captures and displays this progression. This makes it possible to diagnose oversmoothing and to design GNN architectures that are both more interpretable and more efficient.

The formal definition, together with the step-by-step construction of the FSAM graph, addresses **RQ1**: they provide a systematic way to derive layerwise graph representations from statistical dependencies between neural activities. In the next section, we validate the approach based on FSAM graphs to interpret DNNs by conducting experiments on models and datasets of different levels of complexity.

The dominant cost is the pairwise correlation loop. Ranking each neuron once costs $\mathcal{O}(d_\ell n \log n)$, after which the $\binom{d_\ell}{2}$ correlations cost $\mathcal{O}(d_\ell^2 n)$, so construction is quadratic in layer width and linear in the number of nodes.

Algorithm 1 Construction of the FSAM graph for a trained GNN

Require: Trained GCN f with L layers; graph $G = (V, E)$ with features X ; threshold θ

Ensure: Layerwise FSAM graphs $\{G^{(\ell)}\}_{\ell=1}^L$ and neuron-class associations $\{r_{pb}^{(\ell)}\}$

- 1: $H^{(0)} \leftarrow X$
- 2: **for** $\ell = 1$ **to** L **do**
- 3: $H^{(\ell)} \leftarrow \text{ReLU}(\hat{A}H^{(\ell-1)}W^{(\ell)})$ $\triangleright$ forward pass, activations retained
- 4: $A^{(\ell)} \leftarrow H^{(\ell)}$
- 5: **end for**
- 6: $\hat{\ell}_v \leftarrow \arg \max_c \text{softmax}(H^{(L)})_{v,c}$ for all $v \in V$ $\triangleright$ predicted labels
- 7: **for** $\ell = 1$ **to** L **do**
- 8: $N^{(\ell)} \leftarrow \{1, \dots, d_\ell\}$ $\triangleright$ neurons are vertices
- 9: $R_j \leftarrow \text{rank}(a_j^{(\ell)})$ for all $j \in N^{(\ell)}$ $\triangleright$ rank once per neuron
- 10: **for all** pairs $(i, j), i < j$ **do**
- 11: $\rho_{ij} \leftarrow \text{corr}(R_i, R_j)$ $\triangleright$ Spearman, Step 3
- 12: **end for**
- 13: **for all** $j \in N^{(\ell)}, c \in \{1, \dots, C\}$ **do**
- 14: $r_{pb}^{(\ell)}(j, c) \leftarrow \text{pointBiserial}(a_j^{(\ell)}, \mathbb{I}[\hat{\ell} = c])$
- 15: **end for**
- 16: $E^{(\ell)} \leftarrow \{(i, j) : \rho_{ij} \geq \theta\}$, with weight $\rho_{ij} \triangleright$ global weight thresholding, Step 4
- 17: $G^{(\ell)} \leftarrow (N^{(\ell)}, E^{(\ell)})$
- 18: **end for**
- 19: **return** $\{G^{(\ell)}\}_{\ell=1}^L, \{r_{pb}^{(\ell)}\}$

3.2.3 Semantics in FSAM: Model-Internal and Human Tiers

The word *semantic* can be understood at two levels, and we distinguish them explicitly. *Model-internal semantics* concerns the class structure a trained model has itself learned: a group of neurons is meaningful in this sense when it is jointly selective for one of the model’s output classes, as formalised in the definition above. *Human semantics* concerns conceptual content that a domain expert would recognise, such as a research topic, a functional group, or an ontology class. The two need not coincide, and evidence for one is not evidence for the other. This thesis evaluates the first tier only. All claims of semantic structure in this chapter and in Chapter 4 are claims about class-structured activation, measured by class selectivity and, in Chapter 4, by community-class purity and the separability index. The second tier is not claimed. Establishing it would require a grounding protocol that we do not undertake: extracting the top-activating nodes of each community for annotation by

domain experts, testing alignment against an independent semantic source such as topic keywords or an ontology, and reporting concept completeness alongside purity with respect to that source.

3.3 Evaluation: Experimental Validation of FSAM

To evaluate our approach, we run a set of controlled experiments. We begin by describing the model GCN architecture, the datasets, and the experimental protocol. The evaluation has two parts. The first part addresses RQ2. We validate the FSAM representation by measuring how closely per-layer communities align with output classes. We also examine how this alignment changes with depth. The second part addresses RQ3. We apply community detection to the FSAM graph to identify groups of neurons that act together to predict the class. We then relate class-pair confusions to the overlap or proximity of these communities. Their findings help us to understand which groups support related classes across models and datasets of differing complexity.

3.3.1 Datasets and Experimental setup

We used six well-known benchmark datasets for the experiments:

- **Cora** [52] and **CiteSeer** [52]: citation networks with nodes as publications and edges as citation links. Cora contains 2,708 nodes (7 categories); CiteSeer contains 3,312 nodes (6 categories).
- **PubMed** [36]: biomedical citation network with 19,717 publications in 3 disease categories.
- **Amazon Computers** and **Amazon Photos** [32]: product co-purchase graphs. Amazon Computers has 13,752 products; Amazon Photos has 7,650 products.
- **Coauthor CS** [53]: co-authorship network with 18,333 authors in computer science, labelled by research area.

These six datasets were selected to vary the factors that FSAM is intended to measure, while holding the task fixed. All are transductive node-classification benchmarks, which is the setting in which layer-wise community structure can be compared against output classes. Within that setting they span four domains (academic citation, biomedical citation, product co-purchase, and co-authorship), three orders of magnitude in graph size (2,708 to 19,717 nodes), and class counts from three to fifteen. The class count matters directly for our analysis, since community-class alignment becomes harder to achieve as the number of classes grows; Coauthor CS, with fifteen classes, therefore provides the most demanding test of H1, and PubMed, with three, the least. The e-commerce graphs additionally differ in homophily and degree distribution from the citation networks, allowing us to check that the FSAM construction is not specific to citation-style topology. The datasets used in Chapter 5 differ because the task differs: ActiMine-GNN operates on graph classification, for which node-classification benchmarks are not applicable, and requires datasets with either ground-truth motifs (BA-2Motifs) or domain knowledge against which extracted rules can be checked (AIDS, BBBP). The datasets were selected by four inclusion criteria. A dataset was eligible if it (i) poses transductive node classification on a single fixed attributed graph, as assumed in Step 1; (ii) provides real-valued or bag-of-words node features, required as GCN input; (iii) has at least three classes, so that community-class alignment is non-trivial; and (iv) is an established public benchmark with published baselines. Four categories were excluded. Graph-classification benchmarks were excluded as they address a different task and are used in Chapter 5. Synthetic motif benchmarks such as BA-Shapes and Tree-Cycle were excluded because their ground truth is a planted input subgraph, whereas FSAM produces alignment between hidden-layer communities and classes, so the available ground truth cannot validate its output; BA-Shapes also lacks node features. Heterogeneous and temporal graphs were excluded as outside the scope of the FSAM formulation. Large-scale benchmarks were excluded on computational grounds, since the pairwise correlation is quadratic in layer width and linear in the

number of nodes. These exclusions carry a cost. Without a benchmark containing planted ground-truth structure, FSAM communities cannot be validated against a known answer, so all evidence for class alignment is indirect, resting on agreement with the model’s own predictions rather than on recovery of a structure known in advance. All six datasets are also homophilous, so we cannot establish whether the depth-wise behaviour reported in Chapter 4 persists under heterophily. Finally, excluding large-scale benchmarks leaves the scalability of the construction untested beyond roughly twenty thousand nodes.

We varied the number of GCN layers from 1 to 4. This choice allowed us to evaluate how network depth influences classification Accuracy and the resulting semantic graph structure.

To evaluate the proposed FSAM framework, we conducted experiments in a setting transductive node classification using six benchmark data sets. All experiments were implemented in PyTorch Geometric, with the GCNConv layer as the graph convolution operator. The base GCN architecture consisted of multiple graph convolution layers, each followed by a ReLU activation and dropout for regularisation. A final fully connected layer mapped the hidden representations to the C output classes, followed by a log-softmax output.

Model configurations:-

- **Hidden channels:** 32
- **Dropout rate:** 0.5 after each convolution layer
- **Optimiser:** Adam
- **Learning rate:** 0.01
- **Weight decay:** 5×10^{-4}
- **Loss:** Negative Log-Likelihood (NLL)
- **Epochs:** 200

For Cora, CiteSeer and PubMed we use the public Planetoid splits. Coauthor CS, Amazon Computers and Amazon Photos provide no standard split; for these datasets all labelled nodes are used for both training and evaluation, so the reported accuracies are training accuracies and are not generalisation estimates. The same protocol is applied at every depth, so the depth-wise comparison is unaffected. All runs were repeated with a fixed random seed for reproducibility.

3.3.2 (RQ2): Validating the FSAM Representation (layer wise mapping to output classes)

Finding 1. Class-aligned community structure is present at every layer, is sharpest at layer 2, and degrades progressively thereafter. The layer-wise graphs below show distinct, moderately sparse communities at layers 1 and 2, blurred boundaries and rising edge density at layer 3, and extensive merging at layer 4. This supports **H1**, that co-activation structure can be represented per layer as a graph whose communities align with output classes, and answers **RQ2**: the alignment is not constant with depth but peaks early and deteriorates. After constructing an FSAM graph for each layer of the models and datasets described above, the first stage of our evaluation examined how well these graphs reflect the knowledge encoded by the underlying networks. This stage addresses RQ2. We began by testing whether the statistical dependencies captured by FSAM, edges defined by co-activation of units, are compatible with the way the network links hidden representations to output classes.

Concretely, in the final hidden layer, each unit is connected to every class through the classifier weights. At the same time, the FSAM graph records how strongly each unit’s activity is associated with each class. Our first step is to test alignment between the model and FSAM: we check whether activated units that are strongly connected to a class in the model also appear as class-associated in FSAM, and whether groups of these units form clear communities. We then repeat the analysis layer by layer, so that we can see how the alignment between FSAM communities

and output classes changes with depth.

As illustrated in Figures 3.2, 3.3, 3.4 and 3.5, red dots represent the predicted classes and blue nodes the activated neurons. The layerwise FSAM graph visualisations illustrate the evolution of class structure across GNN depths. In Layer 1 (Figure 3.2), we see clear and well-separated communities. Each group shows a distinct activation pattern that corresponds to a different research topic (output class) in the Cora dataset. The edge density is moderate, which suggests that neurons respond in a specialised way. Moving to Layer 2 (Figure 3.3), the communities become more refined and compact. Some groups of closely related classes begin to merge. This increased organisation coincides with the accuracy gains observed in our quantitative analysis.

By Layer 3 (Figure 3.4), early signs of degradation begin to appear. The boundaries of the communities become less clear and the graph shows a higher edge density. This indicates a loss of neuron specialisation.

The most significant changes occur in Layer 4 (Figure 3.5). Here, we see substantial community merging and many more connections between nodes. These patterns indicate advanced semantic collapse. The graph becomes densely connected, with limited clear structure remaining. This is consistent with oversmoothing, where neuron activations lose class-specific patterns.

Overall, the way the plots change across layers is informative. In the early layers, we see clear separate semantic communities. In the deeper layers, these become blurred and mixed together. This visual progression helps explain why GNN performance tends to degrade when the network is too deep.

Finding 2. Class pairs whose FSAM communities share more neurons are misclassified more often. The association is moderate and positive on both datasets for which it is quantified (Pearson $r = 0.4312$ on Cora, $r = 0.3432$ on CiteSeer), indicating that FSAM communities carry information about the model’s errors rather than grouping units arbitrarily. This provides further support for **RQ2** and the first evidence towards **RQ3**. Figures 3.6, 3.7, 3.8 and 3.9 show a positive

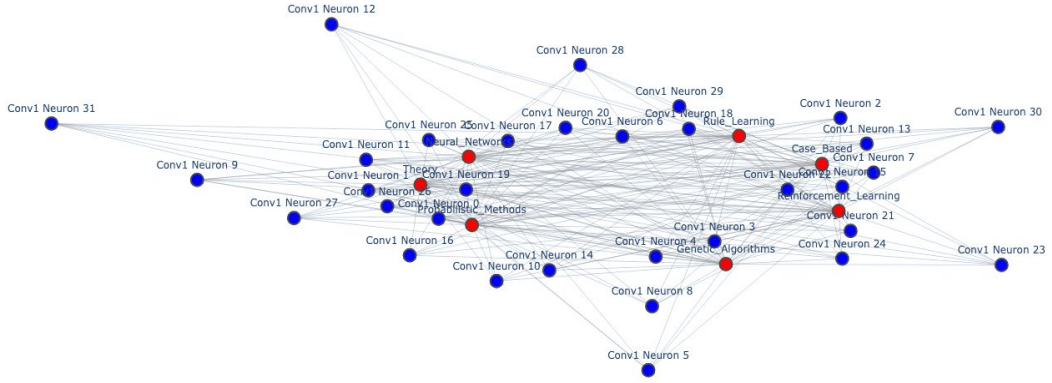


Figure 3.2: FSAM graph for layer k of a 4-layer GCN on Cora. Blue nodes are neurons, red nodes are predicted classes, edges are co-activations retained at $\theta = 0.3$. [Layer 1: communities are distinct and moderately sparse, with each group associated with a different class.]

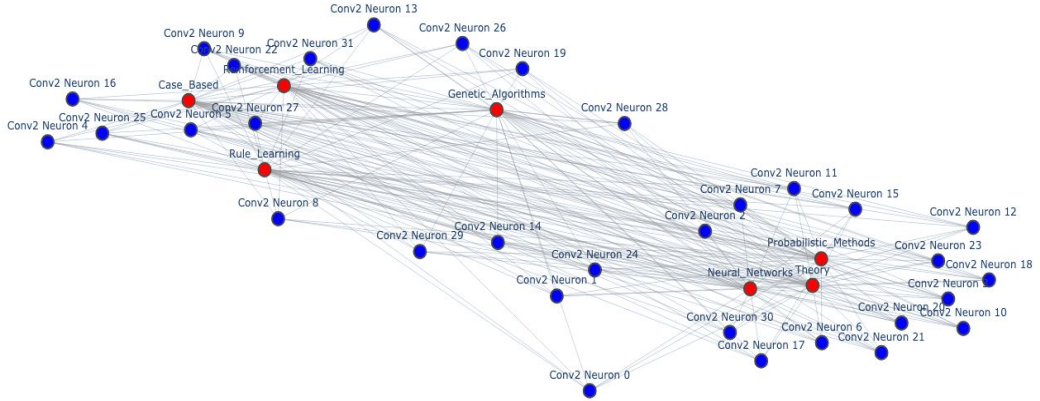


Figure 3.3: Layer 2: communities are more compact and some merge.

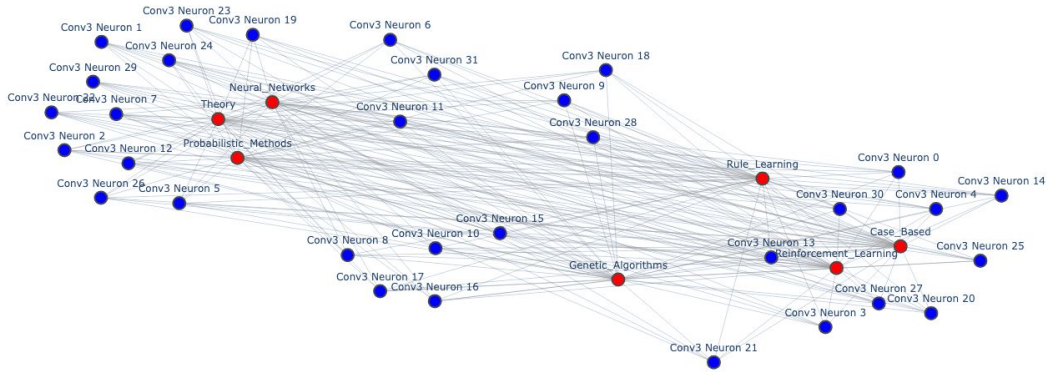


Figure 3.4: Layer 3: community boundaries blur and edge density rises, indicating loss of neuron specialisation.

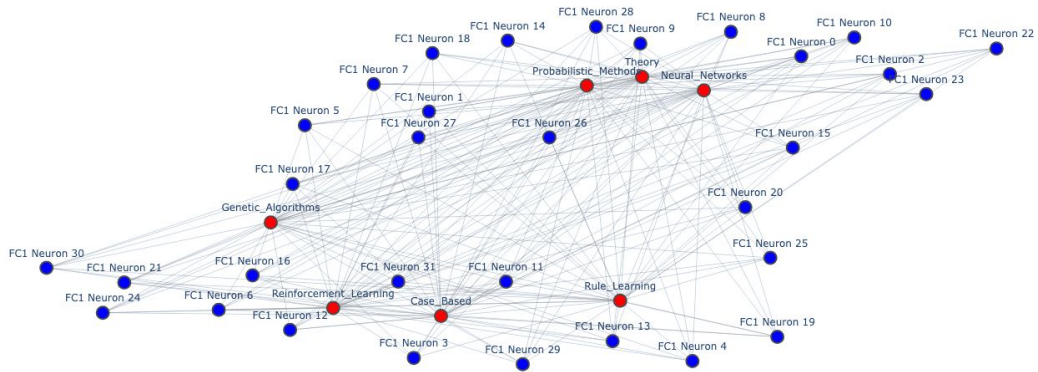


Figure 3.5: Layer 4: communities merge extensively and the graph becomes densely connected, the pattern we term semantic collapse.

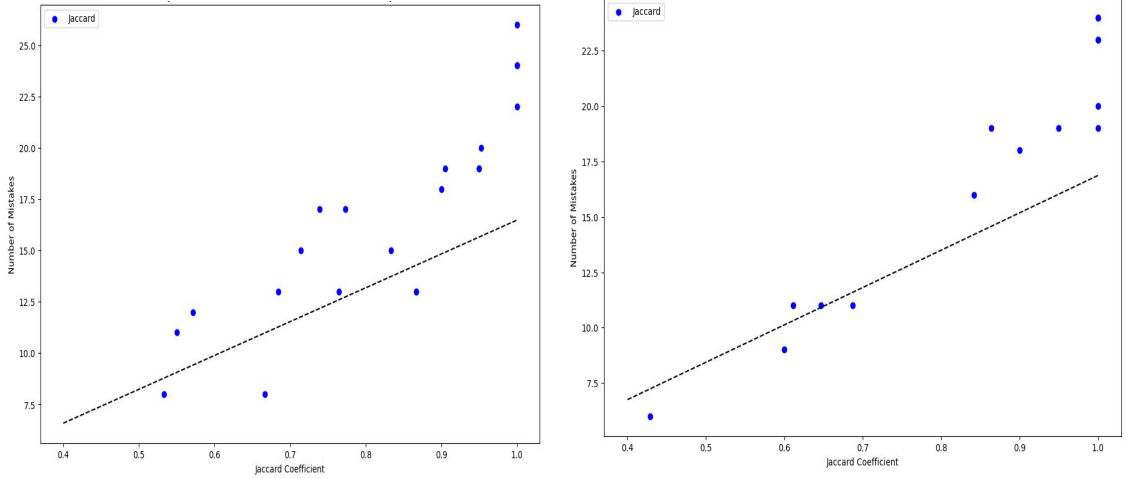


Figure 3.6: Class-pair misclassification count against Jaccard similarity of the corresponding FSAM communities, for Cora (left) and CiteSeer (right). Each point is a pair of classes. Pairs whose communities share more neurons tend to be confused more often (Pearson $r = 0.4312$ and $r = 0.3432$ respectively), a moderate association indicating that shared internal units are one source of class confusion.

correlation between class similarity and the number of mistakes involving those classes. Pairs of classes with greater overlap in their FSAM communities (that is, higher Jaccard similarity) tend to produce more misclassifications.

