Per-Node k-Hop Label Homophily Predicts GNN Accuracy in Multi-Label Node Classification

Veaceslav Guzun[1]

Supervisor(s): Megha Khosla, Elena Congeduti

[1]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: Veaceslav Guzun
Final project course: CSE3000 Research Project
Thesis committee: Megha Khosla, Elena Congeduti, Christoph Lofi

Abstract

Multi-label node classification asks a graph neural network to assign each node a set of labels. For single-label classification the success of these networks is usually attributed to label homophily, the tendency of connected nodes to share labels; for the multi-label case Zhao et al. introduced a homophily metric and reported that it tracks performance across datasets. That metric measures label similarity only between directly connected nodes, even though graph neural networks aggregate information from wider neighbourhoods. It is therefore unknown whether label similarity at larger distances also predicts how accurately a network classifies a node, and at which distance that signal is strongest. We generalise the metric to a per-node label-similarity score at any chosen distance. We then correlate this score node-by-node with trained-model accuracy on three benchmark datasets, repeat the analysis across model depths to separate a data-driven explanation from an architectural one, and test the relationship causally with two synthetic graph generators that plant label structure at controlled distances. The predictive signal is consistently strongest at distance two, not at directly adjacent nodes, across two standard architectures and up to several thousand test nodes per dataset; the most predictive distance does not shift with model depth, so it is a property of the graphs rather than of the model; and controlled synthetic experiments confirm a causal effect of two-hop homophily on accuracy in graphs built to isolate it, with weaker and partially confounded support at three hops. Direct measurements further show that measuring similarity at an exact distance, rather than over a cumulative neighbourhood, is the right unit for identifying that scale. All code, data splits, and figures are released for reproducibility.

1. Introduction

Many interesting real-world problems involve graphs whose nodes carry multiple labels at once. A user on a social platform likes sports, films, and politics at the same time; a protein takes part in metabolism, transport, and signalling simultaneously; a research paper is filed under both databases and machine learning at once. The task of filling in the missing label sets, given a graph and the labels of some of its nodes, is **multi-label node classification**, and the dominant tools for it are graph neural networks (GNNs), in particular the Graph Convolutional Network (GCN) [2] and the Graph Attention Network (GAT) [3].

GNNs work by aggregating information from each node's neighbours. We call a node's directly connected neighbours its one-hop neighbours, the neighbours of those neighbours its two-hop neighbours, and so on; each hop is one step along an edge, and the k-hop neighbourhood of a node is everything reachable in k such steps. A two-layer model folds in the one-hop neighbours directly and the two-hop neighbours indirectly, through a second round of message passing. For this to help rather than hurt, the neighbours have to carry information that is actually predictive of the target node's labels. The property that captures this intuition is **homophily**: the tendency of connected nodes to be alike in terms of labels [5]. When homophily is high, aggregation is a sensible assumption; when it is low, the same aggregation averages in noise, which is the textbook explanation for why GNNs struggle on heterophilic graphs [6, 7].

For the multi-label setting, Zhao et al. [1] built the first careful benchmark and proposed a clean homophily metric (their Definition 1): for every edge, compute the Jaccard similarity of the two incident label sets, and average over all edges. They showed empirically that this number tracks GNN performance, with high-homophily datasets handled well and low-homophily datasets handled poorly. It is a useful and influential diagnostic, and the foundation this paper builds on. We adopt their benchmark datasets, their GCN/GAT architectures and hyperparameters, and their Definition 1 unchanged at one hop, then ask the question their dataset-level, one-hop analysis was not positioned to answer.

That question is left open because the metric is, by construction, a one-hop quantity. A two-layer

GCN has an effective receptive field two hops wide, and practitioners routinely use deeper models that reach three hops or more. There is therefore a mismatch between what the metric records, the similarity of directly adjacent nodes, and the wider neighbourhood that the model actually consumes. An edge-level summary may overlook signal, or interference, arriving from two or more hops away. This is the gap our work sets out to investigate.

> **Research question.** How does the degree of label similarity between a node and its neighbourhoods at increasing distances (one hop versus higher-order: two hops, three hops, and beyond) affect the performance of graph neural networks on multi-label node classification, and at which scale is that similarity most predictive of per-node accuracy?

We pursue this through four guiding subquestions, each taken up in a later section and together tracing the path from observation to cause:

- **SQ1.** Does label similarity beyond the first hop predict per-node accuracy, and at which distance is it strongest?
- **SQ2.** Should similarity be read off an exact-distance shell or a cumulative neighbourhood?
- **SQ3.** Is the most predictive scale a property of the graph or an artefact of the model's depth?
- **SQ4.** Does higher-order homophily cause the accuracy change, or merely co-vary with it?

This work is therefore a direct, higher-order, per-node extension of Zhao et al., with four contributions. **(i)** We generalise their one-hop edge-level metric (Definition 1) to an exact-k-hop, per-node label-similarity score that reduces to the original at $k = 1$ (Section 3.1). **(ii)** On three benchmarks the correlation between k-hop homophily and per-node accuracy is strongest at $k = 2$, not $k = 1$, for both architectures (Section 4.2). **(iii)** That most-predictive distance does not track model depth, isolating it as a property of the data rather than the architecture (Section 4.4). **(iv)** A controlled synthetic experiment establishes a causal effect of two-hop homophily on accuracy in graphs built to isolate it; a second experiment gives weaker, less cleanly isolated evidence of the same at three hops (Section 4.5). We also show, with direct measurements, that the exact shell rather than the cumulative ball is the right unit for locating that scale (Section 4.3).

2. Background

This section reviews the property the study is about, label homophily in the multi-label setting (Section 2.1), fixes the notation used throughout (Section 2.2), and introduces the graph neural networks that act on it (Section 2.3).

2.1 Homophily and the multi-label setting

Homophily, the tendency of connected nodes to share labels, is the assumption underlying neighbourhood aggregation: high homophily injects useful signal, low homophily averages in noise. Pei et al. [6] showed standard GNNs underperform on heterophilic graphs and proposed a geometric aggregation scheme, while Zhu et al. [7] formalised the edge-homophily ratio and catalogued the design choices (ego-neighbour separation, higher-order neighbourhoods, intermediate-layer combination) that help GNNs cope with heterophily. Both are single-label, where similarity is binary; our multi-label setting needs a graded, set-based measure.

Zhao et al. are our central reference: they assembled the benchmark and defined the Jaccard edge-homophily metric we generalise, but their analysis is at the dataset level and stops at one hop. We depart from them in two ways that matter, a per-node and a per-k formulation: the first defeats a sample-size problem (Section 3.2), the second is the research question itself. Everything else (data,

models, training, the $k = 1$ metric) is held at their published settings, so the comparison isolates the higher-order, per-node contribution.

Related work. Prior work touches our question in two ways. Some of it measures homophily: single-label scores like the edge-homophily ratio of Zhu et al. [7] give one number per graph and only check whether neighbours share a class, and Zhao et al. [1] extend this to label sets but still measure it edge by edge at one hop. Other work uses distant neighbours inside the model: H2GCN [7] adds the two-hop neighbours, Geom-GCN [6] reaches similar but distant nodes through a latent space, and multi-scale models combine several hop ranges to steady a noisy one-hop signal [13]. These models assume distant neighbours help and are built to use them, but none checks, for each node, the distance where that help is actually largest. We measure homophily at each exact distance and compare it, node by node, with how well the model does, which turns that assumption into something we can test.

