Evaluating Graph Neural Additive Networks for Multi-Label Node Classification

Interpretability and Performance Across High- and Low-Homophily Graphs

Arsenie Vlas

Supervisors: Elena Congeduti, Megha Khosla

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

June 2026

Name of the student:	Arsenie Vlas
Final project course:	CSE3000 Research Project
Thesis committee:	Elena Congeduti, Megha Khosla, Christoph Lofi

Abstract

Graph Neural Additive Networks (GNANs) extend generalised additive models to graph-structured data, providing interpretability by design rather than through post-hoc explanation. GNANs have been studied on multi-class node classification, but not in the multi-label setting, where a single node may belong to several categories at once. This paper presents the first adaptation and evaluation of GNAN for multi-label node classification. We replace the softmax output with a per-label sigmoid and give the distance function a per-label output, and we benchmark the adapted model against standard baselines on two real-world graphs that span a high- and a low-homophily regime, reporting Average Precision (AP) as the primary metric. We then analyse the learned shape functions and distance function to ask whether the built-in explanations are meaningful. GNAN is competitive with strong message-passing graph neural networks on the high-homophily graph, coming within about four AP points of the best baseline, but it drops to the lower end of the baselines on the low-homophily graph. Its learned distance function adapts to label homophily: a steep, reproducible local decay when neighbours are informative, and a flat, unstable profile when they are not. These results characterise when GNAN's additive structure is an advantage and when it is a limitation, and they demonstrate the practical value of interpretability by design in the multi-label graph setting.

1 Introduction

Node classification is one of the most widely studied tasks in graph machine learning. Given a partially labelled graph, the goal is to infer the labels of unlabelled nodes from their features and their structural relationships with neighbouring nodes. The task has practical relevance across a broad range of domains, including social network analysis, biological network analysis, drug discovery, and recommendation systems [1]. In recent years, Graph Neural Networks (GNNs), in particular variants of graph convolutional networks, have become the dominant approach, achieving strong results on both node-level and graph-level benchmarks [1].

Despite these advances, the majority of GNN research has concentrated on multi-class settings, where each node is assigned exactly one label. Real-world graphs, however, frequently exhibit a richer structure in which nodes belong to several categories simultaneously. Protein–protein interaction networks, where each protein may be associated with multiple biological functions or diseases, and social networks, where users carry several interest labels at once, are canonical examples of this multi-label regime [1]. The benchmark suite introduced by Zhao et al. [1] provides the first systematic evaluation of node classification methods on multi-label datasets, revealing that several design choices that work well in the single-label setting do not transfer straightforwardly. The multi-label scenario therefore remains relatively underexplored, and a thorough understanding of which model properties lead to strong performance is still lacking.

A second open problem is interpretability. Standard GNNs function as black-box models: they can produce accurate predictions, but the reasoning behind individual decisions is not directly accessible. In high-stakes domains such as biology and medicine, this opacity is a significant limitation: understanding *why* a model assigns a particular set of labels to a protein is often as important as the assignment itself [2]. Post-hoc explanation methods, such as GNNExplainer [3] and its successors, offer some insight after the fact, but they provide no correctness guarantees and may fail to expose hidden model biases [2]. GNAN, proposed by Bechler-Speicher et al. [2], addresses this limitation by extending generalised additive models to graph-structured data. By applying separate learned shape functions to each input feature in isolation and weighting neighbourhood contributions through a learned distance function, GNAN produces a model that is interpretable by design: explanations are a direct read-off of the model parameters rather than a post-hoc approximation.

While GNAN has been evaluated on standard multi-class node classification benchmarks and shown to match the accuracy of black-box GNNs [2], its behaviour in the multi-label setting has not been studied. This work addresses that gap. Adapting GNAN for multi-label classification requires replacing the softmax output layer with a sigmoid activation, which treats each label independently. Beyond the adaptation itself, the additive structure of GNAN, which deliberately prevents non-linear interactions between features, raises a natural question about when this architectural constraint is a virtue and when it is a liability. We hypothesise that GNAN performs well on datasets with high label homophily and informative features, where clean neighbourhood signals are sufficient for prediction, but may be at a disadvantage on low-homophily datasets where complex feature interactions contribute to discriminative power.

The central research question of this work is:

> *How does Graph Neural Additive Network (GNAN) perform on different multi-label node classification datasets, and what do the resulting explanations reveal about the data?*

To answer this question, we pursue five sub-questions. First, we verify that GNAN can be successfully adapted for multi-label output and produces stable, non-trivial results across multiple random seeds. Second, we benchmark the adapted GNAN against established baselines on real-world multi-label datasets from the multi-label benchmark suite of Zhao et al. [1], measuring micro-F1, macro-F1, and Average Precision. Third, we systematically vary label homophily using synthetic graphs to test whether GNAN's relative performance follows the predicted pattern. Fourth, we vary the ratio of informative to noisy features to characterise GNAN's sensitivity to feature quality. Fifth, for at least one dataset on which GNAN achieves reasonable predictive performance, we extract and interpret the learned shape functions and distance function to assess whether the model's explanations are meaningful in the context of the data.

The main contributions of this paper are: (i) the first adaptation and evaluation of GNAN for multi-label node classification; (ii) a systematic empirical study of GNAN's performance across datasets with contrasting homophily; and (iii), the central contribution, an analysis of GNAN's built-in explanations on

real-world data that shows *what they reveal about the model's learned strategy and about each dataset*, backed by quantitative checks of cross-seed stability and predictive validity, demonstrating the practical value of interpretability by design in graph learning. Sub-questions three and four, which require synthetic graphs with tunable properties, are scoped as future work and discussed in Section 6.