This pattern suggests that units shared between two classes may be a source of confusion for the model. In other words, we understand that when the same internal units contribute strongly to both classes, the model is more likely to mix them up. This observation forms an initial step towards answering **RQ3**. The correlation values (0.4312 on Cora and 0.3432 on CiteSeer) also provide additional evaluation support for **RQ2**. They indicate that FSAM does not merely group units arbitrarily, We state the evidential basis for these claims explicitly. The quantitative support for class-aligned community structure is the association between community overlap and class confusion, with Pearson correlations of 0.4312 (Cora) and 0.3432 (CiteSeer), together with modularity values of 0.35 and 0.42 for the layer-2 partitions. These are moderate associations: they indicate that FSAM communities are not arbitrary with respect to the model’s classes, but they do not establish that communities correspond

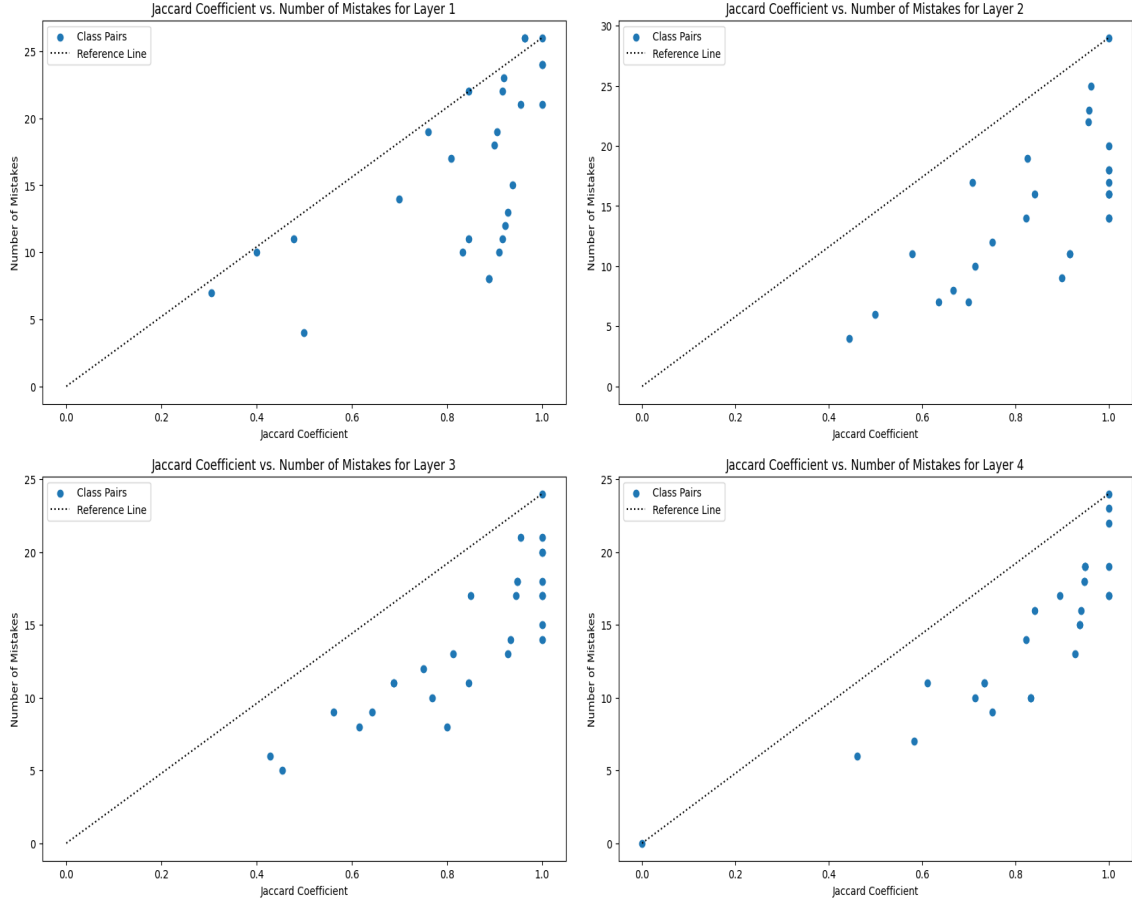


Figure 3.7: Pairwise Jaccard similarity between class-selective neuron sets at layers 1–4 for Amazon Photo Dataset. Brighter cells indicate class pairs sharing more neurons. Similarity increases with depth across most class pairs, consistent with the progressive loss of class-specific structure reported for Cora in Figures 3.2–3.5.

to human-interpretable concepts, and no such claim is made. The topical readings offered for particular communities, such as the grouping of Reinforcement Learning with Genetic Algorithms, are illustrative rather than evidential: they are consistent with the observed groupings but are not independently tested. The layer-wise figures are likewise qualitative illustration, and the quantitative assessment of alignment is carried out in Chapter 4 using community–class purity and the separability index. In the next part of the evaluation, we use graph-theoretic analyses over FSAM. This shows how these structures can be used to interpret the model’s behaviour in a principled way.

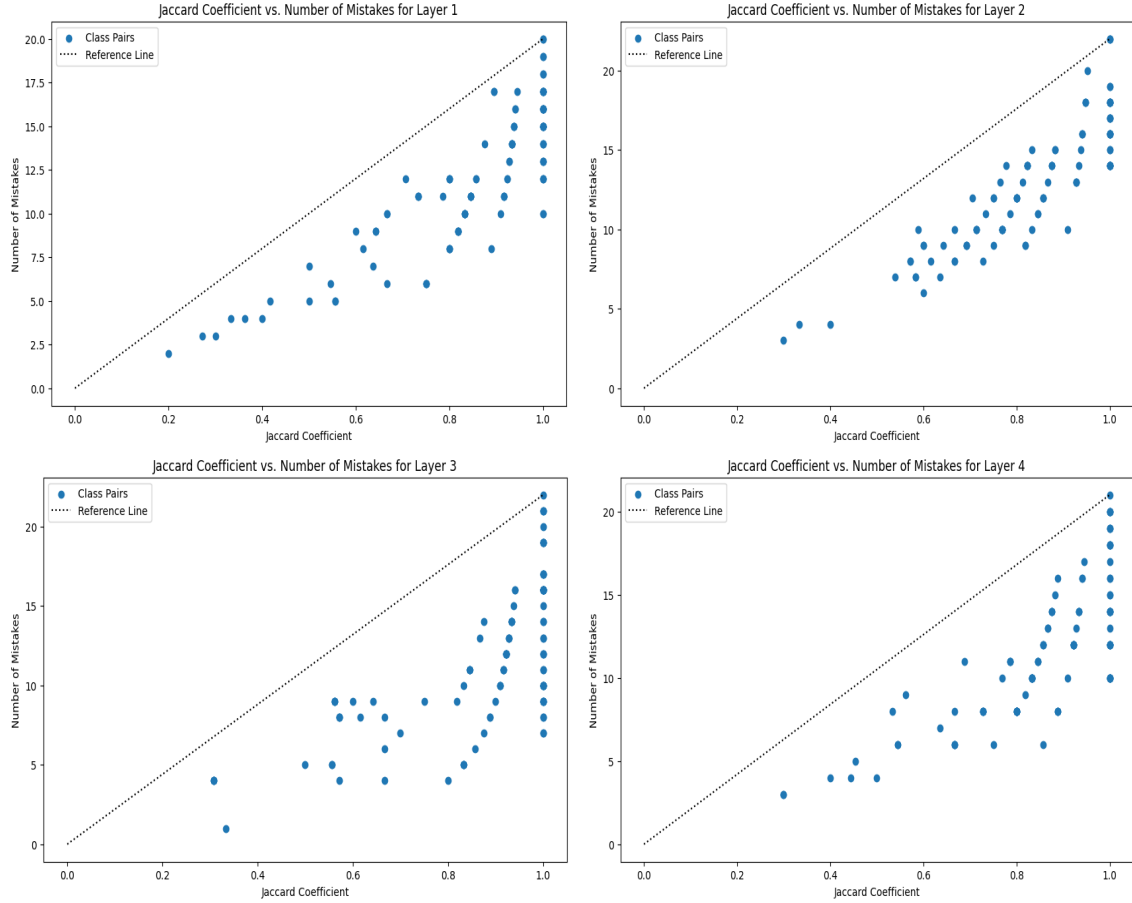


Figure 3.8: Pairwise Jaccard similarity between class-selective neuron sets at layers 1–4 for CoauthorCs Dataset. Brighter cells indicate class pairs sharing more neurons. Similarity increases with depth across most class pairs, consistent with the progressive loss of class-specific structure reported for Cora in Figures 3.2–3.5.

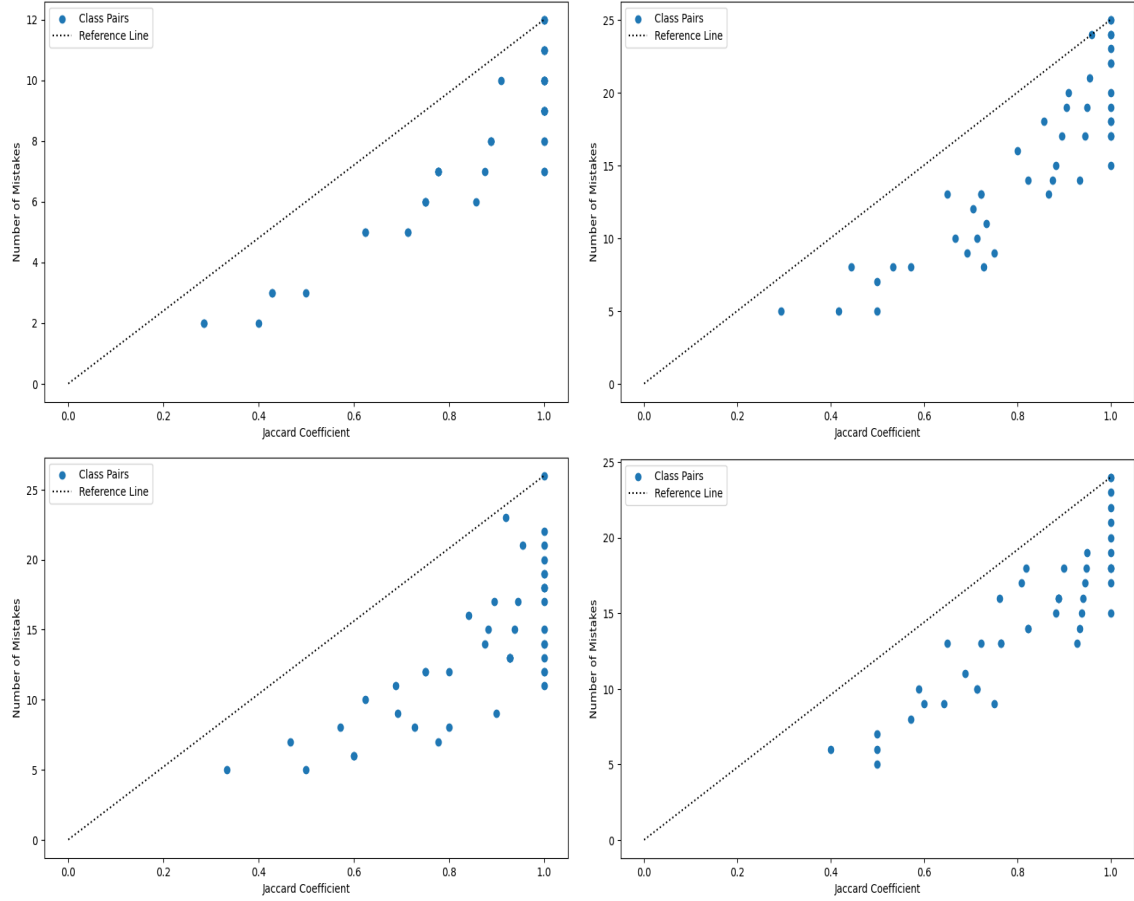


Figure 3.9: Pairwise Jaccard similarity between class-selective neuron sets at layers 1–4 for the computer Dataset. Brighter cells indicate class pairs sharing more neurons. Similarity increases with depth across most class pairs, consistent with the progressive loss of class-specific structure reported for Cora in Figures 3.2–3.5.

3.3.3 (RQ3): Interpreting Class Confusions via Community Detection and Class Pair Similarity

Finding 3. The communities the model forms group classes that it also confuses. Louvain partitions of the layer-2 graphs produce modularity of 0.35 on Cora and 0.42 on CiteSeer, with classes distributed across communities as shown in Table 3.1. Classes sharing a community are those the model represents with overlapping neuron sets, which answers **RQ3**: class-pair confusions can be traced to community co-membership in the FSAM graph. Community structure is a central property of graphs and can reveal how information is organised. When applied to FSAM graphs, communities correspond to groups of units and, by association, classes that tend to co-activate. We analyse the community structure of the layer-wise FSAM graphs across our benchmarks to determine whether it exposes pairs of classes that are similar from the model’s point of view. By relating community overlap and proximity to observed class-pair mistakes, this analysis directly addresses **RQ3**.

We use the Louvain algorithm [6] for community detection. It requires no prespecified number of communities, which is appropriate because the number of class-aligned neuron groups is itself an object of study. Moreover, it operates directly on weighted graphs, preserving correlation strengths rather than binarising them; it is sufficiently efficient to be applied to every layer, model depth, and dataset; and it is widely used in the co-activation graph literature, including by [66], whose thresholding analysis we follow. Louvain maximises modularity, a coefficient in $[-1, 1]$ that measures the deviation of a partition from that expected under a random-graph null model. Higher values indicate denser connections within communities than between communities. As, Table 3.1 lists the detected communities and the classes in both datasets.

For the Cora dataset, adding an extra layer clearly improves the model. As observed in the experiment classes within the same community tend to share similar semantic meanings. For example, Reinforcement Learning and Genetic Algorithms ap-

pear in the same community, reflecting their common roots in Artificial Intelligence methods; Genetic Algorithms are evolutionary search and optimisation techniques that are often grouped with AI methods. We also observed that Neural Network, Theory and Case Based were clustered into a different community. This pattern is consistent with the structure in our initial semantic graph in Figure 3.2. But it also indicates a higher likelihood of error in this class. For the CiteSeer dataset, classes like Database Systems and Information Retrieval were grouped into the same communities, as they both stem from Core Computer Science. Meanwhile, Human-Computer Interaction was placed in a separate community. It is important to note that for both datasets, the modularity exceeded 0.35, indicating that the identified communities are denser regions, though they are not entirely distinguishable. So our initial evaluation shows that our FSAM graph effectively identifies influential neurons, thereby enhancing model interpretability. Community detection has revealed classes perceived as similar by the GNN, where overlapping nodes frequently lead to misclassifications.

Although a more rigorous evaluation is needed, the detected communities appear to group classes that are similar from the model’s perspective. This provides a first step towards answering RQ3: by examining the neurons within these communities and the overlap between class-selective groups, we can begin to trace the origins of class-pair confusions for highly similar classes.

Community	Cora Dataset Classes	CiteSeer Dataset Classes
C1	Reinforcement Learning, Genetic Algorithms	Database Systems, Information Retrieval
C2	Neural Networks, Case-Based, Theory	Agents, Artificial Intelligence, Machine Learning
C3	Probabilistic Methods, Rule Learning	Human-Computer Interaction
Modularity	0.35	0.42

Table 3.1: Distribution of different classes into communities for layer 2

3.4 Qualitative Analysis

FSAM is particularly effective on the Cora and CiteSeer datasets. Both datasets contain a moderate number of classes, and the layer-wise graphs exhibit well-defined community structures in Layers 1 and 2. Furthermore, the relationship between community overlap and class confusion can be quantitatively assessed on a pairwise basis. On these datasets the method does what it is designed to do: it locates the depth at which class structure is strongest, and it identifies which class pairs the model represents with shared neurons. The same construction transfers unchanged to the product and co-authorship graphs, where the layer-wise progression is reproduced across all four benchmarks, indicating that the behaviour is a property of depth rather than of citation-style topology.

Modularity and predictive value carry complementary information. CiteSeer produces the cleaner partition (modularity 0.42 against 0.35 for Cora) while Cora produces the stronger association with confusion (0.4312 against 0.3432). Table 3.1 explains the difference: CiteSeer’s third community contains Human-Computer Interaction alone, and a well-separated isolated class raises modularity while contributing nothing to an analysis of class pairs. Reading the two measures together therefore gives a fuller picture than either alone, and this is how we report them in Chapter 4.

The number of classes is what most affects the granularity of the analysis. With 32 hidden channels throughout, the neurons available per class fall from roughly four and a half on Cora to around two on Coauthor CS, while Louvain returns a number of communities set by graph structure rather than by class count. FSAM still recovers class-aligned structure on the fifteen-class benchmark, which is the most demanding test of H1 in this chapter, with the mapping from community to class becoming many-to-one. Scaling hidden width with class count would improve that mapping further, and is a direct extension of the present design.

3.5 Limitations and Design Trade-offs

We hold the architecture, hidden width, hyperparameters, and threshold fixed across every experiment. This is what allows differences across layers and datasets to be attributed to depth and class structure rather than to configuration, and it is why the depth-wise trends in this chapter can be read directly. The cost is that the results are conditional on that configuration, and the sensitivity of community structure to the threshold θ remains to be characterised. Our quantitative evidence concentrates on Cora and CiteSeer, where the number of class pairs is small enough for the association between community overlap and class confusion to be measured and interpreted pair by pair; the remaining datasets are assessed through their layer-wise graphs. This produces what one can be had, at the cost of a narrower evidential base than six datasets would suggest.

The evidence is observational. Establishing that shared neurons accompany class confusion is what a diagnostic method requires, and it is what FSAM is designed to deliver; establishing that they cause it would require the interventional analysis set out below, which we regard as the natural next step rather than a gap in the present argument.

Finally, we examine depths of one to four layers. This is the range over which these models reach and pass their accuracy optimum, so the transition of interest falls inside it; behaviour at greater depth is untested.

3.6 Summary

As discussed in Section 3.3.2, FSAM produces consistent results across the six node-classification datasets and across GCN models of varying depth. It complements accuracy by offering a global view of the model’s internal organisation, indicating where class structure is formed, merged, or lost as depth increases. In several cases, accuracy rises while class alignment in the FSAM graph deteriorates, suggesting “right for the wrong reasons” behaviour and reinforcing the need for evaluation that

is sensitive to internal structure, not accuracy alone; this observation is quantified in Chapter 4.

In this section, we discuss some limitations of our approach that should be taken into account and indicate possible ways to overcome them. First, FSAM can be computationally expensive for large graphs. As the pairwise dependence grows quadratically with the number of units. Batching and sparsification help, but very large models may still require approximate neighbour streaming estimators. A natural next step is to design subquadratic edge proposals, for example, using approximate k -nearest neighbours in activation space.

Secondly, the choice of the community detection algorithm is another important factor. Different objectives, such as modularity, stability, or information-theoretic criteria, can lead to different partitions. Comparing multiple community detection algorithms and measuring their agreement is left to future work. A natural direction for future work is to apply consensus clustering so that only the most stable structure is revealed. Our study also has several limitations. We focus on standard message-passing GNN backbones and static graphs. Heterogeneous, temporal and very large graphs would require a multi-layer extension of FSAM. Several alternative routes from internal structure to behavioural outcomes could be possible. The most direct is interventional: ablating a class-selective community’s neurons and measuring the change in the confusion matrix, which would establish responsibility rather than co-occurrence. A second replaces the hard partition with continuous representational similarity between class-conditional activation patterns, or with overlapping community detection, allowing a neuron to belong to several class-selective groups. We chose modularity based community detection because it produces an interpretable, dataset-level structure comparable across layers, which is what Hypothesis 1 requires the interventional route is noted as the natural next step for causal claims.