2.2 Notation and setting

Graph, labels, and distance. Let $G = (V, E)$ be an undirected graph with $n = |V|$ nodes. Each node $v \in V$ carries a binary label vector in $\{0, 1\}^L$ over L possible labels, and we write $\ell(v) \subseteq \{1, \dots, L\}$ for the set of indices where this vector is one (the label set of v). The shortest-path distance between two nodes is $d(u, v)$. **Multi-label node classification** is the task of predicting $\ell(v)$ for unlabelled nodes given the graph and the labels of a labelled subset. When we report per-node predictions from a trained model we write $\hat{p}_v \in [0, 1]^L$ for its sigmoid output (the predicted probability per label) and $y_v \in \{0, 1\}^L$ for the true label vector.

Neighbourhoods: shells and balls. The direct (one-hop) neighbourhood of v is $\mathcal{N}(v) = \{u \in V : (u, v) \in E\}$, the nodes connected to v by an edge. To talk about neighbourhoods at greater distances we use two different generalisations, both of which appear in this paper (Figure 1), and the distinction between them matters for the results (Section 4.3).

The **exact-k-hop shell** of v,

$$\mathcal{N}_k(v) \;=\; \{\, u \in V : d(u, v) = k \,\},$$

collects exactly the nodes whose shortest-path distance from v is k. In particular $\mathcal{N}_1(v) = \mathcal{N}(v)$, $\mathcal{N}_2(v)$ contains the second-degree neighbours but not the first-degree ones, and so on. Shells at different k are pairwise disjoint, so $\mathcal{N}_1(v), \mathcal{N}_2(v), \mathcal{N}_3(v), \dots$ slice the graph around v into concentric rings (the "shells" of the name), each one reporting only on the structure at its own distance.

The **cumulative ball** of radius k,

$$B_k(v) \;=\; \{\, u \in V : 1 \le d(u, v) \le k \,\} \;=\; \mathcal{N}_1(v) \cup \mathcal{N}_2(v) \cup \cdots \cup \mathcal{N}_k(v),$$

collects every node within k hops of v, pooling all the shells from one up to k together. The balls are nested: $B_1(v) \subseteq B_2(v) \subseteq B_3(v) \subseteq \dots$, so widening k never removes a node and a statistic computed on $B_k(v)$ mixes information from every distance up to k. The shell is the finer of the two representations: each ball can be recovered as a union of shells, but a shell cannot be recovered from a ball. We will use shells when we need to localise a signal to a particular distance and balls when we want to ask what changes as the radius grows. **Concretely, every experiment in this paper measures homophily on the exact-k-hop shell**; the single exception is the unit comparison of Section 4.3, which additionally computes the cumulative ball, precisely to show why the shell is the right choice. The observational correlation (Section 4.2), the depth sweep (Section 4.4), and the synthetic causal sweeps (Section 4.5) all use shells.

Label similarity. We measure similarity between two label sets with the **Jaccard similarity** [16], $\text{Jaccard}(\ell(u), \ell(v)) = |\ell(u) \cap \ell(v)| / |\ell(u) \cup \ell(v)|$, a value in $[0, 1]$ that is 1 when the two sets are identical

and 0 when they are disjoint.

Homophily metrics. From Jaccard we define the **per-node k-hop homophily** as the mean Jaccard between v and the labels of its k-hop shell,

$$s_k(v) \;=\; \frac{1}{|\mathcal{N}_k(v)|} \sum_{u \in \mathcal{N}_k(v)} \text{Jaccard}(\ell(v), \ell(u)),$$

and the **per-node cumulative homophily** $s_{\leq k}(v)$ analogously as the mean Jaccard over the ball $B_k(v)$. Averaging s_k over nodes, weighted by shell size, gives the **dataset-level k-hop homophily** H_k, a single number summarising label similarity at distance k over the whole graph. At $k = 1$, H_1 coincides with the edge-homophily metric of Zhao et al., which we will denote h when comparing against their reference value (their definition averages the Jaccard similarity over every edge).

This per-node, exact-k score generalises Zhao et al.'s edge-homophily along two axes at once. It is resolved per node, so $s_k(v)$ can be correlated with one node's own prediction quality instead of a single graph-level average; and per distance, so it measures label agreement at any chosen scale rather than only between directly linked nodes. At $k = 1$ it collapses back to the edge metric; for $k > 1$ it quantifies the label context a depth-k GNN aggregates over, which is what makes it the right object to correlate with per-node accuracy and lets us ask at which distance that context is most predictive. Section 3.1 motivates the design choices behind it (Jaccard, the mean over the shell, and the pair-weighting of H_k).

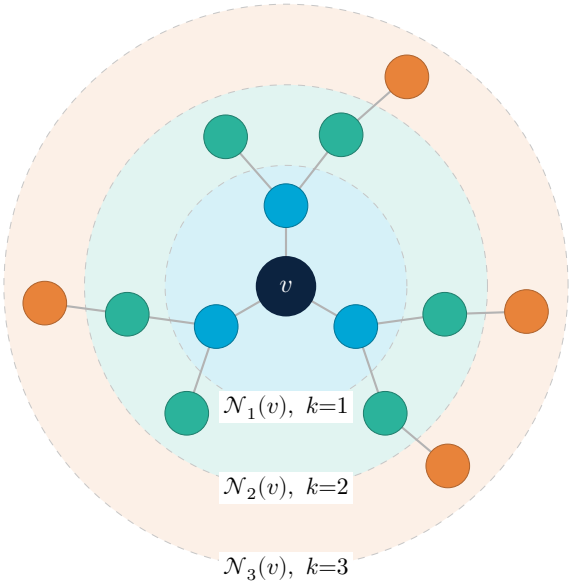

Figure 1: The exact-k-hop shells of a node v. Each node is coloured by its shortest-path distance k from v: the one-hop shell $\mathcal{N}_1(v)$ (the direct neighbours), the two-hop shell $\mathcal{N}_2(v)$, and the three-hop shell $\mathcal{N}_3(v)$. The shells are disjoint concentric rings, and the cumulative ball is their union up to a given radius (for instance $B_2(v) = \mathcal{N}_1(v) \cup \mathcal{N}_2(v)$). The per-node homophily $s_k(v)$ averages label similarity between v and the nodes of a single shell.

2.3 Graph neural networks for node classification

A graph neural network (GNN) classifies each node by repeatedly updating its representation as a function of its neighbours'. A typical layer first aggregates the representations of a node's direct (one-hop) neighbours and then combines that aggregate with the node's own representation; stacking L such

layers folds in information from up to L hops away, so the model's receptive field grows linearly with depth. GCN uses a fixed, degree-normalised average over the one-hop neighbours, so the aggregation weights depend on graph structure alone. GAT instead learns attention weights from the data, so the aggregation can emphasise some neighbours over others. Multi-label classification trains a sigmoid output per label with binary cross-entropy, so each of the L labels is decoded independently.

Using neighbours beyond one hop is a recurring idea, from higher-order aggregation in heterophily-robust models to simply stacking more layers. Stacking has a well-documented cost: Li et al. [8] showed repeated graph convolution drives node representations toward indistinguishability ("oversmoothing"), and Oono and Suzuki [9] proved this loss of expressive power grows exponentially with depth. This tension is directly relevant to our depth experiment (Section 4.4): adding layers widens the receptive field but can also blur it, as we observe for deep GAT on the smallest dataset.

3. Methodology

Section 3.1 fixes the measurement, generalising the one-hop edge metric to the per-node, exact-k-hop score $s_k(v)$, and Section 3.2 explains why we analyse it node by node rather than dataset by dataset. We then run one experiment per subquestion: an observational correlation of $s_k(v)$ with prediction quality, asking at which distance it is strongest (SQ1, Section 4.2); the same analysis with the cumulative ball in place of the shell (SQ2, Section 4.3); a depth sweep over $L \in \{1, 2, 3, 4\}$, testing whether the predictive scale follows the receptive field or stays fixed with the data (SQ3, Section 4.4); and controlled synthetic sweeps that vary one shell's homophily in isolation (SQ4, Section 4.5). Sections 3.3 to 3.5 describe the shared models, datasets, scoring, and generators.