The remainder of this paper is organised as follows. Section 2 reviews the necessary background. Section 3 describes the methodology, including the multi-label adaptation of GNAN, the baselines, the datasets, and the evaluation protocol. Section 4 presents the experimental results. Section 5 analyses GNAN's explanations. Section 6 discusses the findings, Section 7 addresses responsible research, and Section 8 concludes with directions for future work.

2 Background

2.1 Graph Neural Networks for Node Classification

Graph Neural Networks have become the dominant paradigm for learning on graph-structured data [2]. In the node classification setting, a GNN takes as input a graph $G = (\mathcal{V}, \mathcal{E})$ with a node feature matrix $X \in \mathbb{R}^{|\mathcal{V}| \times d}$ and produces a label prediction for each node. The vast majority of GNN architectures follow the *message-passing* framework [2], in which node representations are updated iteratively by aggregating information from neighbouring nodes:

$$\mathbf{z}_i^{(k)} = \text{AGGREGATE}\Big(\mathbf{x}_i^{(k-1)},\, \big\{\mathbf{x}_j^{(k-1)} : j \in \mathcal{N}(i)\big\}\Big), \quad \mathbf{x}_i^{(k)} = \text{TRANSFORM}\Big(\mathbf{z}_i^{(k)}\Big), \tag{1}$$

where $\mathcal{N}(i)$ denotes the one-hop neighbourhood of node i and k indexes the layer. After L layers of aggregation, a classification head maps the final representation $\mathbf{x}_i^{(L)}$ to a label distribution.

The Graph Convolutional Network (GCN) of Kipf and Welling [4] is the canonical instance of this framework. It implements degree-normalised neighbourhood averaging, and has been shown to perform particularly well on graphs where connected nodes tend to share the same label, a property known as *label homophily*. GraphSAGE [5] extends the message-passing idea to inductive settings by sampling a fixed-size subset of neighbours at each layer. Both GCN and GraphSAGE serve as baselines in the present work; their strong performance on standard benchmarks makes them natural points of comparison for a structurally different model such as GNAN.

A well-known limitation of message-passing GNNs is their sensitivity to homophily. On *heterophilic* graphs, where neighbouring nodes tend to carry different labels, standard neighbourhood aggregation can actually hurt performance by mixing together dissimilar feature signals. The multi-label setting, as discussed below, introduces a further complication: the classical binary notion of a homophilic or heterophilic edge does not straightforwardly generalise when nodes carry multiple, overlapping labels [1].

2.2 Multi-Label Node Classification

In the standard node classification formulation, each node i is associated with exactly one label $y_i \in \{1, \ldots, C\}$. Multi-label classification generalises this by allowing each node to carry a *set* of labels, represented as a binary vector $\mathbf{y}_i \in \{0, 1\}^C$ where multiple entries may be active simultaneously. The prevailing approach is to decompose the problem into C independent binary classification tasks, one per label, applying a sigmoid activation at the output layer [1]. While straightforward, this formulation ignores potential correlations between co-occurring labels.

Multi-label graphs arise naturally in biology, where proteins are associated with multiple functions or diseases, and in co-authorship graphs, where a researcher's work may span several fields. Despite this practical relevance, multi-label node classification has received comparatively little systematic attention. Zhao et al. [1] identify a key reason: a single high-quality multi-label benchmark did not previously exist. Their Multi-Label Graph Node Classification (MLGNC) suite addresses this by collecting nine datasets, including co-authorship graphs, biological interaction networks, and synthetic graphs with tunable properties, and establishing a standardised evaluation protocol.

A conceptually important contribution of Zhao et al. [1] is the observation that the standard definition of label homophily does not transfer directly to the multi-label setting. In the multi-class case, an edge is homophilic if its two endpoints share the same single label. In the multi-label case, two nodes may share

some but not all of their labels, so whether an edge counts as homophilic becomes a matter of degree, set by the fraction of labels its two endpoints have in common, rather than an all-or-nothing yes/no property. Zhao et al. redefine homophily for multi-label graphs as the mean pairwise label similarity across edges, and introduce Cross-Class Neighbourhood Similarity (CCNS) to capture the degree to which the label distributions of nodes and their neighbours overlap. These definitions reveal that many real-world multi-label graphs have low homophily under the new measure.

This work uses two real-world datasets from the MLGNC suite, chosen to represent contrasting regimes. **DBLP** is a co-authorship graph with roughly 28,000 nodes, 68,000 edges, 300 TF-IDF word features, and four labels, exhibiting high label homophily ($r_{\text{hom}} = 0.76$). GCN achieves strong results on this dataset under the MLGNC benchmark [1], making it a suitable environment to test GNAN under favourable conditions. **PCG** is a protein–phenotype interaction network with approximately 3,000 nodes, 37,000 edges, 32 features, and 15 labels, and exhibits low label homophily ($r_{\text{hom}} = 0.17$). Its dense connectivity and overlapping label structure make it a more challenging benchmark, suitable for testing GNAN's performance limits.

2.3 Interpretable Graph Learning and GNAN

As graph-based models are deployed in high-stakes domains such as medicine and fraud detection, the question of interpretability has become increasingly important. Bechler-Speicher et al. [2] draw a precise distinction between *interpretability* and *explainability*: interpretable models are comprehensible by design, whereas explainability methods are post-hoc tools applied to opaque models after training. Post-hoc methods such as GNNExplainer [3] identify subgraphs or features that are important for a specific prediction, but they provide no correctness guarantees: the explanation is an approximation of the model's behaviour, not a faithful description of it [2].

The interpretability-by-design approach roots its transparency in the model architecture itself. Generalised Additive Models (GAMs), introduced by Hastie and Tibshirani [6], achieve this by expressing predictions as a sum of univariate functions of individual features:

$$\hat{y} = \sigma\left(\sum_{k=1}^{d} f_k(x_k)\right), \tag{2}$$

where σ is an activation function and each f_k is a *shape function* that can be visualised as a curve. Because the features contribute independently and additively, the influence of each feature on the prediction can be read off directly from its shape function. Agarwal et al. [7] proposed learning the shape functions with neural networks rather than splines, yielding Neural Additive Models (NAMs).