FSAM also provides layerwise evidence that can inform choices about architecture and dataset curation, such as where additional capacity is helpful and where it may instead blur class structure. More broadly, it makes the internal organisation of

a GNN visible and measurable in a systematic way. This does not solve all interpretability issues, but it can support the design of graph learning systems that are easier to analyse, monitor, and refine, while still leaving clear opportunities for scaling, temporal extensions, and richer semantic grounding.

In this chapter we introduced FSAM as a layerwise representation that captures that captures strong co-activations between neurons and encodes them as semantic graphs. Nodes represent units within a layer, and weighted edges reflect dataset-level dependence between their activations. This formalisation, together with the construction protocol, addresses **RQ1**. It offers a reproducible and tractable way to define FSAM nodes and edges, and to control both the threshold and the resulting sparsity.

Our evaluation provides evidence in favour of **H1** and answers **RQ2**. Per-layer communities in FSAM align with output classes, and the strength of this alignment changes systematically with depth. In addition, pairs of classes whose FSAM communities show greater overlap or lie closer together are precisely those that generate more misclassifications. This links the internal structure to observed errors and addresses **RQ3**. In practical terms, it suggests that editing or regularising regions of the activation graph where such overlaps occur may help to reduce confusion between similar classes. Finally, the depth profiles we observe provide initial support for **H2**. In some settings, increasing depth is linked to weaker class-specific structure. In other words, semantic collapse can occur even when overall accuracy remains stable or improves slightly. The next chapter, Chapter 4, develops this hypothesis explicitly. It quantifies how discriminative power changes with depth and identifies the points at which semantic collapse begins. We extend the analysis beyond Cora, CiteSeer, and PubMed to additional benchmark datasets, in order to assess the stability of the FSAM framework across different graph domains.

Chapter 4

Measuring Semantic Collapse: Depth, Oversmoothing, and Discriminative Power

4.1 Introduction and Motivation

In Explainable AI, evaluating and comparing deep models on more than test-set accuracy has led to methods that offer global interpretation tools that characterise how a model behaves overall, not just how often it is correct. GNNs pose a particular challenge because oversmoothing can make node representations increasingly similar as depth grows, which reduces discriminative power [62, 28, 45]. The semantic implications of this effect are less well understood. This chapter builds on Chapter 3, the foundation of FSAM characterises network behaviour layer by layer by constructing, at each depth, a semantic activation graph whose vertices are units and whose edges encode strong co-activations. These graphs are termed semantic in the sense formalised in Section 3.2.2 allowing us to compare how class- or concept-specific structure emerges, merges, or dissipates across depth. We focused on a key question in GNN design to what extent does adding layers improve the model’s ability to represent meaningful patterns, and does higher accuracy necessarily correspond to

more discriminative internal structure? FSAM offers a different, but complementary, perspective. Instead of only confirming that oversmoothing happens on average, it shows *where* and *when* individual neurons lose their class specificity. It also measures how much community separability declines at each layer. In this way, the layerwise consequences of oversmoothing for class-selective structure become measurable. FSAM therefore provides diagnostics that practitioners can use directly during model development. We also observe cases where accuracy improves without a corresponding gain in FSAM quality, indicating correct predictions that rely on less discriminative, homogenised representations. FSAM therefore helps to explain *how* predictions are obtained, clarifies the trade-off between depth and interpretability, and supports the selection of the depth before semantic collapse occurs.

Therefore, FSAM is more likely to be a diagnostic tool for global-level model behaviour, indicating where oversmoothing might occur. Furthermore, our research has practical implications. We have observed cases where GNN performance improves, but FSAM quality does not. This suggests that the model may make correct predictions, but not always for the right reasons. In such cases, it may rely on less meaningful features or on oversmoothed representations. FSAM therefore offers an interpretability-driven way to study oversmoothing. It provides insight into why the model makes its predictions, not just how accurate they are. The main contributions of the chapter are:

1. **Broader validation of FSAM:** As detailed in Section 4.2, we move beyond the Jaccard-based similarity–error analysis of Chapter 3 and carry out a full FSAM-based validation on all benchmark datasets (Cora, CiteSeer, PubMed, Amazon Computers, Amazon Photos, Coauthor CS). In Chapter 4, this includes examining community structure, class–community alignment, and depth-dependent behaviour (1–4 layers), providing a more systematic assessment of FSAM as a diagnostic tool.
2. **Depth-wise alignment of semantics and performance (RQ4)** Depth-wise Behaviour and Discriminative Power. How do accuracy, class specificity,

and FSAM community coherence change as depth increases, and at what depth does discriminative power begin to deteriorate relative to accuracy trends?

3. **Community overlap and confusion (RQ5):** Building on the Jaccard-based similarity–error analysis of Chapter 3, we move from a single fixed architecture to a depth-wise study. We compare the inter-class community overlap in the FSAM semantic graph with the observed class-pair confusion patterns. However, it helps us track how both overlap and misclassification change as model depth increases. We find that deeper architectures often show more overlap between classes and higher confusion rates. Therefore, this can happen even when overall Accuracy stays the same or improves slightly.

This chapter addresses an important research gap in GNN explainability. It examines how increasing depth affects not only predictive performance but also the semantic coherence and discriminative power of the internal representations. We introduce the concept of semantic collapse, the progressive degradation of class-specific semantic patterns in deeper GNN layers, and demonstrate how FSAM can quantitatively measure and visualise this phenomenon.

4.2 Extended Validation of FSAM Across Layer Configurations

This section evaluates whether FSAM captures model behaviour consistently across different depths. We vary the number of layers from one to four on the datasets listed in Section 2.1.5. For each depth, we construct FSAM graphs for every layer and compare three aspects of behaviour: (i) accuracy on the evaluation split; (ii) misclassification patterns at class and class–pair level; and (iii) FSAM-derived structure, quantified by community coherence and class community alignment.

Finding 1. Accuracy peaks at shallow depth while the association between community overlap and error rises monotonically. Across all three

datasets, r reaches its minimum at the depth of best accuracy and increases thereafter, indicating that errors become progressively concentrated in class pairs whose communities have merged. This answers **RQ4** and supports **H2**. The procedure is as follows. After training a model at a given depth, we compute FSAM for each layer. We then detect communities and measure how class-specific they are. To do this, we use community class purity and a separability index that compares the association between and within communities. In parallel, we build class-pair confusion matrices from the model’s predictions. Across datasets, we see a similar pattern. At first, increasing depth improves both accuracy and the structure seen in FSAM; then, after a task-dependent point, this trend reverses. Community coherence becomes weaker, and the alignment between classes and communities deteriorates. Accuracy, however, often flattens out or improves only slightly. The depth at which discriminative signals start to decline then becomes informative. It offers a semantics-aware criterion for choosing the model, rather than relying on accuracy alone. This behaviour is consistent with oversmoothing, an independently established property of embedding geometry; in the FSAM graphs it shows as a layerwise loss of class-selective organisation.

Table 4.1: Accuracy and Pearson Correlation (r) for depths (1–4) across three datasets.

Depth	Amazon Photos		Coauthor CS		Amazon Computers	
	Acc.	r	Acc.	r	Acc.	r
1	0.95±0.038	0.681	0.98±0.010	0.589	0.89±0.038	0.683
2	0.96±0.053	0.650	0.97±0.045	0.756	0.91±0.036	0.630
3	0.94±0.042	0.752	0.96±0.035	0.819	0.88±0.038	0.785
4	0.93±0.048	0.780	0.95±0.032	0.834	0.86±0.042	0.917

Note: r is the Pearson correlation across class pairs between the Jaccard overlap of two classes’ FSAM communities and the number of misclassifications between them. This differs from the Spearman correlation in Section 3.2.2, which is applied to neuron level activation patterns where nonlinear monotonic relationships are expected.

In Table 4.1, we present the results for the Amazon Photo dataset. These results show how layerwise Accuracy changes across configurations. They also illustrate how FSAM captures the relationship between classification errors and community structure.

Table 4.2: Layerwise analysis with statistical validation.

Depth	Amazon Photos		Coauthor CS		Amazon Computers	
	Acc. (95% CI)	r (p)	Acc. (95% CI)	r (p)	Acc. (95% CI)	r (p)
1	0.95 [0.91, 0.99]	0.681 (0.041*)	0.98 [0.97, 0.99]	0.589 (0.038*)	0.89 [0.85, 0.93]	0.683 (0.025*)
4	0.93 [0.88, 0.98]	0.780 (0.008*)	0.95 [0.92, 0.98]	0.834 (0.002*)	0.86 [0.82, 0.90]	0.917 (0.001*)
Key comparisons						
L1 vs L4 r increase	p=0.072 ($\Delta=-0.02$) p=0.037*		p=0.011* ($\Delta=-0.03$) p=0.003*		p=0.029* ($\Delta=-0.03$) p<0.001*	

* Significant at $\alpha = 0.05$. r is the Pearson correlation across class pairs between the Jaccard overlap of two classes' FSAM communities and the number of misclassifications between them. The final row tests the increase in r from L1 to L4.

In Layer 1, the model achieves an accuracy of 95%, with a Pearson correlation of 0.681. At this depth, errors are already partly concentrated in class pairs whose communities overlap, though accuracy is high and the FSAM graph shows clear class structure. At this early network layer, the FSAM graph shows a clear class structure, demonstrating effective feature differentiation. With the addition of a second layer, Accuracy increases to 96%, while the Pearson correlation slightly decreases to 0.650. This layer further enhances class-level separation without introducing notable overlap in neuron activations. FSAM visualisations at this depth indicate that the added layer helps the model make more correct predictions. At the same time, it preserves the distinction between classes, suggesting that the model maintains good semantic coherence. In Layer 3, the accuracy begins to decline, falling to 94%, while the Pearson correlation increases to 0.752. This increased correlation value indicates a greater overlap in neuron activations, signalling a loss of distinctiveness among class-specific features. Here, FSAM shows that oversmoothing is starting to appear. Class representations begin to blur as neuron activations overlap. This is consistent with our earlier findings, where classes with high node overlap in the FSAM graph tended to produce more errors. This pattern points to the need for better strategies to keep class representations separate. At Layer 4, the accuracy drops further to 93%. The Pearson correlation increases to 0.780 as it confirms substantial activation overlap and a loss of clear separation between class representations. FSAM visualisations at this depth show large overlaps between neuron communities. These findings help us to understand that the deeper layers are contributing to oversmoothing. In partic-

ular, nodes that are shared between similar classes are good candidates for targeted adjustment. If we can reduce this overlap, we are likely to improve class separation and reduce confusion between those classes. Overall, the results support FSAM as a useful tool for monitoring behaviour across network depth. In the early layers, Accuracy increases and activation overlap stays low. As we move to deeper layers the overlap between activations is more strongly associated with misclassifications. This positive relationship between class similarity and error rates underlines the diagnostic value of FSAM. It points to specific regions where performance could be improved by minimising activation overlap between similar classes. In turn, this can help to balance depth and semantic clarity in GNNs. Similarly, for the Coauthor CS dataset (Table 4.1), our findings strongly support the hypothesis that FSAM effectively captures layerwise shifts in the network behaviour.

In the first layer, the model reaches a high accuracy of 98%. The Pearson correlation is relatively low at 0.589. Errors are therefore only weakly explained by community overlap at this depth. As we add more layers, accuracy starts to fall slightly. At Layer 2, accuracy is 97%, while the correlation increases to 0.756. This points to a gradual rise in activation overlap. As we observed in the third layer, accuracy drops further to 96%. The Pearson correlation increases to 0.819. This marks the beginning of oversmoothing, as neuron activations overlap more and class boundaries become less clear. In the fourth layer, with an accuracy of 95% and a correlation of 0.834, this trend persists, showing that additional depth now undermines the model’s ability to separate classes effectively.

These findings illustrate that FSAM consistently mirrors the evolving behaviour of the network across layers, accurately capturing the interaction between model accuracy and neuron overlap and confirming its usefulness in diagnosing the point at which further layers no longer benefit performance.

In the Amazon Computers dataset (Table 4.1), we apply the same methodology, analysing how variations in accuracy between layers relate to the structures of the underlying graphs of FSAM. In the first layer, with an accuracy of 89% and a Pear-

son correlation of 0.683, the FSAM graph captures a balanced representation of the behaviour of the network. At this depth, errors are already partly concentrated in class pairs whose communities overlap, though accuracy is high. When a second layer is added, the accuracy increases slightly to 91%. At the same time, the Pearson correlation falls to 0.630. The lower correlation, together with the higher accuracy, suggests that neuron activations remain well separated. This means the model can still distinguish between classes effectively. The FSAM graph at this depth reflects this behaviour and shows a clear class structure. However, in the third layer, the situation changes. The accuracy drops to 88%, and the Pearson correlation rises to 0.785. A higher correlation means there is more overlap in neuron activations across different classes. As a result, the boundaries between classes become weaker. This pattern fits with oversmoothing, where deeper layers cause class representations to blur together. The FSAM graph reflects this effect. It shows that the network is less able to keep class representations distinct as neuron activations converge. In the fourth layer, accuracy falls further to 86% while the correlation rises to 0.917, its highest value across all datasets and depths. Almost all remaining error is now concentrated in class pairs whose communities have merged. At this stage the FSAM graph shows that, although the model still predicts most labels correctly, it no longer preserves the class-specific structure that supported those predictions at shallower depths.

These experiments show that FSAM can describe network behaviour across different configurations. When accuracy improves, the FSAM activation graph usually aligns more closely with the semantic structure. In the early layers, such as the second, the model reaches higher accuracy with low correlation, which indicates good class separation. Beyond this point, extra layers tend to reduce accuracy and increase neuron overlap. This confirms that FSAM can capture the trade-off between model accuracy and class separation. It also shows that FSAM is consistent when representing GNN behaviour across different depths. In addition, these results support the third contribution listed above, identifying situations where the quality of the

FSAM graph declines even as accuracy improves, and underlining FSAM’s value in diagnosing subtle mismatches between the network’s internal structure and the data. Table 4.1 reports the raw layerwise metrics, while Table 4.2 presents the statistical validation. Taken together, they show that: (1) FSAM reliably captures GNN behaviour (all correlations are significant, $p < 0.05$); (2) accuracy declines depend on the dataset (they are significant for CoauthorCS and Amazon Computers); and (3) the association between community overlap and error strengthens with depth (Δr up to $+0.328$, $p < 0.001$), indicating that misclassifications become increasingly concentrated in class pairs whose communities have merged. These findings support our mathematical formulation of FSAM. In that formulation, we defined layerwise activation matrices $A^{(i)}$ and correlation-based measures to quantify semantic structure and class separation. The experimental results confirm the theoretical expectation that, in a transductive node-classification setting, oversmoothing appears as an increase in r , the association between community overlap and error, together with a drop in predictive accuracy.

The observed layerwise patterns, of initial accuracy gains accompanied by low r , followed by accuracy plateaus or declines accompanied by rising r , are consistent with our hypothesis that deeper layers progressively blur class-selective activation patterns.

4.3 (RQ4): Depth-wise Behaviour and Discriminative Power

This section examines how increasing the number of layers affects both predictive performance and the discriminative structure of the internal representation. We vary the depth in a controlled manner from one to four layers. For each configuration, we construct FSAM graphs per layer, derive community partitions, and quantify class specificity using community class alignment and a separability index based on between/within community association. In parallel, we track accuracy and class

pair confusions on the evaluation.

The results show a consistent pattern across datasets. Up to a certain depth, which depends on the task, performance improves. Beyond this point, the trend reverses. Communities start to merge. Overlap between classes increases. Confusions between class pairs also become more frequent, even when overall accuracy stays flat or improves only slightly. In other words, deeper layers gradually blur the class-selective structure visible in FSAM. At the same time, they offer little, if any, gain in accuracy. We make this relationship clear by comparing how accuracy changes with depth to how FSAM separability and community overlap change with depth. The depth that gives the best accuracy does not always match the depth that gives the most coherent FSAM structure. Taken together, this analysis offers a semantics-aware criterion for model selection.

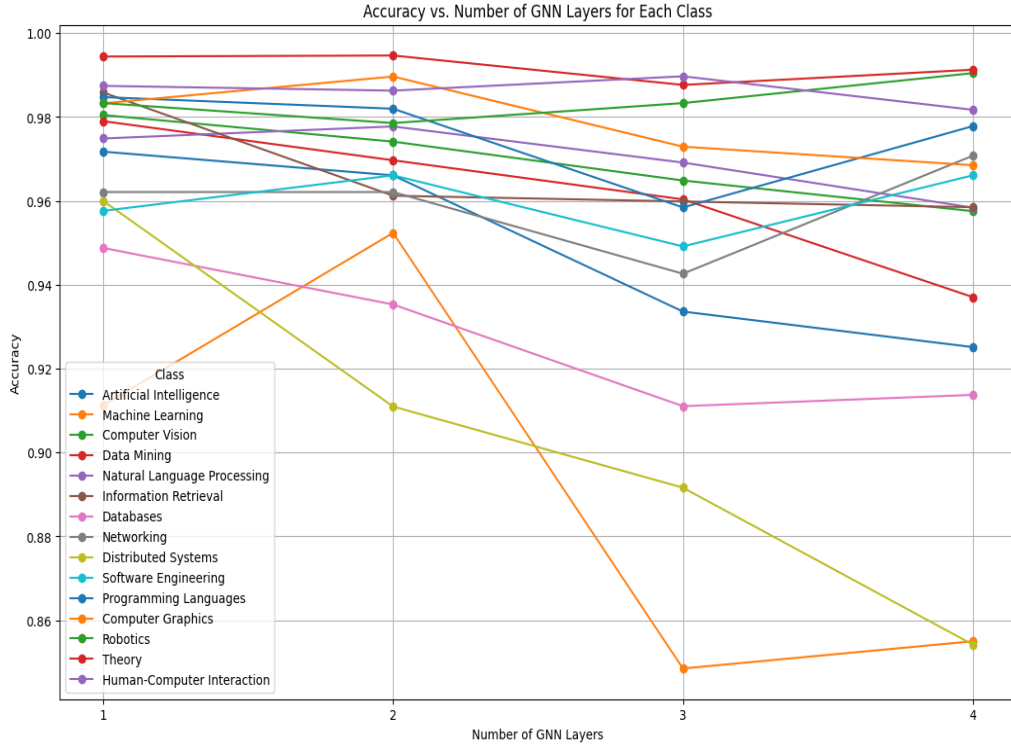


Figure 4.1: Layer wise accuracy per class contribution of CoAuthorCs Dataset

In our analysis, communities are defined within individual GNN layers, where each neuron is a node in the FSAM co-activations graph and edges are weighted by Spearman’s rank correlation of their activation patterns. Community detection (Louvain method) is applied separately to each layer’s graph, producing clusters of

neurons whose activations are strongly correlated for the same input graph. Community detection is performed independently at each layer, and the resulting community indices are not comparable across layers. We therefore track structure through class membership rather than neuron identity: each community is identified by the set of classes its neurons are most strongly associated with, and we follow how those class groupings split and merge with depth. This requires no alignment of neurons across layers, which would in any case be ill-defined since the units of different layers are distinct. The labels C0-C3 in Table 4.3 are within-layer identifiers and should not be read as the same community observed at different depths. Communities play an important role in our analysis. They group together neurons that behave in a similar and semantically meaningful way. By tracking how the size, cohesion, and separation of these communities change with depth, we can see how deeper layers affect the organisation of learned representations. This, in turn, helps us to understand how class-specific accuracy changes and when oversmoothing begins to appear. This evolution is quantified in Table 4.3 throughout this section, *within-community misclassifications* denotes the number of test nodes assigned to a class other than their own, where both classes belong to the same FSAM community at the layer in question. It is computed on the same evaluation split at every layer and depth, and is a subset of the total test error, which is reported as accuracy in Table 4.1. The two quantities can move in opposite directions, since errors may shift between class pairs that share a community and pairs that do not.