3.1 From edges to exact-k-hop shells

The symbols are in Section 2.2; here we give the reasoning. We use Jaccard because label sets vary in size, so a raw intersection count is not comparable across pairs; it normalises by the union, stays in $[0, 1]$, and is the similarity Zhao et al. use, so our $k > 1$ values sit on the same scale as the established one-hop number. The per-node score is the mean Jaccard over the shell, mirroring the mean aggregation a GCN or GAT performs, and the dataset-level H_k is pair-weighted (each node weighted by its shell size), so that H_k is exactly the mean Jaccard over all node pairs at distance k; this makes H_1 equal Zhao's h identically and keeps high- and low-degree nodes on an equal per-pair footing rather than letting sparse nodes swing the statistic. The metric is parameter-free, so the profile of H_k over k is a reproducible property of the graph.

We measure on the exact-k-hop shell rather than the cumulative ball: shells at different k are disjoint, so each reports a value that depends only on the structure at that distance, whereas the ball pools every distance up to k and its movement with k can be driven by any of them (Section 4.3 shows this matters in practice). Self-loops are stripped before the breadth-first search, since a self-loop would place v in its own one-hop shell at Jaccard 1.

3.2 Per-node unit of analysis

The intuitive test of "does H_k predict performance?" plots one $(H_k, \text{accuracy})$ point per dataset and correlates. This collapses immediately: the benchmark has four datasets and only three overlap with the published GCN/GAT scores, so $n = 3$. A Spearman correlation over three points can take only a handful of values, and if the dataset ranking is preserved across k (which it is here) the correlation is identical at every k, making the experiment structurally blind to the research question.

We therefore drop a level of aggregation and let every **test node** be its own data point: the pair $(s_k(v), q(v))$, where $q(v)$ is the trained model's per-node prediction quality. This turns three points into one to five thousand points per dataset, and the correlation can vary freely with k. Holding the dataset, model, and training run fixed and varying only the node also removes the cross-dataset confounds (size, label cardinality, feature dimension) that a per-dataset analysis cannot.

3.3 Models, training, and datasets

We test on GCN and GAT for four reasons. They are the exact architectures Zhao et al. benchmarked, so reusing their configuration isolates our per-node, per-k contribution and reproduces their one-hop homophily as a pipeline check (Section 4.1). They bracket the two dominant aggregation rules, GCN's fixed degree-normalised average and GAT's learned attention, so a finding that holds for both is unlikely to be an artefact of one. Both are mean-style aggregators matching the mean-Jaccard score, and each layer aggregates exactly one hop, so depth L gives a clean L-hop receptive field that the depth experiment relies on (Section 4.4). Finally, they are well-characterised, so deep GAT's seed-instability on the smallest dataset reads as the textbook oversmoothing signature rather than an opaque quirk.

We use Zhao's hyperparameters unchanged: a two-layer GCN (hidden size 256, ReLU, dropout 0.5, sigmoid output) and a two-layer GAT (first layer 8 heads of 8 dimensions, second layer one head to the label dimension, ELU, dropout 0.5), trained with Adam (learning rate 0.01, weight decay 5×10^{-4}), binary cross-entropy loss, and early stopping on validation loss (patience 100). For the depth experiment we additionally train $L \in \{1, 3, 4\}$ variants, and every (dataset, model, depth) configuration is trained under three seeds (42, 123, 456), with each correlation computed per seed and reported as mean and standard deviation.

Three datasets carry the per-node study; Table 1 additionally reports Yelp, which we use only for the dataset-level homophily profile of Section 4.1, since it is too large for exact breadth-first search beyond one hop. HumanGo and DBLP are single graphs, where the model runs on the whole graph with train/validation/test node masks; PPI comes as separate graphs (20 for training, 2 for validation, 2 for testing) of the protein-protein interaction benchmark [12], using the standard inductive split [4] loaded through PyTorch Geometric [10]. As a faithfulness check we adopt Zhao's exact architectures and hyperparameters and re-tune nothing, and our one-hop homophily matches their published H_1 on DBLP to two decimals (0.755 against their 0.76) and, once the self-loop policy is reconciled, on Yelp as well (Section 4.1), confirming that we load and process the same graphs under the same model configuration.

3.4 Scoring a node's prediction

Given the model's sigmoid output $\hat{p}_v \in [0, 1]^L$ and true labels $y_v \in \{0, 1\}^L$, we score each node's prediction with **per-node Average Precision (AP)**, which treats the L labels as a ranking task and computes `average_precision_score` [11]; it is threshold-free, and we exclude only the small fraction of nodes whose labels are all-zero (where AP is undefined) or all-one (where it is trivially 1). Where we speak of "per-node accuracy" we mean this AP score (and test Micro-F1 for the synthetic graphs of Section 4.5); we use "accuracy" as an umbrella for prediction quality, since no node-level classification-accuracy metric is reported. We correlate using Spearman's r rather than Pearson because the hypothesis is monotonic, not linear, because AP saturates near 1 (a ceiling that shrinks Pearson but not Spearman), and because $s_k(v)$ is heavily right-skewed (so rank-based statistics resist outlier hubs).

3.5 Synthetic generators for causal tests

Observational correlations cannot rule out a hidden confounder (for instance, features that both help the GNN and coincide with homophily). To establish causation we generate graphs in which we vary only the homophily of one shell and hold the rest at baseline, using Gaussian-noise node features so the only useful signal is topology. This makes each sweep a controlled intervention rather than an observation: we set the targeted shell's homophily H_k to a chosen value while pinning H_1 at baseline and the features at pure noise, so H_k is the only quantity that changes from one sweep point to the next. Any monotone change in test accuracy along the sweep is therefore attributable to H_k itself rather than to a confound, which is precisely the inference an observational correlation cannot license. The two generators are given as Algorithms 1 and 2 in Appendix B. A hub-spoke generator (disjoint stars) places controllable label structure at exactly two hops: spokes of a common hub are distance-2 pairs, so tuning their shared-prototype purity moves H_2 from baseline to nearly 1 while H_1 stays flat. A hexagon-cycle generator (disjoint 6-cycles with three shared prototypes) places it at exactly three hops, since opposite nodes of a hexagon are distance-3 pairs. Each sweep runs five target homophily values, three seeds, and all four depths.

4. Results

We first report the datasets' homophily profiles and check the metric against the reference values (Section 4.1), then take the four subquestions in turn: which scale is most predictive (Section 4.2, SQ1), whether the exact shell or the cumulative ball is the right unit (Section 4.3, SQ2), whether that scale is set by the data or the model's depth (Section 4.4, SQ3), and whether the relationship is causal (Section 4.5, SQ4).

4.1 Dataset homophily profiles

Table 1 reports H_k for $k = 1, 2, 3$ and verifies the metric against Zhao et al. Our DBLP one-hop value matches almost exactly ($H_1 = 0.755$ vs their $h = 0.76$); Yelp appears lower only because we strip self-loops, and adding the per-node self-loop contributions back recovers Zhao's published 0.22, so our H_1 is identical to Zhao's h up to that policy choice. All four datasets decay monotonically with distance, but the rate is independent of the starting level: HumanGo decays fastest ($H_3/H_1 = 0.49$), DBLP and PPI sit in the middle (0.66 and 0.63), Yelp slowest (0.77), so a profile has two distinct features (level and shape) rather than one. Yelp is too large for exact breadth-first search at $k > 1$, so its H_2 and H_3 are estimated from a uniform sample of 10,000 source nodes (a consistent estimator of the pair-weighted mean); all other values are exact.