Graph Neural Additive Networks extend this framework to graph-structured data. The key idea is to compute a representation for each node i that aggregates contributions from all other nodes j, weighting each contribution by both the transformed feature values and a function of the graph distance:

$$[\mathbf{h}_i]_k = \sum_{j=1}^{N} \frac{1}{\#\text{dist}(j,i)} \cdot \rho\left(\frac{1}{1+\text{dist}(j,i)}\right) \cdot f_k([\mathbf{x}_j]_k), \tag{3}$$

where $\#\text{dist}(j,i)$ is the number of nodes at distance $\text{dist}(j,i)$ from node i, ρ is a learned *distance function*, and f_k is a learned *shape function* for feature k. The final prediction for node i is obtained by summing across features and applying an activation: $\sigma(\sum_k [\mathbf{h}_i]_k)$.

The architecture enforces that each feature k is processed independently by its own shape function. The distance function ρ controls how much weight is given to nodes at each graph-distance level, revealing whether the model relies primarily on local neighbours or on more distant structural information. Because both ρ and the $\{f_k\}$ can be plotted directly, the trained model admits exact global explanations without any post-hoc approximation.

Bechler-Speicher et al. [2] evaluated GNAN on six multi-class node classification benchmarks (including Cora, Citeseer, and PubMed) and several graph-level tasks, finding that GNAN matches or exceeds the accuracy of black-box GNNs despite its lower architectural complexity. The multi-label setting, however, was not considered. This work closes that gap.

3 Methodology

3.1 Multi-Label Adaptation of GNAN

GNAN was originally formulated for binary and multi-class classification, where a softmax activation produces a single distribution over mutually exclusive classes. To adapt it for multi-label classification we make two changes. First, the output activation is replaced by an element-wise *sigmoid*, so that the C label scores are produced independently and a node may be assigned any subset of labels; the model is trained with binary cross-entropy (`BCEWithLogitsLoss`) summed over labels. Second, following the multi-class extension of GNAN [2], the shape functions f_k and the distance function ρ emit a C-dimensional vector rather than a scalar, giving each label its own shape and distance behaviour. Concretely, Eq. 3 is evaluated per label c, and the logit for node i and label c is $\sum_k [\mathbf{h}_i]_{k,c}$, passed through a sigmoid. All other components of the architecture are unchanged, preserving GNAN's exact interpretability. Because Eq. 3 sums over *all* nodes, a naive implementation is infeasible for the larger graph; we use an exact, memory-efficient formulation described in Section 3.5.

3.2 Datasets

We use two real-world datasets from the MLGNC suite [1], chosen to contrast a high- and a low-homophily regime (Table 1). DBLP is a co-authorship graph in which nodes are authors, the four labels are research areas, and features are 300-dimensional TF-IDF word vectors; it has high label homophily. PCG is a protein–phenotype interaction network with 15 labels and low label homophily.

Table 1: Dataset statistics (MLGNC suite [1]). r_{hom} is the multi-label label homophily.

Dataset	Nodes	Edges	Feat.	Labels	r_{hom}	Avg. labels/node
DBLP	28,702	68,335	300	4	0.76	1.2
PCG	3,233	37,351	32	15	0.17	2.5

3.3 Baselines

We compare GNAN against baselines reported in the MLGNC benchmark [1], which use the same datasets, splits, and metric: a feature-only MLP (no graph structure), GCN [4], and GraphSAGE [5]. We also refer to DeepWalk, the strongest non-GNN baseline on PCG. We take these numbers directly from the published benchmark rather than re-running them; this is sound because the datasets, splits, and primary metric are identical, and it avoids the risk of misreporting other authors' methods.

3.4 Evaluation Protocol

We follow the MLGNC protocol [1]: each result is the mean ± standard deviation over the three provided 60/20/20 splits. We do *not* tune hyperparameters; they are fixed to the defaults of the original GNAN work (Appendix A). The validation set is therefore not used for hyperparameter selection but for *model selection over training time*: during training we monitor the validation loss, keep the parameters from the epoch at which it is lowest, and stop once it has not improved for 100 epochs (early stopping). Because this still requires data held out from training, a validation split is retained even though no hyperparameters are tuned; the test set is untouched during both training and this selection and is used only for the reported metrics. Our *primary* metric is the macro-averaged **Average Precision (AP)**, which Zhao et al. adopt as their main metric because it is threshold-free and robust under label sparsity, where AUC-ROC is unreliable [1]; the baseline numbers we compare against are the AP values reported in the MLGNC paper. We report micro-F1 and macro-F1 as secondary metrics, computed at a fixed 0.5 threshold without per-label threshold tuning, so that the decision rule is identical across methods.

3.5 Implementation and Hyperparameters

We keep GNAN's hyperparameters fixed to the defaults of the original work [2], listed in Appendix A. The only implementation concern is scale: because Eq. 3 sums over all nodes weighted by graph distance, a naive implementation would materialise an $N \times N$ distance matrix, which is infeasible for DBLP ($N \approx 28{,}700$). We instead precompute and cache the exact shortest-path hop counts once, and evaluate

the per-node aggregation in batches over the small set of distinct hop values, since ρ depends only on those. This is mathematically identical to the reference implementation. All experiments run on CPU, and the cached distances and adapted code are documented so that the results can be reproduced.