The structure of the community is delineated in **Layer 1**, with minimal overlap of neurons between different fields. The community **C0** groups Machine Learning, Data Mining, NLP, and AI, while separate clusters represent **C1** for Theory, Programming Languages, and Software Engineering, **C2** for HCI, Robotics, Computer Vision, Computer Graphics, and Computer Networking, and **C3** for Databases and Information Retrieval. Within-community misclassifications number 1318 at this depth. This suggests that the network maintains clear boundaries between these communities. The structure is consistent with high Accuracy and low overlap in

neuron activations, and this is reflected clearly in the FSAM graph. Upon analysing class-wise Accuracy for this dataset in **Layer 1**, we observed that **C2**-comprising Human-Computer Interaction, Robotics, Computer Vision, and Computer Graphics unexpectedly includes Computer Networking. Although the model placed Computer Networking within this group, the other members of **C2** are perception- and interaction-oriented fields, suggesting that Computer Networking may not belong here. Our class Accuracy representation, Fig. 4.1, supports this observation, yet further evaluation is necessary to confirm the optimal alignment of community structures within the network.

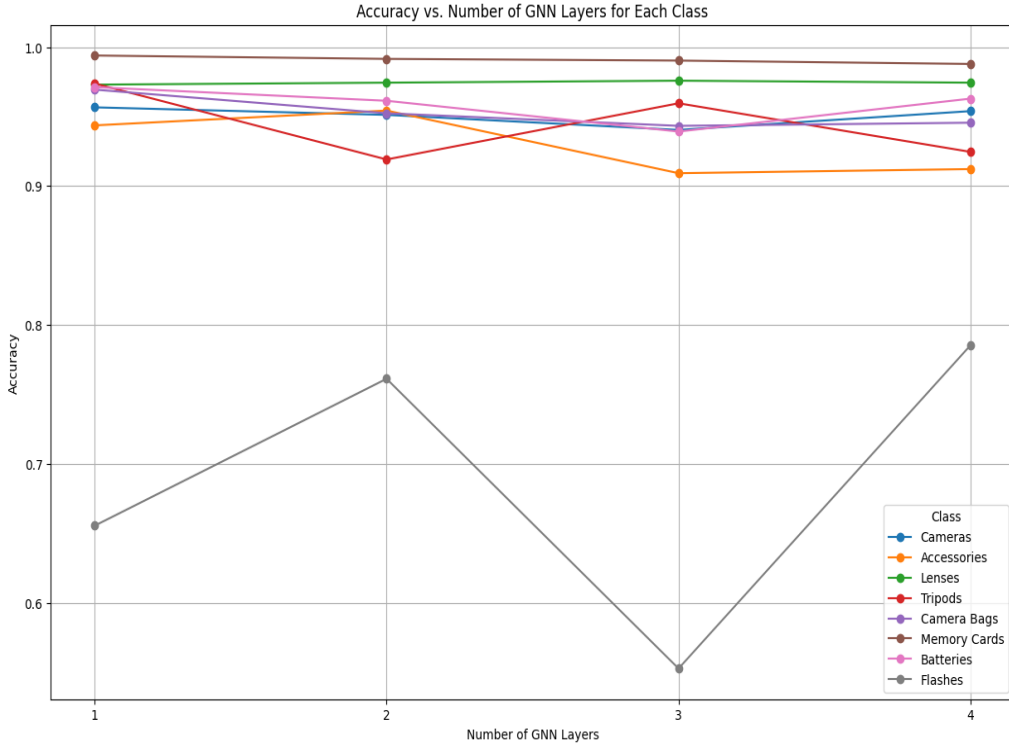


Figure 4.2: Layer wise accuracy per class contribution of AmazonPhoto Dataset

In **Layer 2**, the community structure starts to change. HCI merges into Community **C0** (Machine Learning, Data Mining, NLP, AI, HCI), marking the beginning of greater activation overlap as fields with related meanings cluster more closely together. Within-community misclassifications increase to 1388 as activations begin to overlap between communities. This trend also appears in the FSAM semantic graph. Activations become more correlated, so class representations that were separate at earlier layers start to blend. By **Layer 3**, this effect is clearer. The *Theory* class is

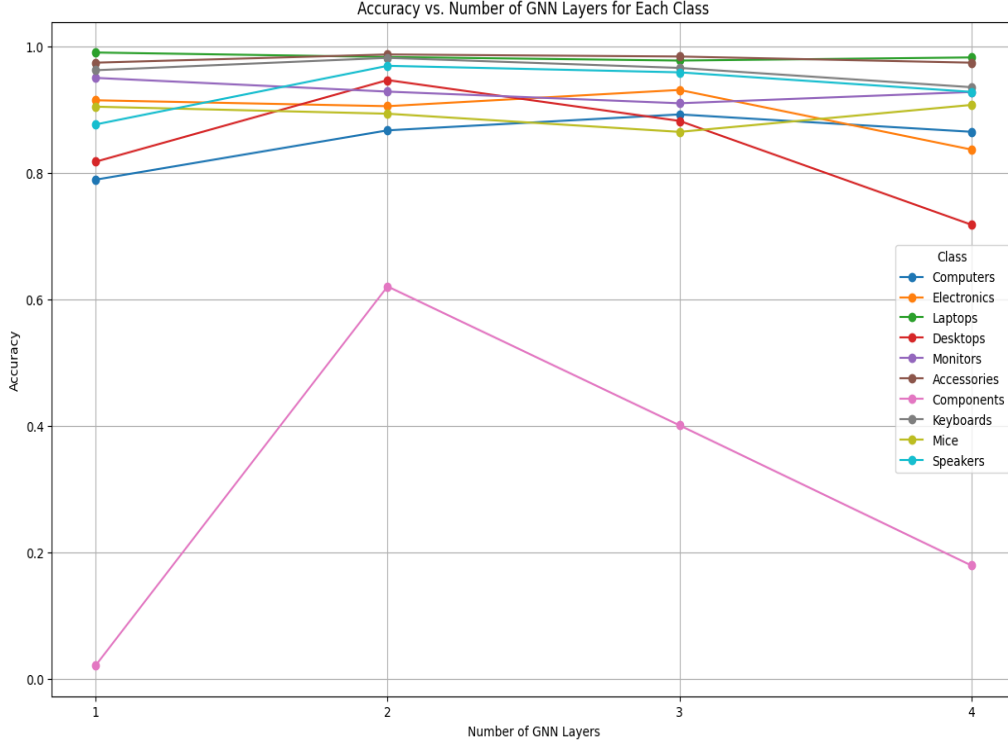


Figure 4.3: Layer wise accuracy per class contribution of Computers Dataset

now part of Community **C0**, and *Programming Languages* and *Software Engineering* form a tighter group in Community **C1**. Within-community misclassifications increase to 1410 as class boundaries become less distinct. The Pearson correlation is also higher at this layer, indicating that a greater share of errors falls between classes sharing communities. In the FSAM graph, this appears as oversmoothing: the communities look less distinct as depth increases. Taken together, these results show that the CoauthorCS community structure changes as layers are added. Class groups that were initially separate begin to merge when neuron activations overlap. This pattern points to a limitation of deeper layers in maintaining class specificity and shows that FSAM can help track how network behaviour shifts across depth. The rise in misclassification and correlation values shows that FSAM works as a practical diagnostic tool. It makes the trade-off between layer depth and community coherence visible and supports the results in Table 4.1.

The error analysis between communities in the Amazon Photos dataset, summarised in Table 4.3, helps to explain how community structure changes with depth and how this affects performance. This breakdown shows that FSAM can track structural

shifts as layers are added and reveal how the model’s view of class similarity evolves. In **Layer 1**, the communities are clearly separated into three groups: **C0** (Cameras, Lenses, Camera Bags), **C1** (Memory Cards, Flashes, Batteries), and **C2** (Accessories, Tripods). Within-community misclassifications number 452, suggesting that the model still preserves good class separation at this early stage, with only limited overlap in neuron activations across communities.

In **Layer 2**, the network reorganises these communities. Community **C0** focuses on Cameras and Lenses, while **C1** expands to include Camera Bags, Memory Cards, Flashes, and Batteries. Within-community misclassifications fall to 410, indicating an improvement in how the model separates these groups. FSAM reflects this adjustment by showing updated relationships between the class representations.

In **Layer 3**, model performance worsens and within-community misclassifications rise sharply to 497. Community boundaries become less clear, with **C0** now containing Memory Cards, Lenses, Flashes, Batteries, and Camera Bags. This broader mix of classes points to increased overlap in neuron activations and aligns with the higher misclassification rate. FSAM captures this change by showing more blurred and less distinct community structure at this depth.

By **Layer 4**, the network exhibits signs of oversmoothing, where distinctions between communities become less clear. Although within-community misclassifications fall from 497 to 406, this apparent improvement may be misleading, as FSAM reveals considerable overlap among communities. In this layer, **C0** isolates to represent Cameras alone, while **C1** groups Flashes, Tripods, and Camera Bags, and **C2** encompasses a diverse mix of Accessories, Memory Cards, Batteries, and Lenses. This indicates that, although errors may decrease, the underlying community distinctions are weakened, suggesting that the model may achieve accuracy without a robust semantic foundation.

This layerwise community analysis, as detailed in Table 4.3, demonstrates that FSAM not only reflects accuracy trends but also captures the nuanced structural shifts within the model as depth increases, reinforcing its utility in diagnosing when

additional layers may lead to diminished class coherence.

In Table 4.3, the Amazon Computers dataset’s analysis effectively captures effective shifts in network behaviour across different layers, especially in cases where accuracy trends diverge from FSAM correlation trends. Changes in community structures and pattern of mistakes in layers demonstrate this.

In **Layer 1**, the FSAM community structure exhibits clear distinctions: **C0** groups components like "Mice" and "Speakers," **C1** includes more complex devices such as "Desktops" and "Laptops," and **C2** contains "Monitors" and "Electronics." Within-community misclassifications number 452 at this layer, indicating that the network maintains distinct activations with reasonable classification performance. This structured community alignment suggests a strong class separation in the network’s internal representation.

At **Layer 2**, the community structure changes. Products such as "Keyboards" and "Mice" move from **C1** to **C0**, as indicated by the labels ^{A»} and ^{C»}. Accuracy improves in this layer, and within-community misclassifications fall to 410. FSAM correlation also falls to its lowest value at this depth, so accuracy and internal structure agree that the representation is well separated.

In **Layer 3**, overall accuracy begins to decline while within-community misclassifications fall to 397, indicating that errors shift towards class pairs that no longer share a community. FSAM’s semantic graph structure shows more overlap within the community **C0**, now grouping “Speakers”, “Laptops”, and “Keyboards” together. The positive increase in correlation in this layer suggests stronger overlap in neuron activations and growing oversmoothing. The network still classifies reasonably well, but its activations no longer align with clear semantic categories.

Similarly, in **Layer 4**, accuracy decreases again and within-community misclassifications rise to 406. FSAM shows that **C0** now mixes “Desktops”, “Speakers”, and “Laptops”, showing even greater overlap between distinct product types. The higher correlation, together with lower Accuracy, indicates that additional depth is now harming class separation. These results suggest that the deeper layers blur class-

specific representations and reduce the semantic coherence of the network’s internal features.

These findings substantiate our hypothesis by demonstrating the ability of FSAM to capture alignment and divergence between accuracy and semantic quality in GNNs. As seen in **Layer 2**, where accuracy improves and r falls, the two signals agree that the representation is well separated. Conversely, in **Layer 4**, where both the accuracy and quality of FSAM degrade, FSAM effectively reflects the reduced class-specific representation, underscoring its utility as a diagnostic tool to evaluate GNN behaviour in layers.

In summary, the depth analysis in this section shows that beyond a task-dependent point additional layers erode class-selective organisation in FSAM while offering diminishing returns on accuracy. This provides a semantics-aware criterion for model selection: among the depths whose accuracy lies within a tolerance of the best observed, prefer the shallowest whose community coherence and class association have not fallen below their maximum.

4.4 (RQ5): Community overlap and class similarity

Finding 2. Class pairs whose communities merge with depth are those whose confusions increase. The association strengthens monotonically with depth on every dataset examined (Δr up to +0.328, $p < 0.001$), answering **RQ5**.

We now test whether the overlap between class-associated communities in FSAM predicts the observed confusions. For each trained model and depth, we detect communities layer by layer. Classes are then linked to communities using the association strength. For every layer, we compute the Jaccard index to measure how much the unit sets for each pair of classes overlap. Class-pair confusion is taken from the evaluation confusion matrix. We then examine how community overlap and confusion counts relate across class pairs, layers, and datasets. Correlation, defined as in Section 4.2, has been utilised to understand the strength and consistency of the relationship. A positive, robust association would support the view that oversmoothing

appears as increased sharing of communities between classes, and that this sharing is reflected in higher mistake rates for those class pairs.

Our analysis shows that FSAM can reveal discrepancies between Accuracy metrics and semantic understanding. As discussed in Section 4.3, we observe two important points (1) cases where Accuracy improves while FSAM graph quality deteriorates, suggesting that the model produces correct predictions without solid semantic grounding; and (2) cases where Accuracy decreases while FSAM quality improves, which may indicate more meaningful learning from misclassifications. These findings highlight FSAM’s value as a model-level diagnostic tool. FSAM helps identify when models are “right for the wrong reasons” and offers insight into how well the internal organisation of a model aligns with its own decision boundary. Because the analysis is anchored to predicted rather than ground-truth labels, it characterises the class structure the model has learned, not the semantics of the training data; establishing the latter would require a separate ground-truth-based analysis.

- **Per-Class Accuracy vs. Number of GCN Layers:** By examining class-specific accuracy across layers, this figure reveals which classes experience increased misclassifications as layer depth grows, underscoring the model’s reduced capacity to maintain distinct representations for these categories as shown in figure 4.1 , 4.2, and 4.3.
- **FSAM Graphs Showing Neuron Activations for Specific Classes:** These graphs show neuron activation patterns within each class, allowing us to track the ability of the GNN to capture class-specific characteristics across layers. They reveal the points at which neuron activations overlap, indicating where class boundaries lose distinctiveness, as shown in figure 3.2, 3.3, 3.4, and 3.5.
- **Community Structures Highlighting Class Groupings:** This visualisation illustrates the community structures of neuron activations, clustering classes based on co-activations. These clusters indicate the relationship be-

tween certain classes and provide insight into the knowledge organisation of GNN, revealing where class separability degrades with additional layers, as detailed in table 4.3.

- **Jaccard Coefficient vs Number of Mistakes at Layer 3** presents the Jaccard similarity between misclassifications for class pairs, demonstrating a positive correlation between high similarity in neuron activation overlaps and error rates. This relationship supports our observation that classes with more significant overlap in FSAM exhibit more frequent misclassifications, as shown in the figure 3.7, 3.8, and 3.9.

Our extended analysis revealed a positive correlation between class similarity and the number of mistakes involving them, as illustrated with examples from the datasets **CoauthorCS** and **Amazon Photos** datasets. Table 4.1 shows that class pairs with higher overlap in the FSAM graph exhibit more misclassifications. Our findings on the **CoauthorCS** dataset show that Layer 1 gives the best performance. When we add more layers, the accuracy drops. This is consistent with what we see in the FSAM graphs. The Jaccard similarity at Layer 2 follows the same pattern. As depth increases, there is more overlap in neuron activations. This makes it harder for the model to tell apart closely related fields such as *Machine Learning* and *Data Mining*. Because these fields appear in the same community, their similarity makes them more likely to be confused and misclassified.

A similar trend appears in the **Amazon Photos dataset**, where Accuracy increases from Layer 1 to Layer 2 but declines with additional layers. This pattern, shown in the Table 4.1, 4.3, is consistent with our Jaccard similarity analysis at Layer 3. In this layer, product categories such as *Memory Cards* and *Accessories* have a high Jaccard similarity. This leads to frequent misclassifications, as their neuron activations overlap within the same community. This suggests that the GNN finds it difficult to tell these classes apart, because they share a large portion of their activation patterns. These results indicate that adding more layers beyond an optimal depth does not always improve how knowledge is represented. Instead, it can cause

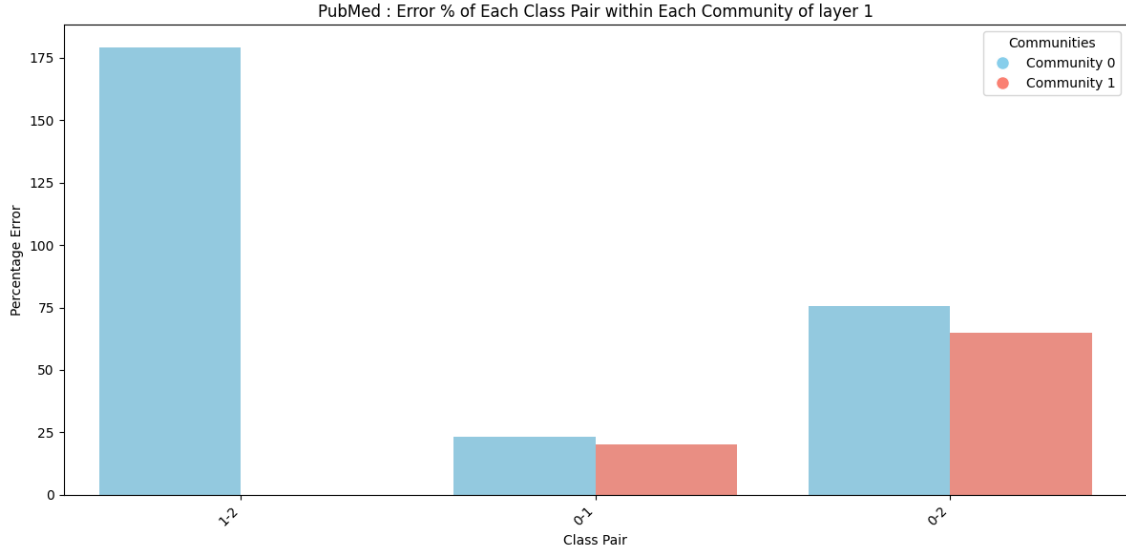


Figure 4.4: Per-class misclassification rates in each community between classes pair in layer 1

an oversmoothing effect. In this regime, neuron activations for different classes become less distinct. As a result, the model finds it harder to differentiate between those classes. Our correlation analysis substantiates this effect, which shows that pairs of classes with significant overlap in the FSAM graph tend to experience higher misclassification rates.

Our analysis of community structures aligns with this observation, allowing us to identify classes that the GNN perceives as similar based on FSAM patterns. We use the Jaccard similarity coefficient to measure how much neuron activations overlap for each pair of classes. This allows us to assess how these overlaps affect the decisions made by the GNN.

In the Amazon Photos dataset, for example, product categories such as *Memory Cards* and *Accessories* exhibit high Jaccard similarity. Consistent with this, the model tends to confuse these categories and makes a larger number of misclassifications between them. A similar pattern is observed across the other benchmarks. For each dataset and each depth, we relate class-pair Jaccard similarity to the corresponding misclassification counts. For illustration, Figures 4.4, 4.5, 4.6, and 4.7 report the layerwise results for PubMed.