Table 1: Dataset-level k-hop homophily H_k and verification against the reference one-hop values.

Dataset	H_1	H_2	H_3	Ref. H_1 [1]
Yelp	0.182	0.181	0.140	0.22
PPI	0.346	0.269	0.217	n/a
DBLP	0.755	0.576	0.501	0.76
HumanGo	0.398	0.277	0.195	n/a

4.2 The headline result: the predictive scale is two hops (SQ1)

For every test node we pair its k-hop homophily $s_k(v)$ with its per-node AP and compute the within-dataset Spearman correlation. Table 2 and Figure 2 report the result for the two-layer models, averaged

over three seeds, across $k = 1$ to 4. In all six (dataset, model) cells the correlation is strictly larger at a higher level than at $k = 1$, and $k = 1$ is never the maximum. In five of six cells the correlation is **strongest at** $k = 2$, the receptive field of a two-layer GNN; in the sixth (HumanGo GAT) it is essentially tied across $k \in \{2, 3\}$. The most distant level we measured, $k = 4$, is never the strongest and on PPI is the weakest scale of all, so the gain concentrates at the lower of the higher-order shells rather than growing without bound. Standard deviations at $k = 2$ are at most ± 0.02, an order of magnitude below the higher-order-versus-$k = 1$ gap (which ranges from $+0.13$ to $+0.26$ at $k = 2$), so the effect is not a seed artefact; the only cell with wider bands is HumanGo GAT (up to ± 0.06 at $k = 4$), the small graph whose deep runs we flag as oversmoothing-unstable in Section 4.4.

Table 2: Within-dataset Spearman r between per-node $s_k(v)$ and per-node AP (two-layer GNNs, mean over seeds 42/123/456). Bold marks the strongest correlation across the levels studied; for HumanGo GAT the $k = 2$ and $k = 3$ means tie at $+0.82$ (they differ by 0.002, within the seed spread, and the per-seed arg-max splits between the two), so both are bolded. The number of scored test nodes at $k = 1$ is 1029 for HumanGo, 5411 for DBLP, and 5298 for PPI, and falls at higher k as nodes with an empty outer shell are dropped, modestly for PPI and HumanGo and by about a quarter for DBLP (DBLP 5411 $\rightarrow$ 4020 by $k = 4$, against PPI 5298 $\rightarrow$ 5295 and HumanGo 1029 $\rightarrow$ 1025).

Dataset	Model	$k = 1$	$k = 2$	$k = 3$	$k = 4$
HumanGo	GCN	$+0.58$	$+\mathbf{0.71}$	$+0.62$	$+0.40$
HumanGo	GAT	$+0.62$	$+\mathbf{0.82}$	$+\mathbf{0.82}$	$+0.59$
DBLP	GCN	$+0.20$	$+\mathbf{0.41}$	$+0.36$	$+0.30$
DBLP	GAT	$+0.20$	$+\mathbf{0.46}$	$+0.40$	$+0.33$
PPI	GCN	$+0.69$	$+\mathbf{0.87}$	$+0.77$	$+0.43$
PPI	GAT	$+0.67$	$+\mathbf{0.86}$	$+0.79$	$+0.49$

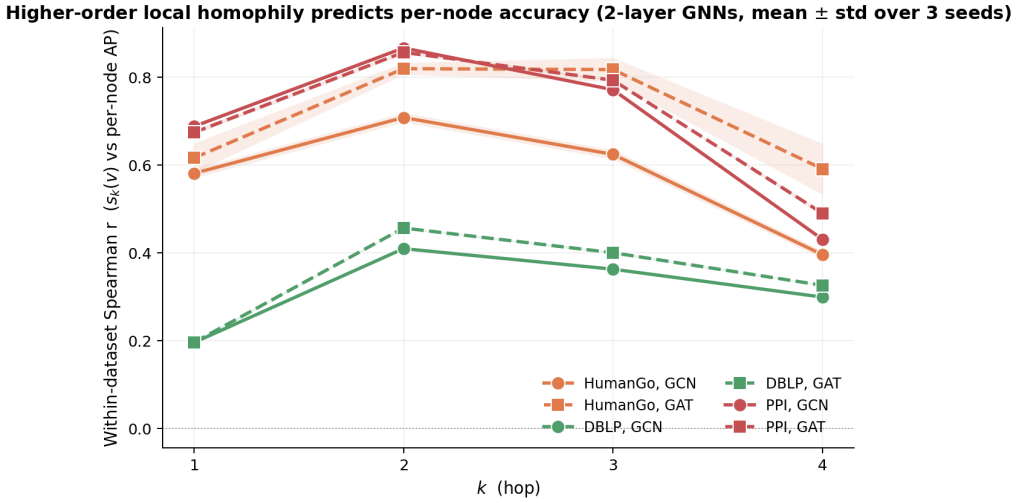

Figure 2: Within-dataset Spearman r between per-node $s_k(v)$ and per-node AP for every (dataset, model) pair. Lines are the seed mean, shaded bands are one standard deviation (mostly thinner than the line). Every line is strongest at $k = 2$ or $k = 3$, never at $k = 1$.

The effect is robust across architectures (GCN and GAT trace nearly parallel curves), so it is a property of the homophily metric relative to two-layer GNNs rather than a quirk of attention. DBLP shows the lowest absolute correlations because most DBLP nodes are predicted near-perfectly, leaving little variance to explain, yet it shows the largest relative jump, more than doubling from $k = 1$ to $k = 2$.

We read the concave shape (rising from $k = 1$, highest at $k = 2$, decaying outward) as the resolution of two competing pressures. The first is coverage. One-hop shells are often tiny: the median node has between 3 and 16 direct neighbours, and a large share (34% on HumanGo, 44% on DBLP) have at most two, so one-hop homophily rests on a handful of Jaccard values that a single unusual edge can dominate. The two-hop shell is substantially larger (Figure 3; larger than the one-hop shell for 68% to 99% of nodes), so it averages over many more neighbours and gives a steadier estimate of the label context, which is why $k = 2$ predicts better than the one-hop signal Zhao et al. measure. The extra coverage costs little in relevance, since homophily falls only to about 0.70 to 0.78 of its one-hop value from one hop to two; multi-scale aggregation has likewise been used to stabilise a noisy single-hop signal in the single-label setting [13].

The second pressure is relevance decay: label correlation falls with distance (the monotone H_k decay of Section 4.1), and the outer shells are also far larger, so their mean Jaccard regresses toward the graph-wide label frequency and carries little information specific to the node. This echoes two threads in the wider literature: low higher-order homophily has been shown to throttle the class-specific signal that reaches a node, through under-reaching and over-squashing [14], and the same distant signal is independently attenuated by the over-smoothing that deep aggregation induces, compounded by the over-squashing it also produces [15].

Beyond two hops the second pressure wins: the metric approaches a global constant and its correlation with accuracy fades, sharply on PPI. Two hops is where the two pressures balance, far enough to escape single-edge noise, near enough that the labels are still about this node. Whether that balance is set by the data or by the two-layer receptive field is what Section 4.4 tests.