4 Experimental Results

4.1 Performance Against Baselines

Table 2 reports macro-AP for GNAN against the best baseline (GCN) and the graph-agnostic lower bound (MLP) on both datasets. On the high-homophily DBLP dataset, GNAN reaches 0.850 ± 0.002, within roughly four points of GCN (0.893) and far above the feature-only MLP (0.350); it also exceeds GraphSAGE-level methods reported by the benchmark (GraphSAGE 0.868; DeepWalk 0.585) [1]. On the low-homophily PCG dataset, GNAN reaches 0.160 ± 0.009, comparable to the feature-only MLP (0.148) and below the GNN baselines and DeepWalk (GCN 0.210, GraphSAGE 0.185, DeepWalk 0.229) [1]. GNAN therefore comes within five points of the best baseline on the high-homophily dataset, the regime where its additive design is expected to be sufficient. For reference, GNAN's secondary metrics are micro-F1 0.793 / macro-F1 0.769 on DBLP and micro-F1 0.006 / macro-F1 0.004 on PCG; the near-zero PCG F1 is explained in Section 4.2.

Table 2: Average Precision (macro) on the test set, mean over three splits. The MLP and GCN rows are the published MLGNC baseline results [1] (lower bound and best baseline, respectively); the GNAN row is our adapted model and is the contribution of this work. Other baselines are discussed in the text.

Method	DBLP (AP)	PCG (AP)
MLP (baseline)	0.350	0.148
GCN (best baseline)	0.893	0.210
GNAN (ours)	0.850 ± 0.002	0.160 ± 0.009

4.2 Per-Label Learnability: Why Most PCG F1 Scores Are Near Zero

On PCG, micro/macro-F1 at the 0.5 threshold are close to zero. This is a *calibration* artifact, not a failure to learn. PCG labels are *sparse*: on average only about 13% of nodes carry any given label, so a classifier that simply ranks nodes well can still keep almost all of its sigmoid outputs below 0.5, in which case the thresholded prediction is almost always negative and F1 collapses. The threshold-free AP tells a different story, because it measures whether the model *ranks* positive nodes above negative ones regardless of where the threshold sits. We call a label *learnable* when its per-label AP clearly exceeds the prevalence baseline, i.e. the AP a random ranker would obtain, which equals the label's prevalence; "clearly" here means by a margin of more than 0.02 AP. By this measure 10 of the 15 PCG labels are learnable even though most have F1 = 0 at 0.5 (Table 3, right). Per-label AP is strongly correlated with both label prevalence and per-label homophily (Spearman +0.98 and +0.94). In other words, *which* labels are learnable is governed by properties of the data, how common and how homophilic a label is, rather than by a model defect.

On DBLP the picture is the opposite (Table 3, left): all four labels are well learned, with per-label AP between 0.77 and 0.94 and substantial F1 at 0.5. The two more homophilic labels (AI, DB) achieve the highest AP, and per-label AP again correlates positively with prevalence and homophily (Spearman +0.80 and +0.60).

Table 3: Per-label statistics for both datasets. **Prev.** (prevalence) is the fraction of all nodes carrying the label; it also equals the AP a random ranker scores, which is the baseline each label's AP must beat. **Hom.** (per-label homophily) is the probability that a neighbour shares the label, $P(\text{neighbour has } c \mid \text{node has } c)$, estimated over all edges. **AP** is the per-label test-set Average Precision (scikit-learn's `average_precision_score`); **F1** is the per-label score at the fixed 0.5 threshold; both are averaged over the three splits. Left: DBLP (4 labels, all learnable, F1 usable). Right: PCG (15 labels; AP clearly above the prevalence baseline for most labels even where thresholded F1 is 0).

DBLP

Label	Prev.	Hom.	AP	F1
AI	0.437	0.74	0.935	0.859
DB	0.292	0.79	0.908	0.833
DM	0.202	0.56	0.785	0.701
IR	0.249	0.57	0.772	0.683

PCG

Label	Prev.	Hom.	AP	F1
0	0.315	0.41	0.352	0.016
12	0.297	0.40	0.313	0.003
8	0.201	0.28	0.274	0.005
4	0.196	0.35	0.254	0.010
9	0.144	0.32	0.191	0.000
13	0.114	0.21	0.148	0.000
1	0.124	0.19	0.143	0.000
6	0.102	0.22	0.140	0.019
7	0.089	0.13	0.122	0.000
3	0.068	0.12	0.109	0.000
14	0.054	0.12	0.095	0.000
10	0.066	0.16	0.092	0.000
11	0.068	0.13	0.078	0.000
2	0.052	0.09	0.056	0.000
5	0.040	0.07	0.041	0.000

5 Explanation Analysis

A central advantage of GNAN is that its explanations are *exact*: the learned shape functions f_k and distance function ρ *are* the model, so plotting them is a faithful description of its behaviour, not a post-hoc approximation. We use this to ask what GNAN's explanations reveal, both about the strategy the model has learned and about the two datasets, and whether they match our prior assumptions. We state three assumptions and evaluate each against the learned functions (Table 4). For the shape functions and the local node-importance views we show a single representative split; for the distance function we summarise all three splits, since its cross-split behaviour turns out to be informative.

5.1 Shape Functions

Each shape function f_k maps the value of feature k to its additive contribution to a label's score; because GNAN is additive, a node's score for a label is the sum of these per-feature contributions (Eq. 3), so each shape function can be read on its own. Rather than catalogue the geometric shape of every curve, we summarise what the features *do* to the labels, and quantify those effects across the three seeds.

On DBLP the model relies on a small, stable set of features, each with a clear meaning. Influence is concentrated: of the 300 features only about 26 carry appreciable weight (importance at least 10% of the most influential feature), the 30 most influential account for 58% of the total importance, and this set is highly reproducible across seeds (identical top-10 features, Table 5). For 90% of the influential feature–label pairs the contribution moves in a single direction as the feature grows, so each feature has an easily stated effect: more of it means consistently more, or consistently less, evidence for a given label. Most usefully, over half (58%) of the influential features act as *label discriminators*: the same feature is positive evidence for some research areas and negative evidence for others.