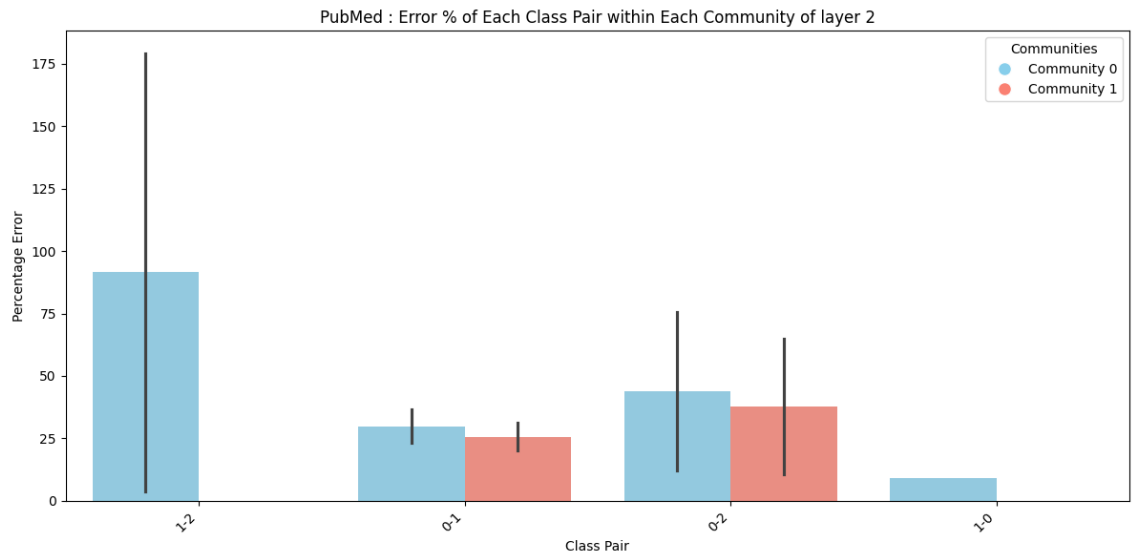


Figure 4.5: Per-class misclassification rates in each community between classes pair in layer 2

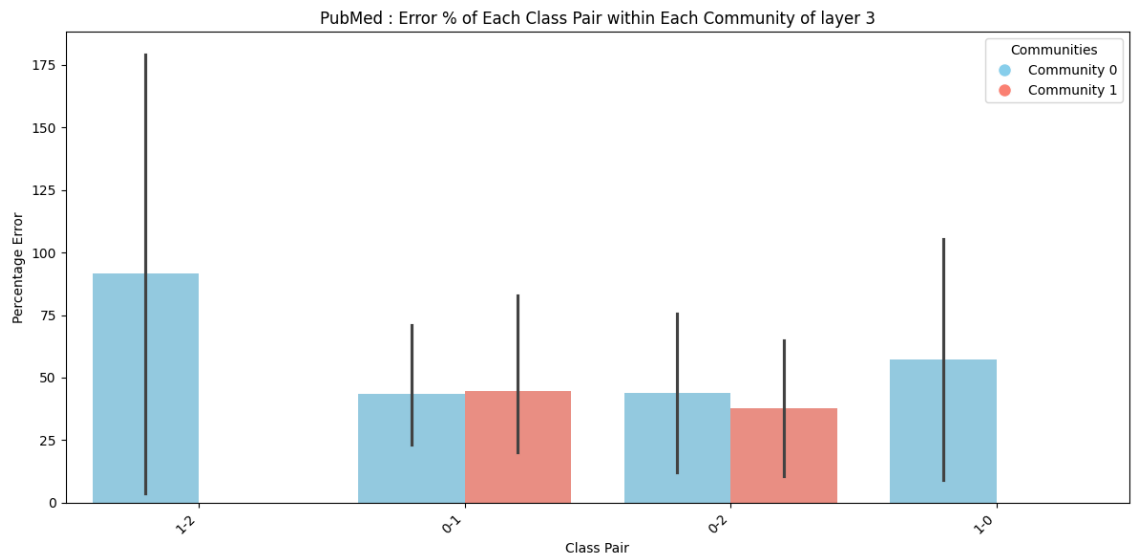


Figure 4.6: Per-class misclassification rates in each community between classes pair in layer 3

Table 4.3: Community structure and within-community misclassifications across layers. A within-community misclassification is a test node assigned to a different class where both classes belong to the same FSAM community at that layer.

A» = accuracy increased, **A«** = accuracy decreased, **C»** = community coherence improved, **C«** = community coherence decreased.

Dataset	Layer	Community (classes)	Within-comm. misclass.
Coauthor CS	1	C0 : Machine Learning, Data Mining, NLP, AI; C1 : Theory, Programming Languages, Software Engineering; C2 : HCI, Robotics, Computer Vision, Computer Graphics, Computer Networking, Distributed Systems; C3 : Databases, Information Retrieval.	1318
	2	C0 : Machine Learning, Data Mining, NLP, AI, HCI; C1 : Theory, Programming Languages, Software Engineering; C2 : Robotics, Computer Vision, Computer Graphics, Computer Networking, Distributed Systems; C3 : Databases, Information Retrieval.	1388 ^{A«,C«}
	3	C0 : NLP, AI, HCI, Machine Learning, Data Mining, Theory; C1 : Programming Languages, Software Engineering; C2 : Robotics, Computer Vision, Computer Graphics, Computer Networking, Distributed Systems; C3 : Databases, Information Retrieval.	1410
	4	C0 : AI; C1 : Computer Networking, Computer Graphics, Information Retrieval, Distributed Systems, Databases; C2 : Machine Learning, Theory, HCI, Data Mining, NLP, Computer Vision, Robotics, Programming Languages; C3 : Software Engineering.	1542
Amazon Photos	1	C0 : Cameras, Lenses, Camera Bags; C1 : Memory Cards, Flashes, Batteries; C2 : Accessories, Tripods.	452
	2	C0 : Cameras, Lenses; C1 : Camera Bags, Memory Cards, Flashes, Batteries; C2 : Accessories, Tripods.	410 ^{A»,C»}
	3	C0 : Memory Cards, Lenses, Flashes, Batteries, Camera Bags; C1 : Accessories, Tripods, Cameras.	497
	4	C0 : Cameras; C1 : Flashes, Tripods, Camera Bags; C2 : Accessories, Memory Cards, Batteries, Lenses.	406
PubMed	1	C0 : Diabetes Mellitus Type 1, Diabetes Mellitus Type 2; C1 : Diabetes Mellitus Experimental.	46
	2	C0 : Diabetes Mellitus Type 1, Diabetes Mellitus Type 2; C1 : Diabetes Mellitus Experimental.	38
	3	C0 : Diabetes Mellitus Experimental; C1 : Diabetes Mellitus Type 1, Diabetes Mellitus Type 2.	32 ^{A»,C»}
	4	C0 : Diabetes Mellitus Experimental; C1 : Diabetes Mellitus Type 1, Diabetes Mellitus Type 2.	62
Cora	1	C0 : Case-Based, Neural Networks, Genetic Algorithms; C1 : Theory; C2 : Reinforcement Learning, Probabilistic Methods.	220
	2	C0 : Case-Based, Genetic Algorithms; C1 : Reinforcement Learning, Rule Learning, Probabilistic Methods; C2 : Neural Networks, Theory.	356
	3	C0 : Neural Networks, Theory; C1 : Case-Based, Rule Learning, Genetic Algorithms; C2 : Reinforcement Learning, Probabilistic Methods.	364 ^{A»,C»}
	4	C0 : Case-Based, Neural Networks, Probabilistic Methods, Theory; C1 : Genetic Algorithms, Reinforcement Learning, Rule Learning.	378
Amazon Computers	1	C0 : Components, Mice, Speakers; C1 : Desktops, Laptops, Keyboards, Computers, Accessories; C2 : Monitors, Electronics.	452
	2	C0 : Keyboards, Components, Mice, Speakers; C1 : Desktops, Laptops, Computers, Electronics; C2 : Monitors, Accessories.	410 ^{A»,C»}
	3	C0 : Speakers, Laptops, Keyboards, Components, Mice, Accessories; C1 : Desktops, Monitors, Electronics, Computers.	397
	4	C0 : Desktops, Speakers, Laptops, Keyboards, Computers, Components, Accessories; C1 : Monitors, Electronics, Mice.	406

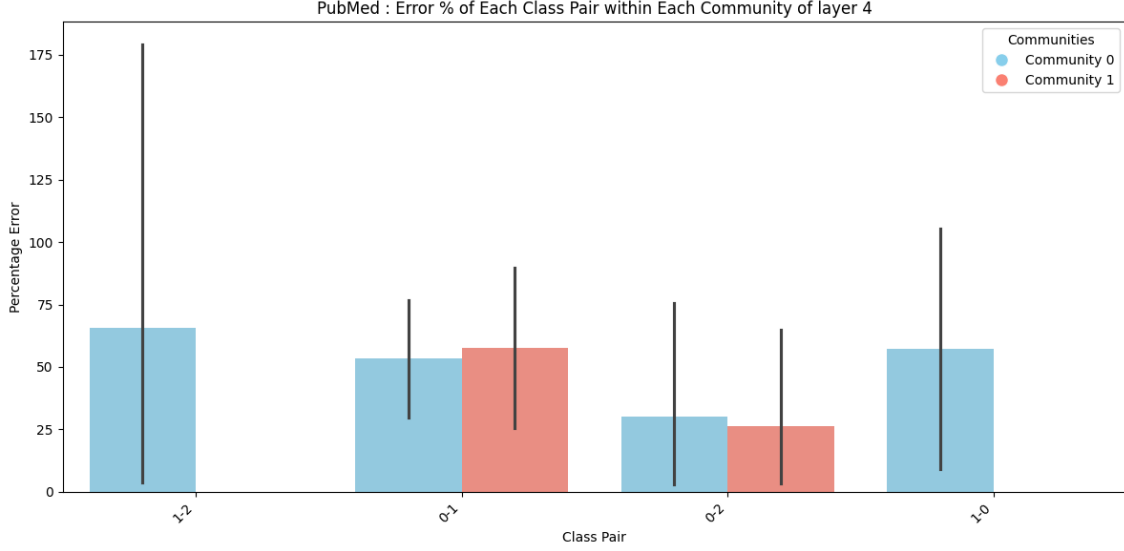


Figure 4.7: Per-class misclassification rates in each community between classes pair in layer 4

4.4.1 Scalability and Computational Complexity

The complexity of monitoring activations and computing semantic correlations does not grow in a simple, linear way. It increases sharply as the number of nodes and edges rises. More complex GNN architectures make this harder. They introduce additional activation patterns, increase memory usage, and create bottlenecks that slow down computation. There are several ways to reduce these costs. Node and edge sampling can limit the size of the problem. Parallelisation techniques, such as GPU acceleration or distributed computing, can speed up the calculations. Approximate correlation methods can also help by trading a small amount of precision for much better efficiency.

4.4.2 Reading the Figures as an Ensemble

The figures in this chapter present the same phenomenon at three levels of resolution, and are intended to be read together. At the coarsest level, Tables 4.1 and 4.2 record what happens to accuracy and to r as depth increases: accuracy peaks early and declines, while r rises monotonically. This establishes that the two signals diverge, but says nothing about which classes are involved.

The per-class accuracy figures (Figures 4.1, 4.2, and 4.3) resolve the aggregate accuracy into its components. They show that the decline with depth is not uniform: some classes hold their accuracy while others lose it sharply, which is what one expects if depth erodes particular class distinctions rather than degrading the representation evenly.

The layerwise misclassification figures for PubMed (Figures 4.4–4.7) resolve the same effect to the level of class pairs, showing where within-community errors concentrate at each depth. Read alongside Table 4.3, which records how class groupings merge, they show that the pairs accumulating errors are those whose classes come to share a community.

Taken as an ensemble, the three levels tell one story: rising r in the tables is not a diffuse loss of quality but the accumulation of specific class pairs merging in the FSAM graph and confusing each other in the confusion matrix. The tables show that it happens, the per-class figures show that it is selective, and the pairwise figures show which pairs are responsible.

4.5 Limitations

The design of this chapter trades breadth for control. Depth is varied from one to four layers while architecture, width, and hyperparameters are held fixed, which is what allows the observed changes to be attributed to depth; the results are correspondingly conditional on that configuration. The evidence is observational: we establish that community overlap accompanies class confusion and that both increase with depth, not that the merging of communities causes the errors. Finally, the depth range examined spans the accuracy optimum on all datasets, so the transition of interest falls inside it, but behaviour at greater depth is untested.

4.6 Summary

We proposed a method for global interpretability and deepened the analysis of GNN behaviour by using FSAM to relate model depth, performance, and semantic structure. FSAM represents each layer as a semantic activation graph whose nodes correspond to neurons and whose weighted edges encode strong co-activations. This construction provides a layer-by-layer view of how class-specific structure is established, merged, or lost. It serves as the basis for our evaluations in §4.2, §4.3, and §4.4. Across multiple datasets (see §2.1.5), FSAM behaves consistently. It captures meaningful relationships between classes. It also highlights communities of neurons that support specific predictions. Together, these findings suggest that the FSAM graph structure reflects the knowledge encoded inside the GNN. In particular, the community patterns recovered from FSAM correspond to class similarity and, where communities overlap, misclassifications rise. These results show that FSAM can distinguish closely related classes and expose where separability deteriorates.

We also examined depth-wise behaviour (Section 4.3). Increasing the number of layers does not always improve semantic organisation. After a task-dependent point, representations become more homogeneous. Class-specific structure becomes weaker, even when overall Accuracy stays the same or improves. FSAM makes this degradation visible. It shows the layers at which community separability starts to decline. It also links the overlap between classes to the confusion observed in the predictions. Taken together, these results support FSAM as a global, activation centred diagnostic. It does not replace Accuracy, but rather complements it. FSAM reveals how internal representations are organised. It indicates that adding more layers stops bringing semantic benefits. It also highlights pairs of classes that are most at risk of being confused.

However, FSAM remains a diagnostic tool. It shows *where* and *when* semantics form or collapse, but it does not, on its own, provide explicit, human-readable rules. Nor does it give per-instance explanations that can be inspected and used directly. This limitation motivates Chapter 5, where we introduce *ActiMine-GNN*. In that

chapter, we take the recurrent activation patterns found by FSAM and turn them into small sets of rules. These rules are instantiated locally for each example and are linked to exemplar subgraphs. In this way, they give practical, rule-based explanations that connect the global structure of the model with the reasoning behind individual predictions.

Chapter 5

Local Rule-Based Explanations for GNNs: A Semantic Activation-Mining Approach (ActiMine–GNN)

5.1 Introduction

A local explanation describes why the model made a particular prediction for one data point. However, explainability still presents open challenges that hinder trustworthy deployment [59]. These include a lack of human-understandable reasoning [68] and the absence of standardised evaluation protocols [47]. In the graph domain, Most explainability techniques for GNNs follow an instance-centred pattern. Methods such as GNNExplainer and PGExplainer [69, 31] highlight influential nodes, edges, or features in a local neighbourhood, typically by learning a mask around the prediction of interest. While these approaches can reveal which parts of the input are important, they do not make the internal reasoning process transparent: explanations are tied to individual instances, difficult to audit across many cases, and only loosely connected to the organisation of hidden representations. More recent

approaches, such as DT+GNN [35], translate layers into decision trees and provide a clearer symbolic description of the decision process, but at the cost of reduced flexibility and scalability as models and graphs grow.

Chapters 3 and 4 introduced Functional Semantic Activation Mapping (FSAM) [44, 43], which provides global, layer-wise diagnostics by tracking co-activation patterns among neurons. In this chapter, our goal is to provide easy-to-understand, visually interpreted building upon the idea of the activation-centred diagnosis of Chapter 3 (FSAM [44]), we now move from where and when class-specific structure emerges to how to turn that internal signal into compact rules that instantiate on individual cases.

We demonstrate the utility of the method in an experiment on BA-2Motifs [69], AIDS [34], and BBBP [61], as we focus on the classification of the graph and evaluate the approach on these three standard benchmarks. These datasets combine synthetic motif detection with ground truth substructures, antiviral activity on molecular graphs and blood-brain barrier permeability, which have chemically meaningful structure which helps us evaluate different domain datasets and qualitative inspection of factual and counterfactual instantiations.

The main contributions of this chapter are as follows. First, we extend the FSAM activation framework to the graph-classification setting. We do this by mining compact local activation rules from the hidden layers. The goal is to achieve high fidelity under a fixed sparsity budget (addressing **RQ6**). Second, we show how these rules can be instantiated on individual graphs as minimal supporting subgraphs, producing factual, rule-based explanations for predictions and, in particular, for misclassifications (**RQ7**). Third, we also demonstrate that these rule subgraph pairs complement FSAM’s global diagnosis for local explanations, which are suitable for counterfactual analysis and for linking hidden activations to human interpretable graph structure.

5.2 Methodology

We instantiate ActiMine-GNN on GCNs [26] for binary graph classification, as our benchmark datasets are binary tasks, and this facilitates comparative fidelity analysis. This choice is not limited to this 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 using class-conditional fidelity at matched sparsity. The approach is built upon the activation extraction approach of FSAM[44], which provides a global understanding of co-activation of a trained GNN by (i) extracting layerwise 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 offers global model-level diagnostics but does not produce the explicit decision rules. We therefore keep the activation extraction for each hidden layer, and binary activation matrices are derived from activations. In place of FSAM’s co-activation graph, we introduce a rule-mining pipeline. For each layer, r -hop ego neighbourhoods define transactions; frequent co-activation patterns are mined under support (top- M) constraints and used as activation-pattern features and a depth limited decision tree is fitted to these features to produce the interpretable rule then the selected rules are instantiated per instance to produce masks for evaluation. Figure 5.1 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.

5.2.1 From Node-Level FSAM to Graph-Level ActiMine

The two settings differ in what constitutes an instance, and the shift is required by the task rather than adopted for convenience. In Chapters 3 and 4, the data is a single fixed attributed graph $G = (V, E, X)$ with $|V| = n$ nodes, each carrying a label, and the activation matrix $A^{(\ell)} \in \mathbb{R}^{n \times d_\ell}$ has one row per node. Statistical

dependence between two neurons is therefore estimated across the n nodes of that one graph, which is what makes the correlation ρ_{ij} well defined. In this chapter the data is a collection $\mathcal{G} = \{G_i\}_{i=1}^N$ of graphs, each carrying a single label y_i , so a per-graph activation matrix $H_i^{(\ell)} \in \mathbb{R}^{n_i \times K}$ has as many rows as that graph has nodes, and the number of rows varies between instances.

This difference explains why a different construction is needed. In Chapters 3 and 4 is the class structure of a node population, and a dataset-level co-activation graph is the natural object, since all nodes are comparable and correlations can be estimated over them. The explanandum here is an individual graph’s predicted label, for which a dataset-level correlation structure gives no per-instance evidence, and neuron–neuron correlation across graphs of different sizes would not be comparable. ActiMine-GNN therefore retains FSAM’s first step, the extraction and binarisation of layerwise activations, and replaces the second. Where FSAM computes correlations over rows to build a co-activation graph, ActiMine treats each row $B_i^{(\ell)}(v, \cdot)$ as a transaction and mines frequent co-activation patterns across the pooled rows of the training graphs, so that a pattern is defined at node level but its presence $\Phi_I(G_i) = \frac{1}{n_i} \sum_{v \in V_i} \mathbb{1}\{I \subseteq T_{i,v}^{(\ell)}\}$ is aggregated to graph level. This aggregation is what carries a node-level signal into a graph-level decision, and it mirrors the mean pooling by which the GCN itself forms its graph representation.