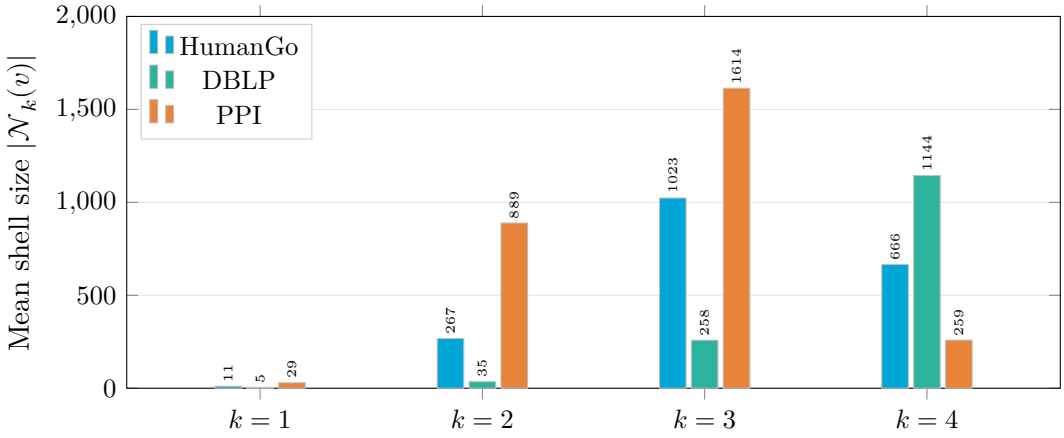

Figure 3: Mean exact-shell size $|\mathcal{N}_k(v)|$ as a function of distance k, per dataset (averaged over test nodes; bar labels give the mean count). One-hop shells are small (means of 4, 11, and 29 nodes for DBLP, HumanGo, and PPI), whereas two-hop shells are roughly an order of magnitude larger, so two-hop homophily averages over many more neighbours and gives a lower-variance estimate of a node's label context.

4.3 Shell versus cumulative: why the unit of measurement matters (SQ2)

Section 3.1 argued for exact shells on principle; here we show it with data. We recomputed every correlation using the cumulative ball $s_{\leq k}(v)$ (the size-weighted mean of shells $1 \dots k$) and compared it against the shell $s_k(v)$ (two-layer GCN, seed-averaged over seeds 42/123/456). Figure 4 gives the result. Raw predictive strength does not separate the two cleanly: the shell is the stronger predictor on DBLP, whereas on HumanGo and PPI the cumulative ball actually reaches higher correlations at the larger radii. The more useful difference lies elsewhere: the cumulative readout and the shell can point

to different predictive scales, and the direction of that difference depends on the graph. On PPI the cumulative correlation climbs monotonically to a maximum at $k = 4$ (+0.91), which on its own would point to the four-hop scale, even though the four-hop shell on its own carries little node-specific signal (+0.43); the high cumulative value is mostly retained two-hop signal plus the variance reduction of averaging more nodes. On DBLP the cumulative understates the higher-order signal (+0.24 at $k = 2$ against the shell's +0.41), because the high-mass but weakly-predictive one-hop shell dominates the ball's average. On HumanGo the two track closely up to $k = 2$ but the cumulative again stays elevated where the shell falls away (+0.64 versus the shell's +0.40 at $k = 4$), a milder echo of the PPI pattern.

The cumulative readout is therefore less suited to this paper's task of locating the most predictive scale. On PPI it relocates the apparent peak outward, from the shell's two hops to four hops. On DBLP it keeps the peak at two hops but compresses it so far toward the one-hop value (a rise of only 0.04, from +0.20 to +0.24) that the higher-order signal is all but flattened away, whereas the per-shell view marks the same two-hop peak at its full strength (+0.41). The mechanism is simple: the cumulative score is a size-weighted average of shells, so widening the radius re-mixes the blend rather than adding a fresh measurement, and the curve is pulled up by an informative new shell or diluted by an uninformative one, which is why it rises then falls instead of trending and why the direction of the shift follows each graph's shell-size distribution.

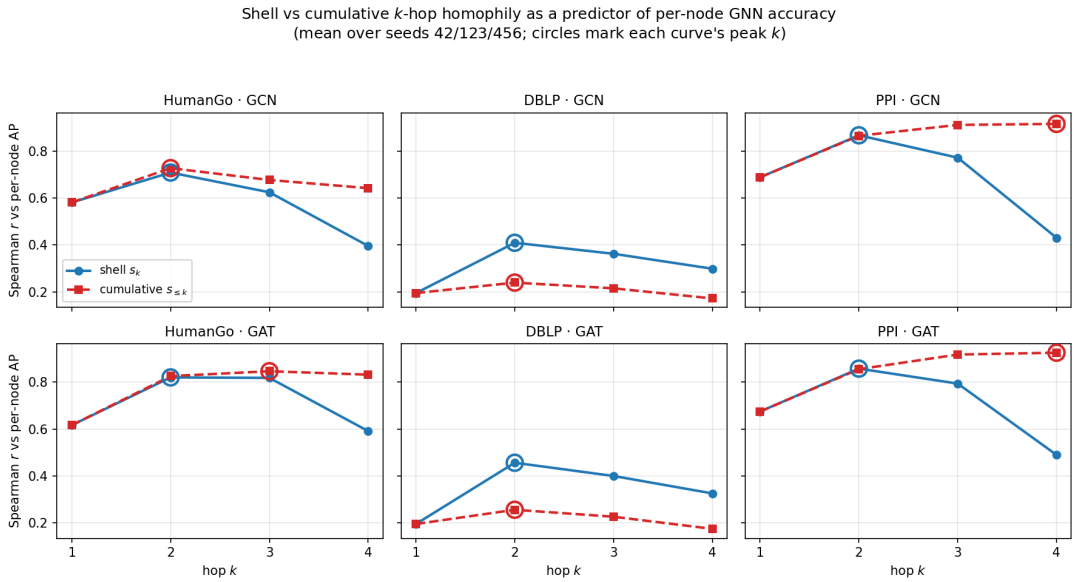

Figure 4: Shell (s_k, solid) versus cumulative ($s_{\leq k}$, dashed) k-hop homophily as a predictor of per-node accuracy (within-dataset Spearman r vs per-node AP, two-layer GCN, mean over seeds 42/123/456); circles mark each curve's maximum. On PPI (right) the cumulative climbs to a later maximum at $k = 4$ (+0.91) while the shell falls back (+0.43); on DBLP (centre) the cumulative sits well below the shell across all k (maximum +0.24 at $k = 2$ versus the shell's +0.41); on HumanGo (left) the two track closely at low k and the cumulative stays mildly elevated at $k = 3, 4$ (+0.64 versus the shell's +0.40).

4.4 The predictive scale does not track model depth (SQ3)

That two hops is most predictive is consistent with a receptive-field story (a two-layer GNN aggregates over two hops), but "consistent with" is not "caused by". Two readings compete: a model-driven one (the most predictive distance follows depth, so it would move to $k = 1$ for a one-layer model and $k = 3$ for a three-layer model) and a dataset-driven one (the two-hop shell is intrinsically the most informative scale, independent of depth). We discriminate by repeating the experiment at $L \in \{1, 2, 3, 4\}$ and

extending homophily to $k = 4$.

The receptive-field-shift prediction fails. Of 24 (dataset, model, depth) cells, not one is strongest at $k = L$ when $L \in \{1, 4\}$. The one-layer rows are never strongest at $k = 1$, and the four-layer rows are strongest at $k = 2$ in five of six cells; the $k = 4$ column is never the strongest and is the strict minimum across all k in every PPI cell (Figure 5). The strongest correlation stays anchored near $k = 2$ regardless of depth, so the dataset-driven reading is what the data supports. Depth modulates the flanks of the curve and, on DBLP, sharpens the maximum (the $k = 2$ correlation rises with depth, GCN from +0.41 at $L = 2$ to +0.45 at $L = 4$ and GAT from +0.46 to +0.51), but it does not move which distance is most predictive. The one exception that proves the rule is deep GAT on HumanGo, our smallest graph: at $L = 3, 4$ its per-node AP becomes seed-unstable (mean 0.53 and 0.39, against ≈ 0.60 at shallow depth), the textbook signature of oversmoothing, which the multi-seed protocol correctly surfaces as a wide error band rather than hiding behind one run.