Figure 1 shows three examples. Increasing feature F0 raises the DB, DM, and IR scores together while leaving AI essentially unchanged, so a high F0 value is evidence for those three areas but says nothing about AI. Feature F6 does the reverse for one area, pushing DM sharply down as it raises DB. Feature F2 raises DB, AI, and IR while lowering DM, again setting DM apart from the rest. In each case the effect is read directly off the model, as the additive structure of assumption A1 anticipates. The features are anonymous, however: the MLGNC release ships only numeric TF-IDF values with no vocabulary, so we can see *that* a feature separates these labels but cannot name the underlying word (Section 6).

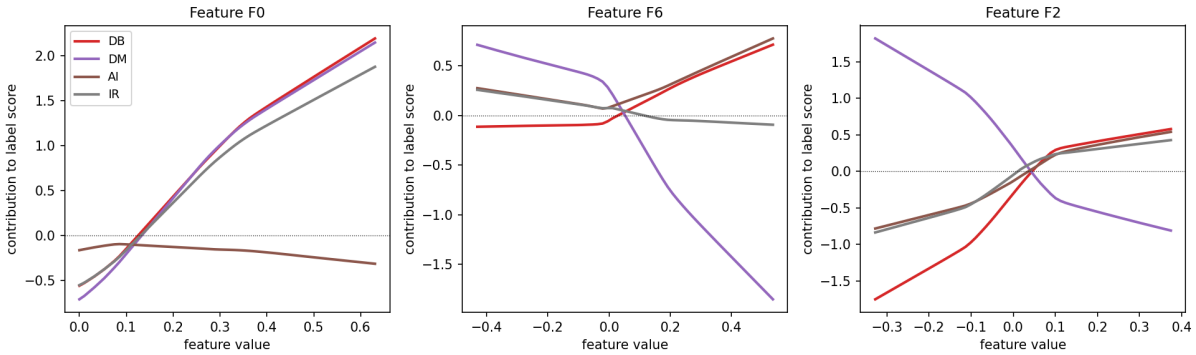

Figure 1: Three DBLP shape functions (seed 1). Each panel is one feature; the x-axis is the feature value, the y-axis is that feature's additive contribution to a label's score, and the four curves are the labels (DB, DM, AI, IR). F0 is evidence for DB/DM/IR but not AI; F6 pushes DM down while raising DB; F2 raises DB/AI/IR and lowers DM.

The shape functions tell a different story on PCG, consistent with its harder, low-homophily regime. Here the model does not isolate a useful few features: all 32 features are influential and importance is spread across them rather than concentrated, and the influential set is unstable across seeds (top-10 feature overlap Jaccard 0.37 versus 1.00 on DBLP, Table 5). The effects are also less clean: only 71% of the feature–label curves move in a single direction, against 90% on DBLP. Figure 2 shows three of the strongest PCG features for three representative labels (best, median, and worst by AP); they still separate labels, but with larger, noisier swings and no compact set of key features to point to. This mirrors at the feature level what the distance function shows at the structural level: on DBLP the model finds a small, stable, interpretable signal, whereas on PCG it spreads its weight thinly and inconsistently.

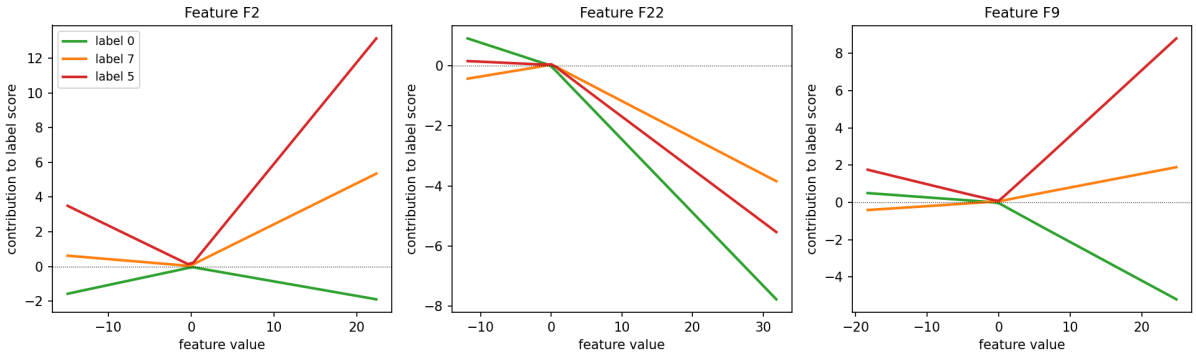

Figure 2: Three of the most influential PCG shape functions (seed 1), shown for three representative labels (best/median/worst by AP: labels 0, 7, 5). Axes are as in Figure 1. The contributions are larger and noisier than on DBLP and rest on no compact set of key features.

5.2 Distance Function

The distance function ρ assigns a weight to every graph distance: in Eq. 3 the contribution of a node j to the prediction at node i is scaled by ρ evaluated at their hop distance $d = \mathrm{dist}(j, i)$. To compare datasets on a common scale we plot the *relative* influence $|\rho(d)|/|\rho(0)|$, i.e. the weight given to a node d hops away divided by the weight given to the node itself ($d = 0$), so every curve starts at 1 and a value of, say, 0.5 at $d = 1$ means a one-hop neighbour counts half as much as the node itself. Figure 3 plots this quantity averaged over a dataset's labels, with the solid line being the mean over the three training seeds and the shaded band ± 1 standard error of that mean *across the seeds*; a narrow band therefore indicates that the learned distance profile is reproducible.