Both settings therefore analyse the same object, the binarised hidden activations of a trained GCN, and differ only in the unit over which statistics are taken: nodes of one graph in Chapters 3 and 4, and nodes pooled across many graphs, then aggregated per graph, here. The datasets differ for the same reason, as node-classification benchmarks admit no graph-level labels and cannot be used for this task.

5.2.2 (RQ6): Mining Compact local Activation Rules (ActiMine)

We begin by converting each continuous GCN embedding into a discrete activation pattern a step motivated by the focus of FSAM on neuron activations [44], but

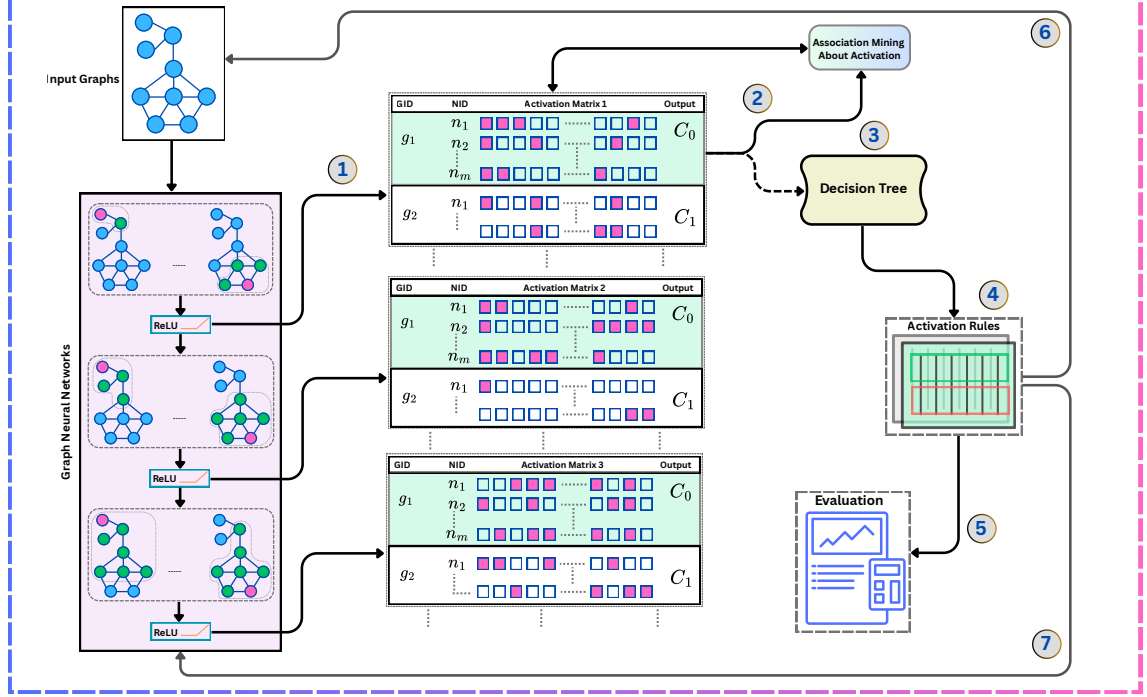


Figure 5.1: **Overview of ActiMine-GNN.** For each hidden layer ℓ : (1) compute node embeddings $H^{(\ell)}$ from the input graph (node colours indicate input categories only and are not explanations); (2) binarise activations to obtain $B^{(\ell)} = \mathbb{K}\{H^{(\ell)} > \tau\}$; (3) mine frequent 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 ($\ell = 1, \dots, L$), producing per layer rule sets and masks. In $B^{(\ell)}$, rows are indexed by (graph ID, node ID) and ones mark active embedding components. Step (7) summarises the resulting insights on the model.

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 $n_{\text{tot}} = \sum_{i=1}^N n_i$.

A trained L -layer GCN produces continuous activations

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

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:

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

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).

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 $I \subseteq \{1, \dots, K\}$ using *FP-growth*. This algorithm compresses the transactions into an FP tree and avoids the large number of candidate sets produced by Apriori. It works well with sparse

binary activations and is fast when K is of moderate size. We also considered Eclat as an alternative, but in our experiments FP-growth was consistently faster in this setting

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

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

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

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

We first filter by $\text{supp}(I) \geq \sigma$ (unsupervised frequency), then rank by conf (class association) and report $\text{lift} > 1$ as evidence over background prevalence. We keep the top- M non-redundant itemsets per layer as *salient* patterns, that is, frequent and class-associated ($\text{lift} > 1$). 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 .

Activation Rule Induction through Decision Trees

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

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

We train a shallow decision tree surrogate on graph features $([\Phi_I(G_i)]_{I \in \mathcal{I}}, y_i)_{i=1}^N$. 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 produced only marginal gains in fidelity against substantially longer rules in validation.

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

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

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

5.2.3 Instantiating Local Explanations from Rules

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

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

where $\text{count}(p_i)$ 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:

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

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. It is worth making explicit what is symbolic in the output of this pipeline, since this is the point at which the neurosymbolic claim of the thesis is realised. A rule $\bigwedge_j (\Phi_{I_j}(G) \bowtie_j t_j) \Rightarrow c$ is a discrete, compositional artefact: its antecedent is a conjunction of predicates over mined activation patterns, and the rule can be read, compared with other rules, and edited independently of the network that produced it. Variable binding occurs at instantiation, where the pattern variables I_j are bound to the specific nodes of a given graph that satisfy them, yielding a concrete explanation for that instance. The counterfactual edits of Section 5.3.2 are operations on this artefact rather than on the model: a node is removed because a rule identifies it, and the effect on the prediction is then observed. In the terms of the taxonomy in Section 2.1.7, this places ActiMine-GNN in the neural-to-symbolic extraction family, where symbolic structure is distilled from a trained network as output rather than injected into training. The neural side supplies the activations and the class signal; the symbolic side supplies the form in which that signal becomes auditable. We do not claim the complementary directions: no knowledge is injected during training, and no logical reasoning is performed over the extracted rules.

How the rule produces a mask. Given a firing rule, we first score nodes by their antecedent support, that is, by how many patterns they satisfy. We then select the highest scoring nodes until the budget α is reached. After that, we apply the chosen mask variant (node-removal, ego, distance-weighted, or top- k edges). This process produces a compact, human-interpretable subgraph that links the global rule to a local rationale. Figure 5.2 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 usually combine a small number of patterns. These patterns appear on substructures that are linked to permeability, such as ring systems and heteroatom contexts. The local mask then highlights the small set of atoms whose Layer 2 activations satisfy the rule antecedent. In this way, the explanation remains concise. For example, at $\alpha \approx 0.17$ we obtain fidelity of 0.88 (Table 5.3).

5.3 Experimental Evaluation

This section describes the empirical setup and presents the results that support our method, the ActiMine-GNN framework.

5.3.1 Datasets, Backbone, and Evaluation Protocol

We assessed GNN interpretability using a three-stage pipeline. First, we performed activation mining. Second, we extracted decision tree rules. Third, we validated these rules using an XGBoost-based procedure. We conducted experiments on three binary graph classification benchmark datasets:

- BA2 [69]: A synthetic dataset containing 1,000 graphs with balanced classes, each featuring either a 5-cycle or a house motif as ground truth

Algorithm 2 ActiMine-GNN: rule mining and instantiation

Require: Trained L -layer GCN f ; graphs $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{test}}$; thresholds $\{\tau_k\}$; support grid Σ ; patterns per layer M ; budget α

Ensure: Per-layer rule sets $\mathcal{R}^{(\ell)}$ and instance masks

```

1: for  $\ell = 1$  to  $L$  do
2:   for all  $G_i \in \mathcal{D}_{\text{train}}$  do
3:      $H_i^{(\ell)} \leftarrow$  forward pass of  $f$  on  $G_i$ 
4:      $B_i^{(\ell)}(v, k) \leftarrow \mathbb{I}\{h_{i,v,k}^{(\ell)} > \tau_k\}$  ▷ binarise
5:      $T_{i,v}^{(\ell)} \leftarrow \{k : B_i^{(\ell)}(v, k) = 1\}$  for all  $v \in V_i$  ▷ transactions
6:   end for
7:   for all  $\sigma \in \Sigma$  do
8:      $\mathcal{I}_\sigma \leftarrow \text{FPGROWTH}(\{T_{i,v}^{(\ell)}\}, \sigma)$ 
9:      $\mathcal{I}_\sigma \leftarrow$  closed itemsets of  $\mathcal{I}_\sigma$  ▷ prune subsumed patterns
10:  end for
11:   $\sigma^* \leftarrow \max\{\sigma \in \Sigma : |\mathcal{I}_\sigma| \geq M\}$  ▷ selected on  $\mathcal{D}_{\text{val}}$ 
12:  rank  $\mathcal{I}_{\sigma^*}$  by  $\text{conf}(I \rightarrow c)$ , ties by lift; retain top- $M$  with lift  $> 1$ 
13:  for all  $G_i \in \mathcal{D}_{\text{train}}, I \in \mathcal{I}$  do
14:     $\Phi_I(G_i) \leftarrow \frac{1}{n_i} \sum_{v \in V_i} \mathbb{I}\{I \subseteq T_{i,v}^{(\ell)}\}$ 
15:  end for
16:   $\mathcal{T}^{(\ell)} \leftarrow$  decision tree of depth  $\leq 4$  fitted on  $([\Phi_I(G_i)]_I, y_i)_{i \in \mathcal{D}_{\text{train}}}$ 
17:   $\mathcal{R}^{(\ell)} \leftarrow$  root-to-leaf paths of  $\mathcal{T}^{(\ell)}$  ▷ activation rules
18: end for

19: function INSTANTIATE( $G, \mathcal{R}^{(\ell)}, \alpha$ )
20:    $R \leftarrow$  rules in  $\mathcal{R}^{(\ell)}$  firing on  $G$ 
21:   score each  $v \in V(G)$  by the number of antecedent patterns it satisfies
22:    $M(G) \leftarrow$  top-scoring nodes until  $|M(G)|/|V(G)| \geq \alpha$ 
23:   return mask  $M(G)$  under the selected mask variant
24: end function

```

- AIDS [34]: 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 [61]: blood–brain barrier permeability (permeable vs non-permeable); 1,640 molecular graphs retained after standard sanitisation and deduplication.

The datasets were selected by four inclusion criteria. A dataset was eligible if it (i) poses graph classification, since the explanandum here is the label of an individual graph; (ii) is binary, so that fidelity is comparable across datasets without class-conditional weighting; (iii) provides either a known ground-truth substructure or domain knowledge against which extracted rules can be checked, which is what

makes a qualitative assessment of the rules possible; and (iv) is an established public benchmark on which the baseline explainers have been evaluated, allowing comparison without re-implementation.

Three categories were excluded. Node-classification benchmarks, including those used in Chapters 3 and 4, provide no graph-level labels and are not applicable to this task. Multiclass graph-classification datasets were excluded because the surrogate here is binary; the pipeline extends to the multiclass setting through one-vs-rest or multiclass surrogates, but we do not evaluate that extension. Large-scale benchmarks were excluded on computational grounds, since pattern mining scales with the total number of nodes across graphs.

These exclusions carry a cost. With three binary benchmarks, two of them molecular, we cannot establish how rule complexity or coverage behaves as the number of classes grows, nor how the method scales to graphs substantially larger than those examined here. BA-2Motifs is the only dataset with a planted ground-truth motif, so the check that the rules recover a known cause rests on a single synthetic benchmark; on AIDS and BBBP the assessment is against domain plausibility rather than a known answer. 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). We use XGBoost’s feature importance only to check that the features in the mined patterns carry a genuine predictive signal. It is *not* used as our sparsity metric. These metrics are particularly important for molecular property prediction, where short and understandable explanations are essential for domain experts. Table 5.1 summarises the principal properties of each dataset. *Neg. Ex./Pos. Ex.* denotes the number of graphs with negative/positive labels in the full dataset 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 5.1: Dataset characteristics and model performance metrics.

Dataset	Neg. Ex.	Pos. Ex.	Avg. Nodes	Avg. Edges	Train Acc.	Test Acc.	Val. Acc.
BA-2Motifs [69]	500	500	25.00	50.92	0.995	0.970	1.000
AIDS [34]	400	1600	15.69	32.39	0.989	0.990	0.975
BBBP [61]	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. The evaluation protocol is nested. Frequent patterns are mined and the decision-tree surrogate fitted on the training split; the minimum support σ and the number of retained patterns M are selected on the validation split; and all reported fidelity, sparsity, and coverage figures are computed on the held-out test split, which is used at no earlier stage.

The depth of the backbone follows from the analysis of Chapters 3 and 4. There we varied depth from one to four layers and found that class-selective structure is strongest at shallow depth and degrades beyond a task-dependent point, with layer 2 the most coherent across datasets. A three-layer backbone therefore sits at the point where the model has sufficient receptive field for graph-level classification while class structure remains intact, and it allows rules to be mined at three depths so that their stability can be assessed. This is why the rules reported throughout this chapter are taken from layer 2: it is the depth at which the preceding chapters found class structure to be sharpest, and Table 5.2 confirms that fidelity peaks there in this setting as well. The depth comparison is therefore consistent across the thesis: layers 1–4 are examined in the node-classification setting, layers 1–3 here, and layer 2 is identified as the operating point in both.

Evaluation metrics. Let $f(G) \in \mathbb{R}^C$ be the frozen softmax scores of GNN for graph G , and $c^* = \arg \max_c f_c(G)$ 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 $G_{\text{keep}}^{(\ell)} = G[M^{(\ell)}(G)]$ and $G_{\text{del}}^{(\ell)} = G[V(G) \setminus M^{(\ell)}(G)]$. 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):

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

Sparsity (mask budget):

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

Unless stated otherwise, we report *means over the covered test graphs* $\mathcal{S}^{(\ell)}$; fidelity is calculated at the (matched) sparsity reported in Table 5.2. Soft variants of fidelity can be obtained by replacing the indicator with $f_{c^*}(G_{\text{keep}}^{(\ell)})$.

In the next section, we evaluate our method in two stages. First, we present a quantitative assessment to test whether the mined activation rules provide faithful, compact surrogates of the GNN. Then, we provide a visual qualitative evaluation, visual analysis in § 5.3.2, where instantiated rules are inspected on exemplar graphs to show clear factual rationales for individual predictions and to guide minimal, validity-preserving counterfactual edits that flip the model’s decision. This evaluation directly addresses **(RQ7)**; it demonstrates that the extracted rules are both accurate enough to trust and readable enough to act upon when probing local decision boundaries.

5.3.2 (RQ7): Visual Analysis of Factual Explanations and Counterfactual Edits

Finding 1. Compact activation rules reproduce the model’s predictions at high fidelity across all three benchmarks. Layer 2 yields 8–13 rules per dataset, achieving fidelity of 0.94, 0.91 and 0.88 on BA-2Motifs, AIDS and BBBP while retaining 15–18% of nodes, and fidelity varies by at most ± 0.08 across layers. This answers **RQ6** and supports **H3**: rules mined from binarised hidden activations are both faithful and compact. Among all datasets, our method, 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 5.2. 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 5.2: Per-layer rule extraction and local explanation metrics for layers L1–L3. “Avg.” denotes the mean over covered test graphs $\mathcal{S}^{(\ell)}$.

Dataset	Layer	# Rules	Avg. Fidelity $\uparrow$	Avg. Sparsity (α) $\downarrow$
BA-2Motifs	L1	14	0.91	0.25
	L2	12	0.94	0.18
	L3	8	0.89	0.22
AIDS	L1	8	0.83	0.28
	L2	10	0.91	0.15
	L3	9	0.88	0.20
BBBP	L1	9	0.85	0.27
	L2	13	0.88	0.17
	L3	15	0.87	0.22

Key observations.

- **Fidelity** ≥ 0.88 . The fidelity scores are 0.94 on BA2-Motifs, 0.91 on AIDS, and 0.88 on BBBP. At the stated budgets, this means that the masked sub-

graph on its own is able to reproduce the original prediction for most test graphs.

- **Sparsity** 0.15–0.18 (mean ≈ 0.17) produces concise explanations, with about 17% nodes retained.

5.3.3 Instance-Level Explanations

Finding 2. Instantiated rules localise the evidence to substructures that can be inspected and edited. On BA-2Motifs the mask induced by a single Layer 2 rule isolates a node whose removal breaks the 5-cycle and flips the prediction; on the molecular datasets the masks highlight small atom sets in ring and heteroatom contexts. This answers **RQ7**: the rules produce factual explanations that are readable and support minimal, validity-preserving edits. ActiMine-GNN discovers twelve high-fidelity activation rules on the BA-2Motifs dataset at Layer 2. The BA-2Motifs dataset contains 1000 graphs: 500 graphs contain a hidden 5-cycle (class c_0) and 500 graphs contain a hidden 5-node “house” motif (class c_1). ActiMine-GNN recovers the explanatory substructure on the large majority of graphs, achieving the highest fidelity among the methods compared.

In Figure 5.2, we show one test graph that contains a 5-cycle. We highlight the support set of Rule 4,

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

in red. When we remove v_1 and its incident edges, the 5-cycle is broken and becomes a simple path. This change causes the GNN’s prediction to flip 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 ties by *lift*, and require a minimum test coverage threshold.

The example in Fig. 5.2 shows the top-ranked layer 2 rule for class c_0 among those that satisfy a predefined coverage threshold. We rank rules using their fidelity on the validation set. When two rules have the same fidelity, we break ties by preferring

higher support and then greater conciseness (that is, shorter path length). This fixed ranking procedure is applied in the same way for all datasets. As a result, we avoid selective reporting and ensure that results are comparable across datasets. The node-removal mask is the simplest intervention, but **ActiMine-GNN** also supports more expressive masking strategies. The *node-removal (keep) mask* keeps only the nodes in V_{mask} and the edges between them, and removes all other nodes and edges. The *ego-graph mask* keeps the full radius- ℓ neighbourhood around each $v \in V_{\text{mask}}$, so that the multi-hop context is preserved. The *distance-weighted mask* assigns continuous weights to nodes or edges that decrease with the shortest path distance from V_{mask} (measured in hops for unweighted graphs, and by path length for weighted graphs). This produces a soft subgraph. Finally, the *top- k edge mask* selects the k edges with the highest weights under the distance weighted scheme, producing a compact 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 .

5.3.4 Comparison with State-of-the-Art Explainers

The baselines were selected by three criteria. A method was eligible if it (i) produces instance-level explanations for graph classification, which is the setting evaluated here; (ii) produce a node or edge mask that can be evaluated under the same sufficiency protocol, allowing fidelity and sparsity to be computed identically for all methods; and (iii) is an established point of comparison in the GNN explainability literature with a public implementation. GNNExplainer and PGExplainer are the standard perturbation-based methods and represent learned soft masks; PGMExplainer represents the surrogate-based family and is the closest baseline in spirit to our own, since it also fits an interpretable model, though per instance rather than globally.

Three families were excluded. Model-level methods such as XGNN [72] produce class prototypes rather than per-instance masks and cannot be evaluated under the

sufficiency protocol used here. Gradient and decomposition methods (for example, Grad-CAM and GNN-LRP) return continuous attribution scores over all nodes rather than a discrete mask, so a comparison would depend on an arbitrary choice of cut-off. Intrinsically interpretable architectures such as DT+GNN [35] require modifying or retraining the model, whereas all methods compared here explain a fixed trained GCN.

The cost of these exclusions is that the comparison is confined to post-hoc, mask-producing explainers. In particular, no rule-extracting method is included, since the available ones operate at the model level rather than per instance, so the comparison establishes that ActiMine-GNN performs well against mask-based explainers, not that it improves on other routes to symbolic explanation. Table 5.3 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. **Sparsity** quantifies conciseness and is the fraction of nodes retained by the mask (lower is better).