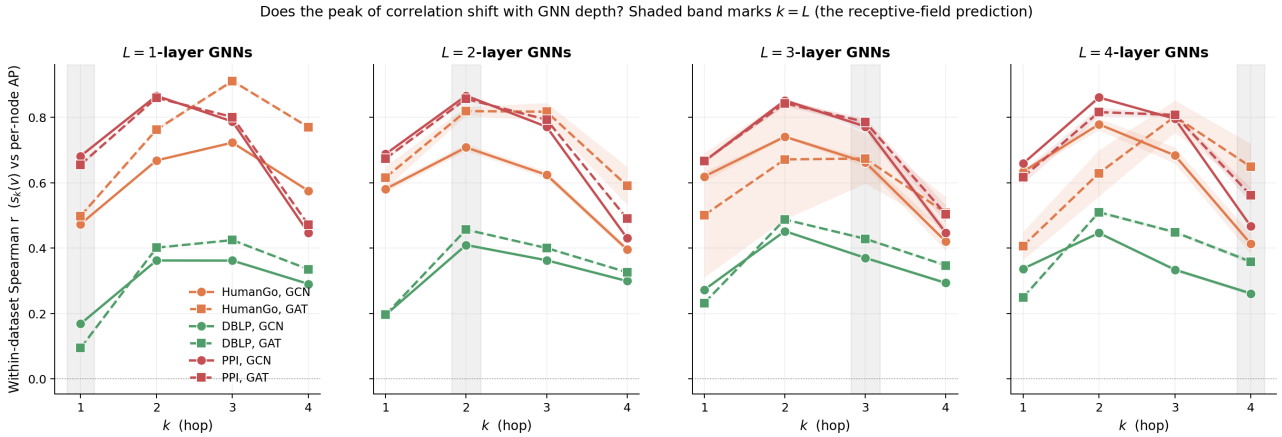

Figure 5: Within-dataset Spearman r between per-node $s_k(v)$ and per-node AP as a function of distance k, with one panel per GNN depth $L \in \{1, 2, 3, 4\}$ and one curve per (dataset, model) (mean over seeds 42/123/456). The grey band marks $k = L$, the distance the receptive-field hypothesis predicts should be most predictive: if that hypothesis held, each panel's curves would be highest inside its own grey band. Instead the curves are highest at $k = 2$ in every panel, so the most predictive distance is set by the graph rather than by the model's depth. Depth changes only how sharply the correlation falls away on either side of $k = 2$ (and destabilises deep GAT on the small HumanGo graph), not which distance is most predictive.

4.5 Causal evidence from synthetic graphs (SQ4)

The previous sections show that two-hop homophily correlates with per-node accuracy, but correlation alone cannot tell us whether the homophily is causing the accuracy or whether some third property of the data (a feature signal that happens to coincide with the local label structure, for instance) is driving both. The synthetic sweeps close that gap by holding everything constant except the two-hop or three-hop homophily and reading off how accuracy responds. Because the generators use Gaussian-noise node features, the model has no choice but to rely on topology, so if accuracy moves with H_k in these graphs, the homophily itself is the cause.

In the hub-spoke sweep that controls H_2 directly (with H_1 held flat at ≈ 0.18, verified by calibration), raising the two-hop homophily from 0.2 to 1.0 lifts test Micro-F1 from ≈ 0.47 (chance) to ≈ 0.96 for GCN at depth $L \geq 3$ and ≈ 0.81 for GAT (Figure 6). The receptive-field prediction holds for GCN: the one-layer line is flat at 0.47 across the sweep while $L \geq 2$ models climb steeply. The one-layer GAT is not flat (it rises to ≈ 0.62), which reflects not two-hop reach but a shortcut: each hub is a fixed node

shared between the training and test spokes, so a one-layer model can memorise its feature signature, and GAT's attention exploits this where GCN's averaging does not. The hexagon-cycle sweep that controls H_3 shows a smaller but real effect (Micro-F1 up by ≈ 0.20), though a weaker isolation than hub-spoke: sharing only three prototypes globally also lifts H_1 and H_2 to ≈ 0.30–0.35 at two of the three seeds (Appendix B.4), and a 6-cycle's local structure is already weakly position-identifying, so its depth separation is muddier. Together the two sweeps establish that, in graphs built to isolate it, higher-order homophily causes accuracy changes rather than merely co-varying with them, and three features make them informative beyond the observational study. The response is graded: accuracy rises monotonically across the five controlled H_k levels, the dose-response signature of a causal driver. The direction is fixed by construction, since H_k is set before training, so the effect can only run from homophily to accuracy and not the reverse. And the effect is gated by depth much as the mechanism predicts: the one-layer GCN cannot reach the two-hop signal and stays at chance while deeper GCNs climb, which both rules out a generic confound and shows the gain flows through the receptive field. The one-layer GAT is the sole model that rises at depth one, and it does so through the hub-memorisation shortcut identified above rather than through two-hop reach, so it is consistent with the same mechanism rather than a counterexample to it.

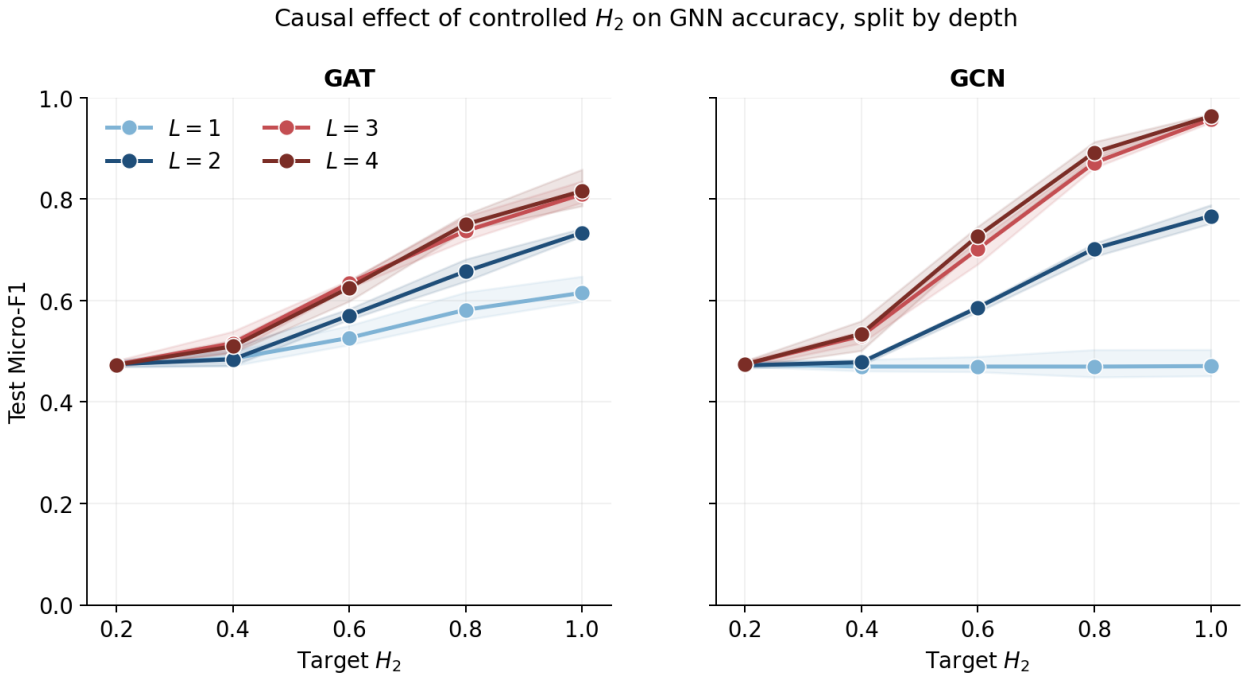

Figure 6: Test Micro-F1 versus the directly controlled two-hop homophily H_2, faceted by architecture and coloured by depth L. The one-layer GCN line is flat (a single hop cannot reach the two-hop signal); deeper GCNs climb steeply with H_2. The one-layer GAT line rises rather than staying flat, a shortcut discussed in the text.