On high-homophily DBLP the curve shows a **steep, reproducible local decay**: relative influence falls to 0.46 at one hop and to near zero by hop six, with 78% of the total neighbourhood influence concentrated within two hops, and the across-seed band is very narrow (standard error ≤ 0.01). GNAN

has learned that when neighbours tend to share labels, only the immediate neighbourhood is worth attending to, and it learns the same profile every time. On low-homophily PCG the curve is **flat and unstable**: relative influence barely decays with distance ($\rho(1)/\rho(0) = 0.79$, only 35% of influence within two hops), and the across-seed band is wide (standard error from 0.12 to 0.28 across hops), so the model does not settle on a consistent distance profile. This instability is itself quantitative: across the three seeds the DBLP relative-distance curves are essentially identical (pairwise Pearson 1.00), whereas the PCG curves are uncorrelated (Pearson −0.26; Table 5), because graph distance carries no reliable label signal on PCG for the model to latch onto. The contrast directly answers the second half of the research question: GNAN's learned distance function *adapts* to the homophily of the data, concentrating locally when neighbours are informative and flattening when they are not.

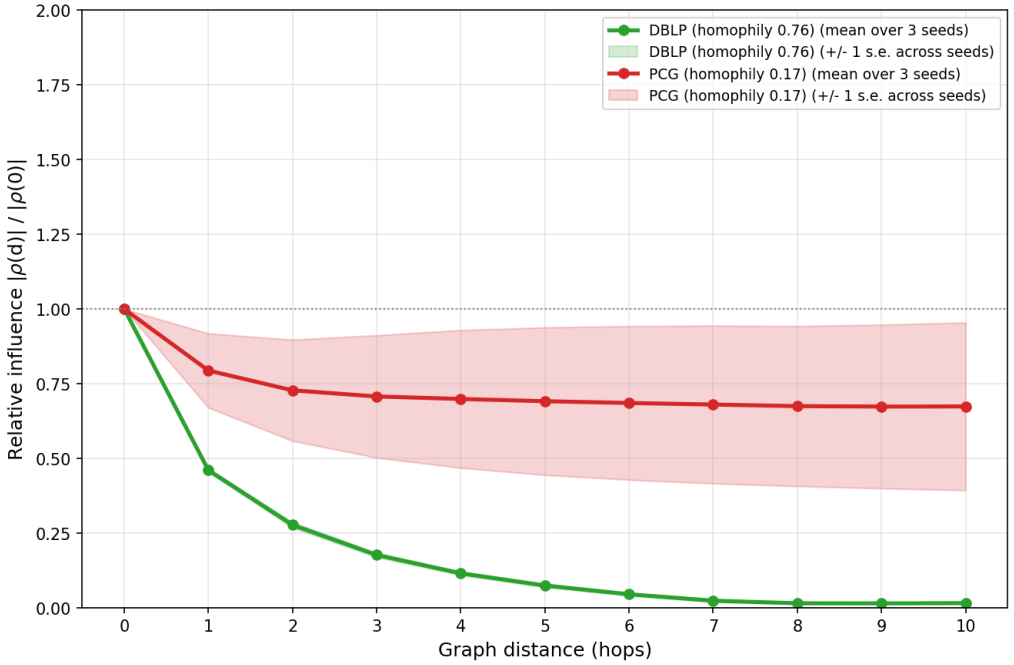

Figure 3: Relative distance influence $|\rho(d)|/|\rho(0)|$ versus hop distance. Each solid curve is the mean over a dataset's labels, averaged over the three training seeds (normalised to 1 at the node itself); the shaded band is ±1 standard error across the three seeds. DBLP shows a steep local decay with a band so narrow it is barely visible (the profile is highly reproducible), whereas PCG is nearly flat with a wide band (the profile is not reproducible across seeds).

5.3 Feature–Distance Interaction

Figure 4 combines the two functions, showing for each label the contribution of each top feature at each hop distance. On DBLP the mass is concentrated at the lowest hops, consistent with the steep ρ decay, and the per-label patterns differ, consistent with the label-specific shape functions.

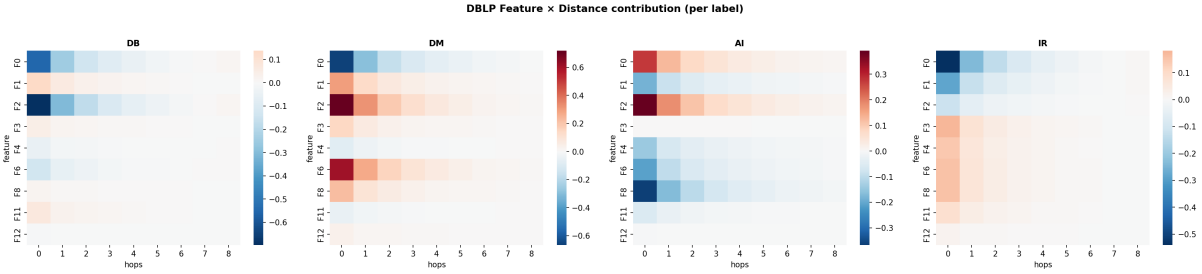

Figure 4: DBLP feature × distance contributions per label. Influence is concentrated at low hops, so the predictive signal comes from a few features in the immediate neighbourhood.

5.4 Local Node Importance

The contribution of each node to a single prediction (Eq. 3) can be visualised as an ego-graph (Figure 5). We deliberately select a correctly predicted node whose neighbourhood is *not* perfectly homogeneous, to show how GNAN behaves when the local signal is imperfect. For the DBLP author shown, the prediction is driven by the node itself and a small number of strongly contributing neighbours, the large majority of which (88%) share the target label; the few different-label neighbours (grey) carry *negative* contributions, i.e. GNAN learns to down-weight them. For the PCG protein, only about half of the influential neighbours share the label and many contribute negatively, reflecting the unreliable neighbourhood signal under low homophily. The local explanation thus reproduces, at the level of a single prediction, the global homophily contrast of Section 5.2.