As shown in Table 5.3, ActiMine-GNN achieves higher fidelity than the baselines while retaining substantially fewer nodes, which suggests that the rule-driven approach is closely aligned with the model’s decision process and provides focused, human-interpretable rationales. The baselines are reported at their own achieved sparsity rather than at a matched budget, so the comparison establishes that ActiMine-GNN reaches higher fidelity with a more concise mask, not that it would dominate at equal sparsity. A matched-budget comparison is identified as future work.

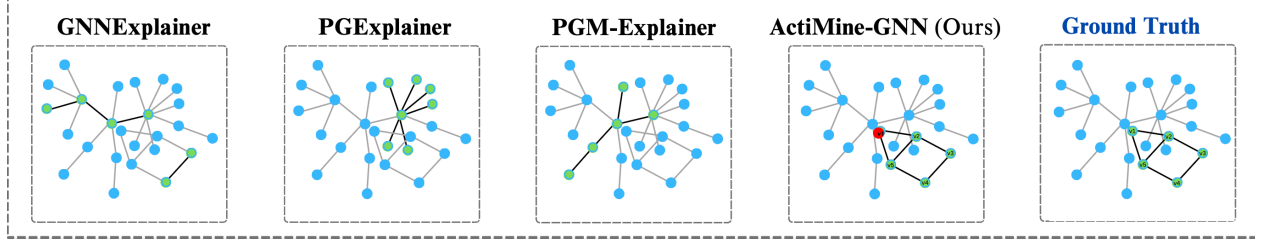


Figure 5.2: Comparison of the explanations on BA-2Motifs with GCN classifier

Table 5.3: Performance of explanation methods across different datasets

Dataset	Method	Fidelity	Sparsity
BA-2Motifs	ActiMine-GNN (L2)	0.94	0.18
	GNNExplainer [69]	0.38	0.61
	PGExplainer [31]	0.35	0.95
	PGM-Explainer [58]	0.90	0.65
AIDS	ActiMine-GNN (L2)	0.91	0.15
	GNNExplainer [69]	0.49	0.50
	PGExplainer [31]	0.46	0.54
	PGM-Explainer [58]	0.47	0.85
BBBP	ActiMine-GNN (L2)	0.88	0.17
	GNNExplainer [69]	0.50	0.50
	PGExplainer [31]	0.51	0.53
	PGM-Explainer [58]	0.48	0.88

5.3.5 Qualitative Analysis

Where the method is clearest. BA-2Motifs is the setting in which the rules can be checked against a known answer, and it is where they perform best (fidelity 0.94 at $\alpha = 0.18$). The Layer 2 rule examined in Section 5.3.2 isolates a single node whose removal converts the 5-cycle into a path, so the rule not only reproduces the prediction but identifies evidence whose removal changes it. This is the strongest form of confirmation available: the mask coincides with the planted motif that defines the class.

Molecular graphs: plausibility in place of ground truth. On AIDS (0.91 at $\alpha = 0.15$) and BBBP (0.88 at $\alpha = 0.17$) no planted motif exists, so the rules are assessed against domain plausibility. The masks concentrate on ring systems and

heteroatom contexts, which are the substructures associated with permeability and with activity, so the rules are chemically interpretable even though no single substructure defines the label. AIDS yields the more concise masks of the two despite being the larger dataset, which is consistent with its smaller average graph size (15.7 nodes against 24.1): fewer nodes are available to enter the mask, and the class signal is correspondingly more concentrated.

Which dataset is most demanding. BBBP is the hardest of the three by every measure we report: it has the lowest rule fidelity (0.88), needs the most rules at Layer 2 (13 against 12 and 10), and its backbone GCN also has the lowest test accuracy (0.787 against 0.970 and 0.990). The three move together, which is informative rather than coincidental: permeability depends on global molecular properties rather than on the presence of a single substructure, so both the model and any subgraph-based explanation of it face the same difficulty. Where the model’s own decision rests on diffuse evidence, a compact mask has less to isolate. That the rules still reach 0.88 under a 17% budget in this setting is the more demanding demonstration of the two.

Interpretation. Taken together, the three datasets suggest that rule fidelity tracks how localised the class evidence is. Where a single substructure determines the label, as in BA-2Motifs, the rules recover it and a minimal edit flips the prediction. Where the evidence is distributed, as in BBBP, more rules are needed and fidelity is lower, though still high in absolute terms. This is a property of the explanandum rather than of the mining procedure, and it predicts that the method will be most informative on tasks with localised structural causes, which is also where explanations are most actionable.

5.4 Limitations and Design Trade-offs

The design of this chapter prioritises interpretability over breadth, which entails several methodological trade-offs. To ensure a consistent basis for comparison across all three datasets, we fix the GNN backbone, binarisation thresholds, tree depth, and

number of retained patterns throughout the experiments. This controlled configuration enables observed differences in fidelity and rule complexity to be attributed primarily to dataset characteristics rather than to variations in experimental settings. Consequently, the depth-wise stability reported in Section 5.3.2 can be interpreted directly under a consistent methodological configuration. The cost is that the results are conditional on that configuration, and the sensitivity of the mined rules to the binarisation thresholds in particular is not characterised. The comparison with existing explainers is made at each method’s own achieved sparsity rather than at a matched budget. This is sufficient to establish that ActiMine-GNN reaches higher fidelity with a more concise mask. Validation against a known answer is available on one dataset. BA-2Motifs has a planted motif, so the rules can be checked against the substructure that defines the class; on AIDS and BBBP, the assessment is against domain plausibility. This is the appropriate standard for datasets whose labels have no single structural cause, but it means the recovery claim rests on the synthetic benchmark. Finally, the evaluation is confined to binary graph classification with a GCN backbone. This is the setting in which fidelity is directly comparable across datasets without class-conditional weighting, and the pipeline extends to multiclass through one-vs-rest or multiclass surrogates; that extension is not evaluated here. A fifth principle, which we identify but do not report here, is stability: the variance of an explanation under repeated training with different random seeds and under small perturbations of the input graph. An explainer that returns a different rule set or mask on each run is of limited use for audit, however high its fidelity, and stability is included in the metric set of Section 2.1.6 for exactly this reason. Reporting the variance of rule sets across seeds and the agreement between masks under minor edge perturbations would complete the blueprint, and we regard its absence from the present evaluation as a limitation rather than an omission from the framework.

5.4.1 Towards a More Systematic Evaluation of Local Explanations

The ActiMine-GNN framework points towards a more systematic and rigorous way to evaluate local explanations in GNNs. It moves beyond isolated metrics and aims for a more complete view of faithfulness and robustness.

This approach rests on several principles. First, it assesses explanation quality at a matched sparsity budget, comparing fidelity for equally concise explanations. This makes the comparison fair and reveals whether an explainer is using its “budget” efficiently to capture the model’s reasoning. We apply this principle across the layers of our own method, where all reported fidelities are obtained at the stated budget α ; extending it to a cross-method comparison, in which each baseline is constrained to the same budget, is the natural next step and is not undertaken here (Section 5.4). Second, the framework checks robustness across model depth. It shows that the fidelity and sparsity of the mined rules stay stable across different hidden layers, with only small fluctuations. This suggests that the extracted rationales are not tied to a single layer. Instead, they reflect core, stable patterns in the GNN’s decision process.

Third, ActiMine-GNN attempts to connect local and global faithfulness. It does this through global surrogate alignment. A surrogate model, such as XGBoost, is trained on features derived from the mined activation rules.

Finally, the evaluation combines quantitative metrics with qualitative, counterfactual checks. Visual inspection, together with the change in prediction caused by removing a minimal masked subgraph (as in the BA-2Motifs example), offers an intuitive sense of causal necessity.

Taken together, matched sparsity, depth-wise robustness, global surrogate alignment, and counterfactual analysis form a more comprehensive evaluation blueprint. This helps to ensure that explanations are judged not only on numerical performance, but also on their true faithfulness, consistency, and interpretability. Findings 1 and 2 together address **H3**: compact rules extracted from binarised hidden

activations capture the factual evidence behind a prediction (Finding 1) and support small, validity-preserving changes to it (Finding 2). Together with the global diagnostics of Chapters 3 and 4, this completes the progression from where class structure forms to what, in a particular instance, the model relied upon.

5.5 Summary

In this chapter, we introduce ActiMine-GNN, a method for explaining why a trained GNN makes mistakes on graph classification tasks. Our approach works in two main steps. First, we binarise the hidden layer activations and use decision tree induction to mine a small set of activation rules. Then, we apply these rules to each graph to generate an instance-level mask, showing which parts of the graph are most relevant to the model’s decision. We quantitatively validated ActiMine-GNN on benchmark datasets, focusing on both the intrinsic quality of the rules discovered and their ability to explain the model’s decisions faithfully. 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. As 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. As our experiments focus on graph level binary classification with a GCN backbone, the approach is model agnostic and extends to multiclass settings (for example, one versus rest or multiclass surrogates).

Our evaluation on three standard graph benchmark datasets, BA-2Motifs, AIDS, and BBBP, shows that the mined rule sets achieve high fidelity at controlled sparsity and remain stable under a reasonable threshold, thereby addressing **RQ6**. In the § 5.3.2, we complemented these results with qualitative analyses of instantiated rules on graphs. This demonstrated that the factual explanations are concise and intelligible, and that they align with domain structure, thereby addressing **RQ7**.

After evaluating the suitability of our approach, the evidence indicates that local ac-

tivation rules can faithfully account for the model’s reasoning, making it suitable for expert audit. While our focus here is on factual explanations, the same instantiation machinery can guide minimal, validity preserving edits as a basis for counterfactual analysis in future work. The chapter thus fills the gap left by global diagnostics in Chapter 4 by providing actionable, instance-level justifications that are grounded in the network’s internal activations. In Chapter 6, we revisit the main findings of this thesis, discuss their implications for explainable graph learning, and outline directions for future work.

Chapter 6

Conclusions

6.1 Introduction

This chapter concludes the thesis by revisiting the three studied hypotheses and the associated research questions studied throughout. It summarises the main outcomes and their impact on Explainable AI for graph neural networks, with implications for neuro-symbolic methods more broadly. We first highlight the key properties of the Functional Semantic Activation Mapping (FSAM) framework and the ActiMine-GNN pipeline, which together form the core technical contributions of this work. We then restate their respective roles in global, activation-centred analysis and local, rule-based explanations, and how they address gaps identified in GNN explainability and neuro-symbolic XAI in earlier chapters. The chapter also reflects on the limitations of the proposed approaches and the open challenges that remain, both in terms of scalability and semantic evaluation. Finally, we discuss how these results can motivate new lines of investigation, outline concrete opportunities for extending FSAM and ActiMine-GNN, and report on what we have learned from our experiments and discuss how this work could influence the direction in the field of Explainable AI for graph-structured data.

6.2 The Functional Semantic Activation Mapping Graph Representation

In Chapter 3, we introduced and experimented with a new approach to analyse and explain the internal behaviour of Graph Neural Networks (GNN). The proposed methodology extracts and represents knowledge from a trained GNN in a graph representation called the Functional Semantic Activation Mapping (FSAM) graph. FSAM assumes a fixed attributed graph $G = (V, E, X)$ in a transductive node classification setting. A GCN processes this graph and, at each hidden layer ℓ , produces an activation matrix $A^{(\ell)} \in \mathbb{R}^{n \times h_\ell}$, where each column gives the activation profile of one neuron across all nodes. These activations define a semantic graph for each layer: neurons become nodes, and weighted edges capture strong co-activations between their activation profiles. In practice, FSAM measures neuron–neuron co-activation using Spearman’s rank correlation, then applies global weight thresholding to keep only the strongest dependencies. In addition, FSAM computes point–biserial correlations between neurons and predicted class indicators, which link hidden units to output classes in a one-versus-rest environment. The resulting graphs are semantic in the sense that their communities tend to align with classes or coherent groups of classes that the GNN has learned.

This representation provides a layerwise view of how the class-specific structure is organised inside the network. At earlier layers, communities often reflect broad patterns; at intermediate depths, they become more class-aligned; and, beyond a certain depth, they may start to merge as oversmoothing takes effect. By construction, FSAM does not change the underlying GNN; it reads off activations after training and converts them into graphs that can be analysed with community detection and overlap measures.

The experiments reported in Chapters 3 and 4 show that FSAM is compatible with the knowledge encoded by trained GNNs. First, we observed that neurons with high point–biserial correlation to a given class are concentrated in specific commu-

nities, and that layers with clear, well-separated communities support better class discrimination. Secondly, we found that class pairs with many shared FSAM nodes or overlapping communities are also those that incur the largest number of misclassifications. This positive association between community overlap and class-pair mistakes indicates that FSAM captures structure that is relevant for prediction, rather than incidental activation noise. It also suggests that overlapping neurons are natural candidates for further inspection or targeted regularisation.

In Chapter 5, this activation-centred view is further extended by using FSAM-derived signals as input to rule mining, closing the loop between neural activity and symbolic explanations. The extensive evaluation conducted across different domains of datasets supports the adequacy of the FSAM representation. We began by showing that FSAM can identify neurons that contribute strongly to particular classes and that these neurons cluster into well-defined communities at suitable depths. We then observed a positive association between community overlap and class-pair mistakes, in that pairs of classes that share many FSAM nodes or communities tend to exhibit higher misclassification rates. These findings indicate that FSAM captures meaningful aspects of the knowledge encoded in the GNN and that graph-theoretic analysis of FSAM graphs can help identify and reflect where and why confusions arise. FSAM also enabled a semantic perspective on depth and oversmoothing. By comparing activation graphs across layers, and across models with different depths, we see a clear pattern. Beyond a certain, task-dependent depth, adding more layers makes the activations increasingly similar to each other. The communities in the activation graph become weaker and less clearly separated. This can happen even when test accuracy stays the same or slightly improves. In turn, it suggests that the discriminative power of the representations may degrade while standard performance metrics still appear satisfactory. FSAM makes this phenomenon visible by revealing where the class structure is established, where it starts to merge, and where it effectively collapses, thus providing a useful tool for diagnosing when we have reached the point of diminishing returns, so no more depth is required.

6.3 Measuring Semantic Collapse: Depth, Oversmoothing and Discriminative Power

In Chapter 4, we investigated how FSAM can be used not only to analyse and explain the inner workings of a GNN, but also to measure how this organisation changes as the network becomes deeper. The first aim was to validate whether FSAM consistently captures model behaviour across different layer configurations. For this, we trained GNNs with one to four layers on six benchmark datasets and, for each depth, constructed FSAM graphs per layer, detected communities, and related these to both accuracy and misclassification patterns. The results showed that across depths and datasets, community structures aligned with class organisation, and changes in FSAM coherence mirrored changes in error structure. This provided further evidence that FSAM reflects the knowledge encoded by the GNN.

The second aim was to examine how depth affects discriminative power. By comparing accuracy across the different layers with the semantic structure of FSAM-derived measures such as community coherence, class–community alignment, and inter-class overlap, we observed a consistent pattern. Initial increases in depth improved both performance and FSAM structure, communities became more class-specific, and separability increased. Beyond a task-dependent point, however, semantic organisation began to deteriorate. FSAM communities merged, overlap between class communities increased, and class–pair confusions rose, even in cases where headline accuracy plateaued or showed only modest gains. This behaviour is consistent with oversmoothing as deeper layers produce more homogeneous representations that are less able to distinguish between classes.

A step further in this analysis was to relate quantitative FSAM signals to practical decisions about model design. The depth at which community coherence peaks identifies a useful operating range, and the class pairs whose communities overlap most indicate where additional supervision or data curation may be most beneficial. Beyond this range, extra layers tend to reduce the separability of FSAM commu-

nities and increase confusion, signalling that the model is becoming unnecessarily deep. This opens up the possibility of using FSAM-based measures to guide early stopping, depth pruning, or even adaptive-depth GNNs that stop message passing once semantic performance converges.

These observations also point to several directions for future work. Our study focused on a family of message-passing GNNs and static graphs; it remains to be seen how semantic collapse behaves in heterogeneous or temporal graphs, or under alternative aggregation and normalisation schemes. Moreover, we varied depth in isolation; interactions with width, regularisation, and training dynamics may require joint analysis. Nevertheless, the evidence gathered in this thesis indicates that measuring semantic collapse through FSAM provides a useful complement to standard metrics and offers a principled way to understand when deeper GNNs genuinely learn more, and when they smooth away the distinctions that make explanations meaningful.

6.4 Generating Local Rule Based Explanations through Semantic Activation Mining

In Chapter 5, we moved from global activation-centred diagnostics to local, rule-based explanations. Building on the FSAM activation extraction pipeline, we proposed ActiMine-GNN. This method mines compact decision rules from hidden-layer activations and instantiates them as factual explanations for individual predictions. In this setting, binarised activations act as a bridge between the continuous internal state of the GNN and a symbolic rule space, so that local behaviour can be described in terms that are easier to inspect and use to highlight the parts of a graph that are most relevant for a particular decision.

The starting point is the binarised activation matrices obtained from a trained GCN on a binary graph classification task. For each hidden layer, these activations are treated as transactions over neurons within r -hop ego-neighbourhoods. From these

transactions, we mine frequent co-activation patterns and use them as features for a shallow decision tree surrogate. Root-to-leaf paths then define short activation rules, which can be instantiated on a given graph to select the nodes whose activation patterns satisfy the rule’s antecedent. The resulting subgraph serves as a factual explanation, as it identifies which recurrent activation patterns were present and how they supported the predicted class. In this way, ActiMine–GNN complements FSAM, as where FSAM describes where and when discriminative structure appears in the network in Chapter 3, ActiMine–GNN makes part of that structure explicit as local, human-readable rules.

We evaluated the approach on three standard benchmarks for GNN explainability, BA-2Motifs, AIDS, and BBBP. These datasets provide a mix of ground-truth motifs and chemically meaningful structure, allowing us to assess both the faithfulness and plausibility of the explanations. Quantitatively, we measured fidelity at the stated sparsity budget, rule coverage, and alignment between the surrogate tree and the underlying GNN (RQ6). Across all three datasets, ActiMine–GNN achieved high fidelity while keeping rule sets compact and stable under changes in support and pattern-cap parameters, supporting the hypothesis that local activation rules can faithfully account for the GNN’s decisions without sacrificing sparsity.

Beyond aggregate scores, we carried out a qualitative analysis of the rules and their instantiations (RQ7). For BA-2Motifs, rules selected by the surrogate corresponded to activation patterns that consistently aligned with the planted motifs and their instantiation produced subgraphs that matched the ground-truth structures used to define the labels. On AIDS and BBBP, instantiated rules highlighted subgraphs that coincided with known functional groups or substructures linked to antiviral activity or permeability, as reported in the chemical literature.

In both cases, the rules provided clear factual explanations for individual predictions; they identified which recurrent activation patterns were present in a given graph and how these patterns contributed to a positive or negative outcome. This supports the view that activation-based rules can act as an interpretable interface

between the GNN and domain expertise.

The results in Chapter 5 indicate that semantic activation mining provides a practical route from hidden-layer activity to local, rule-based explanations. Taken together with the global analyses in Chapters 3 and 4, ActiMine-GNN shows that it is possible to connect neuron-level signals, graph structure, and human-understandable rules in a single neurosymbolic pipeline. In the next section, we discuss how these contributions fit within the wider landscape of Explainable AI for GNNs and outline directions for extending both FSAM and ActiMine-GNN beyond the scope of this thesis.