5. Discussion

Answering the research question. The four subquestions combine into a single coherent answer. On **SQ1**, higher-order label similarity does predict per-node performance, and more strongly than one-hop similarity: the signal is strongest at two hops, consistently across two architectures, four depths, three seeds, and from roughly one to five thousand nodes per dataset (Section 4.2). On **SQ2**, we recover that two-hop scale only when we measure similarity at an exact distance (the shell); if

we instead pool every neighbour within k hops (the cumulative ball), the distance that looks most predictive moves away from two hops, and whether it moves inward or outward depends on the graph (Section 4.3). On **SQ3**, per-node performance stays best correlated with two-hop homophily even when the GNN is deeper than two layers, so the most predictive distance is set by the graph rather than by a two-layer receptive field (Section 4.4). On **SQ4**, controlled synthetic interventions confirm that, in graphs built to isolate it, higher-order homophily causes the accuracy change rather than merely co-varying with it (Section 4.5). This is a sharper answer than the prior dataset-level, one-hop analysis could give, which could not even resolve different values of k.

Model-driven versus data-driven. The depth experiment is the crux of the argument. If two hops were the most predictive distance only because we use two-layer models, that distance would move outward as we add layers; it does not. The two-hop shell is the most informative scale of these particular graphs, and a GNN's depth changes how sharply the correlation falls away on either side of that scale without relocating it. This reframes how higher-order homophily should be reported: not as a single number per graph, but as a profile over k whose maximum characterises the data.

Critical comparison with prior work. Our finding both extends and qualifies the homophily-performance link. It extends Zhao et al. by showing their one-hop metric is not the most predictive scale, and it sits beside the heterophily literature, which already argued that higher-order neighbourhoods carry useful signal, though in the single-label regime and as motivation for new architectures rather than as a measured property of data. Against a naive deeper-is-better reading, the over-smoothing results predict, and we observe, that simply stacking layers does not let a model exploit ever-more-distant homophily (the deep-GAT-on-HumanGo instability). The Section 4.3 contrast is, to our knowledge, a novel cautionary point: a cumulative neighbourhood, the seemingly natural way to look further, can shift which distance appears most predictive, by an amount set by the graph's shell-size distribution.

Threats to validity. The observational study uses a single hyperparameter family (Zhao's) and three seeds, so absolute correlation levels would shift under other settings even if the ranking across k is stable. The synthetic generators are deliberately simple: they isolate single shells to show the mechanism can operate, not that real graphs are built this way. The hub-spoke and hexagon-cycle graphs are disconnected, degree-regular, randomly labelled at the hubs, noise-featured, and decouple homophily across distances, none of which holds in real graphs; these choices are what make the causal isolation clean. We report both observational and causal evidence so each covers the other's weakness.

6. Responsible Research

Reproducibility. Every result is deterministic given the released code and the fixed seeds (42, 123, 456). A four-script pipeline (train, compute per-node homophily, analyse, plot) runs end to end with fixed filenames, and the shell and cumulative experiments differ only by a single `--cumulative` flag; crucially both reuse the same trained models, so the comparison is controlled rather than a confound of two separate setups. The exact commands are in Appendix A. We anchor the metric externally by reproducing Zhao et al.'s one-hop homophily to two decimals (Section 4.1), and all datasets are public and loaded through standard library splits [4, 10].

Honest reporting. We report the case where the methodological choice barely matters (HumanGo in Figure 4) rather than only the cases that favour our argument, and we surface the deep-GAT instability as a wide error band rather than dropping the unstable runs. The multi-seed protocol exists precisely so that a single lucky or unlucky run cannot drive a conclusion.

Use of AI tools. Generative AI assistance was used in three bounded ways: to help draft and

refactor some analysis and plotting scripts, to summarise and extract aggregate values from the raw result tables and model outputs, and to improve the wording of this manuscript. Choosing which analyses to run and interpreting every result remained with the author, who checked each AI-assisted artefact: code was read in full and validated against the reported numbers, extracted values were reconciled against their source, and all text was reviewed against the underlying data. The author takes full responsibility for the code, results, and claims in this paper.

Ethical aspects. The work is methodological and uses only established public benchmarks (citation networks, a protein-interaction graph, and a review network) with no personal or sensitive data introduced by us. The protein and citation graphs carry no individual-level privacy risk in the form used here. A broader caution applies to any homophily-based analysis: a finding that "similar neighbours predict accurate predictions" can, if applied uncritically to social graphs, reinforce the homophily it measures, so deployment on human-subject graphs would warrant fairness scrutiny beyond the scope of this study.

7. Conclusions and Future Work

We asked whether higher-order label similarity predicts how well a GNN classifies a node, and at which scale. It does: the two-hop scale is the most predictive (SQ1), read correctly only on the exact-distance shell (SQ2); the scale is a property of the graphs rather than of two-layer models, since it survives changes in depth (SQ3); and synthetic interventions show the relationship is causal in graphs built to isolate it (SQ4). The finding holds for both GCN and GAT, across four depths and three seeds, observational on three benchmarks and causal under intervention.

The broader message is that higher-order homophily should be reported not as a single edge-level number but as a profile over distance, whose maximum characterises a graph and, our results suggest, anticipates where a model finds its signal. Measured on exact-distance shells, that profile is a parameter-free, model-independent diagnostic computable before training. We are deliberate about scope: these conclusions rest on three multi-label benchmarks and on synthetic graphs that isolate single shells, so they show the mechanism operates and matters here, not that every graph is built this way (the limitations in Section 5 mark where that scope ends). We highlight three directions.

From diagnostic to architecture. If the most predictive scale is a property of the data, a model that explicitly targets it should beat fixed-depth stacking: a shell-weighted GNN that aggregates each exact-k shell separately and learns a per-shell weight, or that uses the measured k-hop profile to set its depth and per-hop attention. For these datasets our result predicts most weight on the two-hop shell, turning the profile from a post-hoc explanation into a design knob. This is the most promising follow-up.

Generalising beyond GCN and GAT. The dataset-driven reading predicts the scale should be largely architecture-independent; the clean test is to rerun the per-node analysis on heterophily-aware and structurally different models (H2GCN [7], Geom-GCN [6], GraphSAGE [4], graph isomorphism networks, graph transformers) and check whether the strongest correlation stays at two hops. A shift would show the scale is partly architectural, sharpening the conclusion either way.

Richer causal control. Our generators plant structure at one shell. A cleaner three-hop generator (whose one- and two-hop neighbourhoods are individually uninformative) would measure the $k = 3$ depth separation as crisply as at $k = 2$, and a multi-scale generator planting competing signals at two and three hops would reveal which scale a GNN prioritises when the data is genuinely multi-scale.