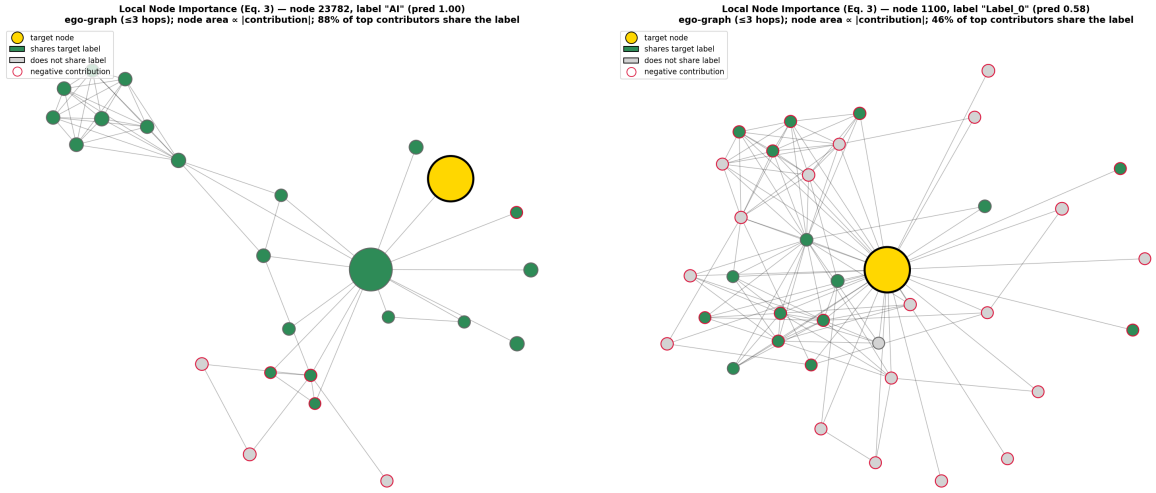

Figure 5: Local node importance as ego-graphs; node area $\propto$ contribution magnitude (Eq. 3), green = shares the target label, grey = does not, red outline = negative contribution, gold = target. Left: DBLP, where same-label neighbours dominate (88% of top contributors here) and different-label ones are down-weighted. Right: PCG, a roughly even mix with many negative contributions. The local explanation reproduces the global homophily contrast.

5.5 Assumptions, Reproducibility, and Domain Reading

Table 4: Prior assumptions and whether the learned explanations agree.

Assumption	Verdict from the learned explanations
A1: shape functions reveal per-label feature importance	Confirmed: features act as label discriminators; sparse and stable on DBLP, spread and unstable on PCG.
A2: ρ reflects neighbourhood reliance; high homophily $\Rightarrow$ local decay	Confirmed: steep reproducible decay on DBLP; flat and unstable on PCG.
A3: additive structure trades expressiveness for interpretability	Confirmed: exact explanations; competitive on DBLP, low-end on PCG.

Reproducibility and stability. GNAN's explanations are reproducible in two senses. First, they are *faithful by construction*: the explanation is the trained model, with no approximation. Second, they are numerically stable on the favourable dataset and *informatively unstable* on the hard one (Table 5). On DBLP the three independently trained seeds yield an *identical* top-10 feature set (Jaccard 1.00) and identical relative distance curves (pairwise Pearson 1.00); on PCG the top-10 sets barely overlap (Jaccard 0.37) and the distance curves are uncorrelated (Pearson -0.26). The instability on PCG is itself

a reproducible, interpretable finding: the model cannot settle on a consistent explanation because the data offers no reliable structure to latch onto.

Validation by ablation. Because GNAN is interpretable by design, the shape functions identify which features the model uses. We verify that these are *genuinely* predictive by an *ablation*: we set the ten most important features to zero and measure the resulting drop in test AP, comparing it against the drop from zeroing ten randomly chosen features (Table 5). On DBLP, ablating the top-10 features drops AP by 0.40, which is 25× the drop from random features (0.016), confirming the explanations point at the truly informative inputs. On PCG the two drops are equal (≈0.003, ratio 1×): no feature is meaningfully more important than any other, consistent with the model's near-baseline performance and unstable explanations.

Table 5: Quantitative checks on the explanations (mean over 3 seeds): cross-seed stability, distance decay, and feature-ablation validation.

Measure	DBLP	PCG
Top-10 feature Jaccard (cross-seed)	1.00	0.37
Feature-importance Spearman (cross-seed)	0.83	0.42
Distance-curve Pearson (cross-seed)	1.00	−0.26
$\rho(1)/\rho(0)$	0.46	0.79
Influence within ≤ 2 hops	78%	35%
AP drop, top-10 features ablated	0.399	0.003
AP drop, 10 random features ablated	0.016	0.003
Top-vs-random AP-drop ratio	25×	1×

Domain reading. The contrast is interpretable from the two domains alone, without specialist expertise. In DBLP, an academic co-authorship graph, co-authors (one hop) tend to work in the same research area, so a steep local ρ and same-label neighbourhoods are exactly what one expects, and GNAN recovers them. In PCG, a biological interaction network, proteins interact across functional modules, so an interaction rarely implies a shared phenotype; the flat, unstable ρ and mixed neighbourhoods reflect that graph distance carries little reliable label signal, which also explains why all graph methods, not only GNAN, struggle on this dataset [1].