6.5 Challenges and Opportunities Ahead

In this thesis, we have developed a novel approach to explain GNN by combining activation-centred representation to understand how GNNs organise class structure in depth, and ActiMine-GNN as a rule-based framework to extract compact, local activation explanations. Together, they suggest one possible way of combining activation analysis with neurosymbolic integration principles for graph explainability. Our research paves the way for several interesting lines of investigation. We believe that these are worth exploring, which are listed below.

Concrete methodological extensions: Several specific extensions follow from choices made in this thesis. Neuron dependence is measured by Spearman correlation, which captures monotonic association only, so mutual information would be a natural alternative to compare against; the analysis is anchored to predicted labels, so repeating it with ground-truth labels would separate representation quality from model error; community detection is run once, so reporting partition agreement across seeds and thresholds would establish how much weight individual assignments bear; and our notion of semantics is model-internal, so a grounding protocol based on expert annotation of top-activating nodes would test correspondence with human concepts. Two settings also remain open: an inductive evaluation on unseen graphs,

and extension beyond GCN to GraphSAGE, GAT, and GIN.

Bias and failure-mode analysis: A recurring theme in our experiments is that FSAM and ActiMine-GNN often show where the model mixes up classes or relies on overlapping communities. This does more than simply describe what the model is doing. It also points to a way of detecting systematic biases and recurring failure modes in a more structured manner. FSAM communities and class-pair overlaps could be used to identify regions of the input space where certain classes are consistently confused, or where particular subpopulations receive systematically different treatment. ActiMine rules, in turn, offer a way to describe these patterns in terms of recurring activation configurations and associated subgraphs. A more fine-grained study of these structures, possibly combined with group-level annotations or fairness labels when available, could help reveal biases in both the data and the learned representation. Analysing paths and motifs within FSAM graphs and examining the distribution of rule support across sensitive subgroups are natural next steps.

Scaling and efficiency: Both FSAM and ActiMine-GNN build on activation analysis, which becomes more demanding as graphs increase in size and models grow deeper. In the case of FSAM, this involves estimating pairwise dependencies and running community detection over hidden units. ActiMine-GNN, in turn, depends on frequent pattern mining and the training of surrogate models. Extending both FSAM and ActiMine-GNN to much larger, dynamic, or heterogeneous graphs remains an open challenge. On the FSAM side, this involves developing more scalable dependence estimation, sparsification, and community detection procedures, and assessing the robustness of semantic signals under variations in depth, feature dimension, and training conditions. On the ActiMine side, further work is still needed. We need to understand how binarisation thresholds, pattern support levels, and surrogate model complexity affect fidelity and sparsity on larger benchmarks. It is also important to evaluate the approach more systematically on multiclass and multi-label tasks.

Independent validation of semantic signals: The depth-wise analysis in this thesis characterises the consequences of oversmoothing for class-selective structure, using measures derived from the FSAM graph together with the model’s classification errors. A complementary direction is to relate these signals to indicators of oversmoothing defined directly on the embedding geometry, such as Dirichlet energy, mean average distance, or embedding variance. Computing such an indicator per layer and testing whether it co-varies with FSAM community coherence would connect two levels of description that are currently treated separately: the geometry of the representation and the organisation of class-selective activation. It would also provide external corroboration of the depth at which semantic collapse is identified, independently of the construction used to measure it.

Human-in-the-loop Validation: Both FSAM visualisations and ActiMine masks are designed to be interpretable to domain experts, but in this thesis, we have relied mainly on quantitative metrics and limited qualitative inspection. A natural extension of the step is to place these explanations in a human-in-the-loop workflow. In such a setting, experts can judge whether the global patterns are plausible and useful, for example the depth at which semantic collapse appears. They can also review local rationales, such as which substructures are highlighted in molecular graphs. Their feedback could then be used to adjust thresholds, choose between different community partitions, or prune and refine the rule sets. In cases where the explanation is uncertain, for example when several rules apply with similar support, it may be better to ask for expert input directly. Uncertainty sampling and active querying of experts could then help to produce explanations that are more informative and more actionable. This would align our approach with broader efforts in human-centred, trustworthy AI while preserving the model grounded nature of the explanations.

Knowledge injection and adaptive architectures: In this thesis, FSAM and ActiMine-GNN have been used as post-hoc tools that analyse trained models with-

out influencing the training process. A longer-term direction is to close the loop and use the extracted information to guide learning. Depth wise FSAM signals could, for example, inform early stopping or architecture selection by indicating when additional layers no longer provide clear semantic benefits. Class-pair overlaps and misclassification patterns might be used to reweight training examples or to construct auxiliary loss terms that encourage better separation of problematic classes. Likewise, rules from ActiMine-GNN could be treated as candidate constraints, either to regularise the model towards more consistent behaviour or to flag cases where the model systematically violates expert-endorsed patterns. Designing such knowledge injection mechanisms while preserving model capacity and avoiding confirmation bias is a non-trivial but promising avenue.

Overall, the contributions in this thesis provide a first step towards a neurosymbolic toolkit for GNN explainability, combining global activation graphs, depth-wise semantic diagnostics, and local rule-based explanations. The challenges outlined above also show how this toolkit could be taken further.

Appendices

Appendix A

Publications on Work from this Thesis

A.1 Journal Publications

- Raj, Kislay, and Alessandra Mileo. "Towards Semantic Understanding of Graph Neural Network Layers Embedding with Functional Semantic Activation Mapping." *Neurosymbolic Artificial Intelligence 2* (2026): 29498732251408218

A.2 Conference Publication

- Raj, Kislay. "A neuro-symbolic approach to enhance interpretability of graph neural network through the integration of external knowledge." *Proceedings of the 32nd ACM international conference on information and knowledge management*. 2023.
- Raj, K., Kumar, T., Mileo, A., & Bendeche, M. (2024, August). OxML Challenge 2023: Carcinoma classification using data augmentation. *IET Conference Proceedings CP887*, Vol. 2024, No. 10. Stevenage, UK: The Institution of Engineering and Technology.
- Raj, Kislay, and Alessandra Mileo. "Towards Understanding Graph Neural

Networks: Functional-Semantic Activation Mapping." International Conference on Neural-Symbolic Learning and Reasoning. Cham: Springer Nature Switzerland, 2024.

- Raj, K., & Mileo, A. (2025). ActiMine-GNN: Activation Mining and Interpretability in Graph Neural Networks. In XAI-KRKG/UCEC-XAI@ ECAI (pp. 22-31).
- Raj, K., & Mileo, A. (2025). Neurosymbolic Methods for Explainable Graph Neural Networks: A Survey.

Appendix B

Awards

- First place in the ML x Cases 2023 competition, organised by AI for Global Goals, CIFAR, and the University of Oxford.

Bibliography

- [1] Burouj Armgaan et al. “Graphtrail: Translating GNN predictions into human-interpretable logical rules”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 123443–123470.
- [2] Alejandro Barredo Arrieta et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. In: *arXiv preprint arXiv:1910.10045* 1 (2019).
- [3] Steve Azzolin et al. “Global explainability of gnns via logic combination of learned concepts”. In: *arXiv preprint arXiv:2210.07147* (2022).
- [4] Federico Baldassarre and Hossein Azizpour. “Explainability techniques for graph convolutional networks”. In: *arXiv preprint arXiv:1905.13686* (2019).
- [5] Pietro Barbiero et al. “Relational concept bottleneck models”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 77663–77685.
- [6] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.
- [7] Margarita Bugueño, Russa Biswas, and Gerard de Melo. “Explainable AI for Graph-Based Learning: A Survey Beyond Graph Neural Networks”. Survey paper. Under review at *ACM Transactions on Intelligent Systems and Technology (TIST)*. Feb. 2026. URL: <https://hal.science/hal-04660442>.

- [8] Jianbo Chen et al. “Learning to explain: An information-theoretic perspective on model interpretation”. In: *International conference on machine learning*. PMLR. 2018, pp. 883–892.
- [9] Brandon C Colelough and William Regli. “Neuro-symbolic AI in 2024: A systematic review”. In: *arXiv preprint arXiv:2501.05435* (2025).
- [10] Tirtharaj Dash, Ashwin Srinivasan, and Lovekesh Vig. “Incorporating symbolic domain knowledge into graph neural networks”. In: *Machine Learning* 110.7 (2021), pp. 1609–1636.
- [11] Lauren Nicole DeLong, Ramon Fernández Mir, and Jacques D Fleuriot. “Neurosymbolic AI for reasoning over knowledge graphs: A survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2024).
- [12] Honghua Dong et al. “Neural logic machines”. In: *arXiv preprint arXiv:1904.11694* (2019).
- [13] Eric Ferreira dos Santos and Alessandra Mileo. “From disentangled representation to concept ranking: Interpreting deep representations in image classification tasks”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2022, pp. 322–335.
- [14] Thorben Funke, Megha Khosla, and Avishek Anand. “Hard masking for explaining graph neural networks”. In: (2021).
- [15] Artur d’Avila Garcez and Luis C Lamb. “Neurosymbolic ai: The 3 rd wave”. In: *Artificial Intelligence Review* 56.11 (2023), pp. 12387–12406.
- [16] Dario Garcia-Gasulla et al. “Building graph representations of deep vector embeddings”. In: *Proceedings of the 2nd Workshop on Semantic Deep Learning (SemDeep-2)*. 2017, pp. 9–15.
- [17] Robert Geirhos et al. “Generalisation in humans and deep neural networks”. In: *Advances in neural information processing systems* 31 (2018).
- [18] Chuqin Geng et al. “Extracting Interpretable Logic Rules from Graph Neural Networks”. In: *arXiv e-prints* (2025), arXiv–2503.

- [19] Leilani H Gilpin et al. “Explaining explanations: An overview of interpretability of machine learning”. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89.
- [20] Martin Grohe. “The Logic of Graph Neural Networks”. In: *Proceedings of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '21)*. IEEE, 2021, pp. 1–17. DOI: 10.1109/LICS52264.2021.9470677. URL: <https://doi.org/10.1109/LICS52264.2021.9470677>.
- [21] David Gunning et al. “XAI—Explainable artificial intelligence”. In: *Science robotics* 4.37 (2019), eaay7120.
- [22] William L Hamilton, Rex Ying, and Jure Leskovec. “Inductive representation learning on large graphs”. In: *arXiv preprint arXiv:1706.02216* (2017).
- [23] Vitor AC Horta et al. “Extracting knowledge from deep neural networks through graph analysis”. In: *Future Generation Computer Systems* 120 (2021), pp. 109–118.
- [24] Qiang Huang et al. “Graphlime: Local interpretable model explanations for graph neural networks”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).
- [25] M Jaleed Khan et al. “A survey of neurosymbolic visual reasoning with scene graphs and common sense knowledge”. In: *Neurosymbolic Artificial Intelligence* 1 (2025), NAI–240719.
- [26] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).
- [27] Yiqiao Li et al. “A survey of explainable graph neural networks: Taxonomy and evaluation metrics”. In: *arXiv preprint arXiv:2207.12599* (2022).
- [28] Meng Liu, Hongyang Gao, and Shuiwang Ji. “Towards deeper graph neural networks”. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, pp. 338–348.

- [29] A. Longa et al. “Explaining the Explainers in Graph Neural Networks: a Comparative Study”. In: *ACM Computing Surveys* (2024). DOI: 10.1145/3696444.
- [30] Antonio Longa et al. “Graph Neural Networks for temporal graphs: State of the art, open challenges, and opportunities”. In: *arXiv preprint arXiv:2302.01018* (2023). DOI: 10.48550/arXiv.2302.01018.
- [31] Dongsheng Luo et al. “Parameterized explainer for graph neural network”. In: *Advances in neural information processing systems* 33 (2020), pp. 19620–19631.
- [32] Julian McAuley et al. “Image-based recommendations on styles and substitutes”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 2015, pp. 43–52.
- [33] Alessandra Mileo. “Towards a neuro-symbolic cycle for human-centered explainability”. In: *Neurosymbolic Artificial Intelligence* 1 (2025), NAI–240740.
- [34] Christopher Morris et al. “TUDataset: A collection of benchmark datasets for learning with graphs”. In: *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*. 2020. URL: <https://arxiv.org/abs/2007.08663>.
- [35] P. Müller et al. *DT+GNN: A Fully Explainable Graph Neural Network using Decision Trees*. arXiv preprint arXiv:2205.13234. 2022. DOI: 10.48550/arxiv.2205.13234.
- [36] Galileo Namata et al. “Query-driven active surveying for collective classification”. In: *10th international workshop on mining and learning with graphs*. Vol. 8. 2012, p. 1.
- [37] Mauparna Nandan, Soma Mitra, and Debashis De. “GraphXAI: a survey of graph neural networks (GNNs) for explainable AI (XAI)”. In: *Neural Computing and Applications* 37.17 (2025), pp. 10949–11000.
- [38] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. “Feature visualization”. In: *Distill* 2.11 (2017), e7.

- [39] Xinjun Pei et al. “Efficient Privacy Preserving Graph Neural Network for Node Classification”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096911.
- [40] Alexander Pluska et al. “Logical distillation of graph neural networks”. In: *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning*. KR '24. Hanoi, Vietnam, 2024. ISBN: 978-1-956792-05-8. DOI: 10.24963/kr.2024/86. URL: <https://doi.org/10.24963/kr.2024/86>.
- [41] Phillip E Pope et al. “Explainability methods for graph convolutional neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 10772–10781.
- [42] Kislay Raj and Alessandra Mileo. “ActiMine-GNN: Activation Mining and Interpretability in Graph Neural Networks.” In: *XAI-KRKG/UCEC-XAI@ECAI*. 2025, pp. 22–31.
- [43] Kislay Raj and Alessandra Mileo. “Towards Semantic Understanding of Graph Neural Network Layers Embedding with Functional Semantic Activation Mapping”. In: *Neurosymbolic Artificial Intelligence 2* (2026), p. 29498732251408218.
- [44] Kislay Raj and Alessandra Mileo. “Towards Understanding Graph Neural Networks: Functional-Semantic Activation Mapping”. In: *International Conference on Neural-Symbolic Learning and Reasoning*. Springer. 2024, pp. 98–106.
- [45] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. “A survey on oversmoothing in graph neural networks”. In: *arXiv preprint arXiv:2303.10993* (2023).
- [46] M. Saarela and V. Podgorelec. “Recent Applications of Explainable AI (XAI): A Systematic Literature review”. In: *Applied Sciences* 14.19 (2024), p. 8884. DOI: 10.3390/app14198884.

- [47] E. Şahin, N. N. Arslan, and D. Özdemir. “Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning”. In: *Neural Computing and Applications* (2024). DOI: 10 . 1007 / s00521 - 024 - 10437 - 2.
- [48] Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. “Interpreting graph neural networks for nlp with differentiable edge masking”. In: *arXiv preprint arXiv:2010.00577* (2020).
- [49] Thomas Schnake et al. “Higher-order explanations of graph neural networks via relevant walks”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.11 (2021), pp. 7581–7596.
- [50] Robert Schwarzenberg et al. “Layerwise relevance visualization in convolutional text graph classifiers”. In: *arXiv preprint arXiv:1909.10911* (2019).
- [51] Ramprasaath R Selvaraju et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.
- [52] Prithviraj Sen et al. “Collective classification in network data”. In: *AI magazine* 29.3 (2008), pp. 93–93.
- [53] Oleksandr Shchur et al. “Pitfalls of Graph Neural Network Evaluation”. In: *Relational Representation Learning Workshop, NeurIPS 2018*. 2018.
- [54] Deep Graph Library (DGL) Team. *TreeGridDataset*. 2025.
- [55] E. Tsamoura, T. Hospedales, and L. Michael. “Neural-Symbolic Integration: A compositional perspective”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2021, pp. 5051–5060. DOI: 10.1609/aaai.v35i6.16639.
- [56] Petar Veličković et al. “Graph attention networks”. In: *arXiv preprint arXiv:1710.10903* (2017).
- [57] Luca Veyrin-Forrer et al. “On GNN explainability with activation rules”. In: *Data Mining and Knowledge Discovery* 38.5 (2024), pp. 3227–3261.

- [58] Minh Vu and My T Thai. “Pgm-explainer: Probabilistic graphical model explanations for graph neural networks”. In: *Advances in neural information processing systems* 33 (2020), pp. 12225–12235.
- [59] Lilapati Waikhom and Ripon Patgiri. “A survey of graph neural networks in various learning paradigms: methods, applications, and challenges”. In: *Artificial Intelligence Review* 56.7 (2023), pp. 6295–6364.
- [60] Xiang Wang et al. “Causal screening to interpret graph neural networks”. In: (2021).
- [61] Zhenqin Wu et al. “MoleculeNet: a benchmark for molecular machine learning”. In: *Chemical science* 9.2 (2018), pp. 513–530.
- [62] Zonghan Wu et al. “A comprehensive survey on graph neural networks”. In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.
- [63] Weiting Xi et al. “A Graph Partitioning Algorithm Based on Graph Structure and Label Propagation for Citation Network Prediction”. In: *International Conference on Knowledge Science, Engineering and Management*. Springer. 2023, pp. 289–300.
- [64] Keyulu Xu et al. “How powerful are graph neural networks?” In: *arXiv preprint arXiv:1810.00826* (2018).
- [65] Zezhong Xu et al. “Ruleformer: Context-aware rule mining over knowledge graph”. In: *Proceedings of the 29th international conference on computational linguistics*. 2022, pp. 2551–2560.
- [66] Xiaoran Yan et al. “Weight thresholding on complex networks”. In: *Physical Review E* 98.4 (2018), p. 042304.
- [67] Pinar Yanardag and SVN Vishwanathan. “Deep graph kernels”. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1365–1374.

- [68] Z. Ye et al. “A comprehensive survey of Graph Neural networks for knowledge Graphs”. In: *IEEE Access* 10 (2022), pp. 75729–75741. DOI: 10.1109/access.2022.3191784.
- [69] Rex Ying et al. “Gnnexplainer: Generating explanations for graph neural networks”. In: *Advances in neural information processing systems* 32 (2019).
- [70] Jiaxuan You et al. “Graph structure of neural networks”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 10881–10891.
- [71] Zhaoning Yu and Hongyang Gao. “Mage: Model-level graph neural networks explanations via motif-based graph generation”. In: *arXiv preprint arXiv:2405.12519* (2024).
- [72] H. Yuan et al. “XGNN: Towards Model-Level Explanations of Graph Neural Networks”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. New York, NY, USA: ACM, Aug. 20, 2020. DOI: 10.1145/3394486.3403085.
- [73] Hao Yuan and Shuiwang Ji. “Structpool: Structured graph pooling via conditional random fields”. In: *Proceedings of the 8th International Conference on Learning Representations*. 2020.
- [74] Hao Yuan et al. “Explainability in graph neural networks: A taxonomic survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).
- [75] Hao Yuan et al. “On explainability of graph neural networks via subgraph explorations”. In: *International conference on machine learning*. PMLR. 2021, pp. 12241–12252.
- [76] He Zhang et al. “Trustworthy graph neural networks: Aspects, methods, and trends”. In: *Proceedings of the IEEE* 112.2 (2024), pp. 97–139.
- [77] Quanshi Zhang et al. “Extraction of an explanatory graph to interpret a CNN”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.11 (2020), pp. 3863–3877.

- [78] Quanshi Zhang et al. “Interpreting cnns via decision trees”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6261–6270.
- [79] Xin Zhang and Victor S Sheng. “Neuro-symbolic ai: Explainability, challenges, and future trends”. In: *arXiv preprint arXiv:2411.04383* (2024).
- [80] Yue Zhang, David Defazio, and Arti Ramesh. “Relex: A model-agnostic relational model explainer”. In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 1042–1049.