6 Discussion

Our results give a consistent answer to the research question. GNAN's additive, feature-independent structure is an *advantage* when the neighbourhood signal is clean: on high-homophily DBLP it is competitive with strong message-passing GNNs while remaining fully interpretable, and its learned distance function correctly identifies that local information suffices. The same structure becomes a *limitation* on low-homophily PCG, where it performs no better than a feature-only baseline; here neighbours are weakly informative and the discriminative signal likely requires feature interactions that GNAN cannot represent. Crucially, the explanations make this failure legible: the flat, unstable distance function shows that the model could not find a useful distance-based aggregation, rather than hiding the difficulty behind an opaque score.

Limitations. (i) *Anonymous features.* The MLGNC DBLP release provides only numeric TF-IDF values and no accompanying word list, so we can identify which feature indices the model relies on but cannot attach a word to each one; the feature-level reading is therefore structural rather than semantic. (ii) *Scope.* We evaluate two real-world datasets. Sub-questions three and four, the effect of homophily and feature quality on synthetic graphs with tunable properties, are not yet addressed and are the primary direction for future work; the synthetic MLGNC generator makes this a natural extension. (iii) *Calibration.* On sparse multi-label data, F1 at a fixed 0.5 threshold understates GNAN's quality; we therefore rely on AP, but per-label threshold calibration would make the thresholded predictions usable. (iv) *Single model variant.* We use one GNAN variant with a per-label distance function, and alternative variants may behave differently. (v) *Baselines from the literature.* We compare against the baseline scores published

with the MLGNC benchmark rather than re-running them; this is sound because we use the identical datasets, splits, and primary metric, although minor protocol differences cannot be entirely excluded.

7 Responsible Research

Reproducibility. All experiments use the publicly available MLGNC datasets [1] with the provided 60/20/20 splits, fixed random seeds, and a documented hyperparameter configuration (Appendix A). Shortest-path distances are deterministic and cached, and the adapted GNAN code is documented so that every number and figure can be regenerated. For completeness, all code was run with Python 3.12, PyTorch 2.11 (CPU build), PyTorch Geometric 2.7.0, and scikit-learn.

Honest reporting. We report Average Precision as the primary metric because it is threshold-free and the metric used by the benchmark we compare against; comparing a tuned threshold for our model against fixed-threshold baselines would be misleading, so we deliberately avoid per-label threshold tuning for the headline comparison and disclose the secondary F1 numbers and their calibration caveat explicitly.

Interpretability is not ground truth. GNAN's explanations are faithful to the model by construction, but faithfulness is not the same as correctness about the underlying domain. A shape or distance function reflects what the model learned from a particular dataset, which may encode dataset biases or spurious correlations; it should not be treated as established scientific knowledge, especially in high-stakes domains such as medicine. Interpretability by design makes the model's reasoning inspectable, which is a precondition for such scrutiny, but does not by itself guarantee that the reasoning is right.

Data and ethics. The datasets are standard, anonymised academic and biological benchmarks and contain no personal data; no human-subject experiments were conducted. Experiments run on a single CPU machine, so the compute footprint is modest.

Use of AI assistance. Generative AI tools were used to assist with code scaffolding, debugging, and editing; all experimental design, results, and their interpretation were verified by the author.

8 Conclusions and Future Work

We presented the first adaptation of Graph Neural Additive Networks to multi-label node classification and evaluated it on a high- and a low-homophily real-world dataset. The adaptation, a sigmoid output with per-label shape and distance functions, is stable and produces non-trivial predictions. GNAN is competitive with strong GNNs on high-homophily DBLP (macro-AP 0.850, within $\sim$4 points of GCN) but falls to the level of a feature-only baseline on low-homophily PCG. Most importantly, GNAN's exact, built-in explanations reveal *why*: its learned distance function adapts to label homophily, concentrating weight locally and reproducibly where neighbours are informative and flattening, without a stable signal, where they are not; and its local node-importance ego-graphs reproduce this contrast at the level of individual predictions. The additive structure is thus an asset under high homophily and a liability under low homophily, and the interpretability by design makes that trade-off transparent.

Future work will (i) run the synthetic homophily and feature-quality sweeps (sub-questions three and four) using the MLGNC generator to confirm the trend across a controlled homophily range; (ii) further verify the explanations, beyond the faithfulness, cross-seed stability, and ablation checks reported here, by comparing them against post-hoc explainers and, where domain knowledge is available, expert judgement, and by attaching word-level semantics to the DBLP shape functions if the original TF-IDF vocabulary can be recovered; (iii) explore per-label threshold calibration for usable sparse-label predictions; and (iv) extend the study to additional datasets and GNAN variants.

References

[1] T. Zhao, N. T. Dong, A. Hanjalic, and M. Khosla, "Multi-label node classification on graph-structured data," *Transactions on Machine Learning Research*, 2023. [Online]. Available: https://openreview.net/forum?id=EZhkV2BjDP

[2] M. Bechler-Speicher, A. Globerson, and R. Gilad-Bachrach, "The intelligible and effective Graph Neural Additive Network," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/a4c3a66ed818455b8bbe591b6a5d0f56-Abstract-Conference.html

[3] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[5] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[6] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*. Chapman & Hall/CRC, 1990.

[7] R. Agarwal, L. Melnick, N. Frosst, X. Zhang, B. Lengerich, R. Caruana, and G. Hinton, "Neural additive models: Interpretable machine learning with neural nets," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

A Hyperparameters

Table 6 lists the GNAN training configuration, identical for both datasets and fixed to the defaults of the original work [2]. We do not tune these values; the validation split is used only for early stopping.

Table 6: GNAN training configuration (identical for both datasets).

Shape/distance layers	3
Hidden width	64
Dropout	0.0
Optimiser	Adam
Learning rate	1×10^{-3}
Weight decay	5×10^{-5}
Gradient clipping (norm)	1.0
Loss	BCEWithLogitsLoss
Max epochs / early-stop patience	1000 / 100
Splits (seeds)	3 (60/20/20)
F1 decision threshold	0.5 (fixed)