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 (TMLR), 2023. arXiv:2304.10398.

[2] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations (ICLR), 2017. arXiv:1609.02907.

[3] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. International Conference on Learning Representations (ICLR), 2018. arXiv:1710.10903.

[4] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive Representation Learning on Large Graphs. Advances in Neural Information Processing Systems (NeurIPS 30), 2017. arXiv:1706.02216.

[5] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology, 27:415–444, 2001. doi:10.1146/annurev.soc.27.1.415.

[6] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang. Geom-GCN: Geometric Graph Convolutional Networks. International Conference on Learning Representations (ICLR), 2020. arXiv:2002.05287.

[7] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. Advances in Neural Information Processing Systems (NeurIPS 33), 2020. arXiv:2006.11468.

[8] Q. Li, Z. Han, and X.-M. Wu. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1):3538–3545, 2018. arXiv:1801.07606.

[9] K. Oono and T. Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. International Conference on Learning Representations (ICLR), 2020. arXiv:1905.10947.

[10] M. Fey and J. E. Lenssen. Fast Graph Representation Learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019. arXiv:1903.02428.

[11] F. Pedregosa et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[12] M. Zitnik and J. Leskovec. Predicting Multicellular Function through Multi-layer Tissue Networks. Bioinformatics, 33(14):i190–i198, 2017. doi:10.1093/bioinformatics/btx252.

[13] Y. Choi, J. Choi, T. Ko, and C.-K. Kim. Hierarchical Uncertainty-Aware Graph Neural Network. 2025. arXiv:2504.19820.

[14] J. Rubin, S. Loomba, and N. S. Jones. Limits of Message Passing for Node Classification: How Class-Bottlenecks Restrict Signal-to-Noise Ratio. 2025. arXiv:2508.17822.

[15] J. H. Giraldo, K. Skianis, T. Bouwmans, and F. D. Malliaros. On the Trade-off between Over-smoothing and Over-squashing in Deep Graph Neural Networks. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM), 2023. arXiv:2212.02374.

[16] P. Jaccard. The distribution of the flora in the alpine zone. New Phytologist, 11(2):37–50, 1912. doi:10.1111/j.1469-8137.1912.tb05611.x.

Appendix A. Reproduction pipeline

Both the shell experiment (Sections 4.2, 4.4) and the cumulative experiment (Section 4.3) run from fixed-filename scripts under `MLGNC/` (paths relative to the repository root). The shared metric, `compute_per_node_homophily` in `khop_similarity.py`, averages Jaccard over the exact shell $N_k(v)$, or over the ball $B_k(v)$ with `--cumulative`; self-loops are stripped so a node never lands in its own shell. The full pipeline is:

```
python MLGNC/per_node_train.py
python MLGNC/per_node_homophily.py --datasets HumanGo DBLP PPI
python MLGNC/per_node_homophily.py --datasets HumanGo DBLP PPI --cumulative
python MLGNC/per_node_analysis.py
python MLGNC/per_node_analysis.py --cumulative
python MLGNC/per_node/cumulative_vs_shell.py
```

In order, these train GCN/GAT at $L = 1 \dots 4$ under seeds 42/123/456; compute the per-node shell score $s_k(v)$ and then the cumulative score $s_{\leq k}(v)$; produce the correlation tables `results_multiseed.csv` (Table 2) and `results_multiseed_cum.csv`; and draw the shell-versus-cumulative comparison (Figure 4). The cumulative variant reuses the same trained models, so only the predictor changes between the two readings. `cumulative_vs_shell.py` only reads the two `results_multiseed*.csv` tables, so it must run last, after both analyses are refreshed.

Appendix B. Synthetic causal experiments: construction and calibration

Both generators (in `MLGNC/causal_sweep.py`) build graphs whose only useful signal is topology, with homophily concentrated at one chosen distance: distance 2 (the hub-spoke generator) and distance 3 (the hexagon-cycle generator).

B.1 Design

A purity knob $\pi \in [0, 1]$ (exposed as $\alpha = \pi/(1 - \pi)$) raises the targeted H_k from baseline (≈ 0.18 for our label density) toward 1 while the other distances stay near baseline. Node features are 64-dimensional standard-normal noise, $\mathcal{N}(0, I_{64})$, carrying no label information, so accuracy can only come from topology. Both sweeps train GCN and GAT at $L \in \{1, 2, 3, 4\}$, three seeds (42/123/456), with the Section 3.3 optimiser (GCN hidden width 128 for these 2000-node graphs), a 60/20/20 split, early stopping on validation loss, and a validation-tuned decision threshold; we report test Micro-F1. Each generator runs at five targets $h \in \{0.2, \dots, 1.0\}$, giving $5 \times 3 \times 4 \times 2 = 120$ runs per sweep.

B.2 The hub-spoke generator (controlling H_2)

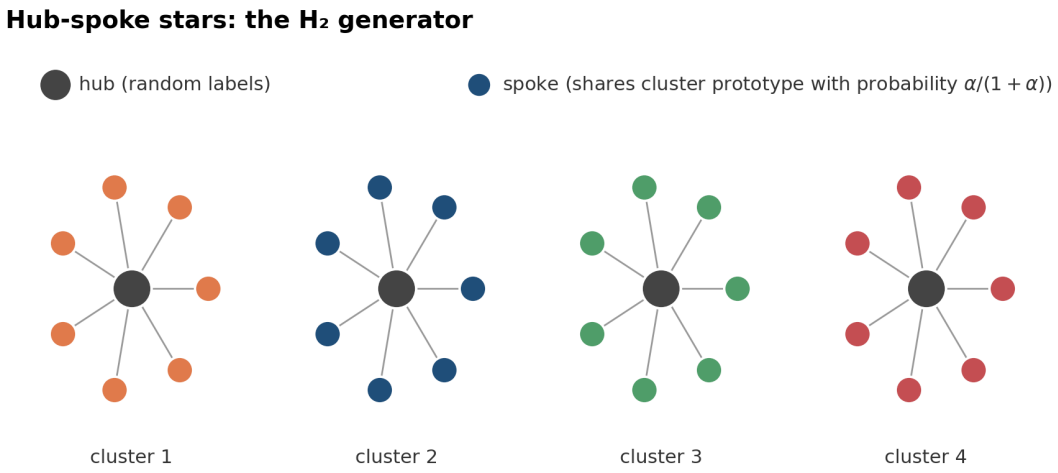

Figure 7: The hub-spoke generator: disjoint stars in which spokes of a common hub are distance-2 pairs. Each spoke copies its cluster prototype with probability π, lifting H_2; hubs are random, so H_1 stays at baseline.

Figure 7 sketches the layout. **Algorithm 1. Hub-spoke generator (targets H_2).** Inputs: 2000 nodes, 20 spokes per hub, a purity knob π between 0 and 1, and $L = 10$ labels.

1. Build disjoint stars, each one hub connected to 20 spokes with no edges between spokes. Two spokes of the same hub are then exactly two hops apart.
2. Give each star a random prototype label set (each of the 10 labels included independently with probability 0.3).
3. For each spoke, with probability π copy its star's prototype, otherwise give it a fresh random label set.
4. Give every hub a fresh random label set, so one-hop (hub-spoke) pairs share no planted structure and H_1 stays at baseline.
5. Give every node 64 dimensions of Gaussian noise as features, so labels can be recovered only from the graph, not from the features.

Two spokes of the same hub both copy the prototype with probability π^2, so raising π drives their two-hop homophily H_2 smoothly from baseline up toward 1, while the one-hop homophily stays flat.

B.3 The hexagon-cycle generator (controlling H_3)

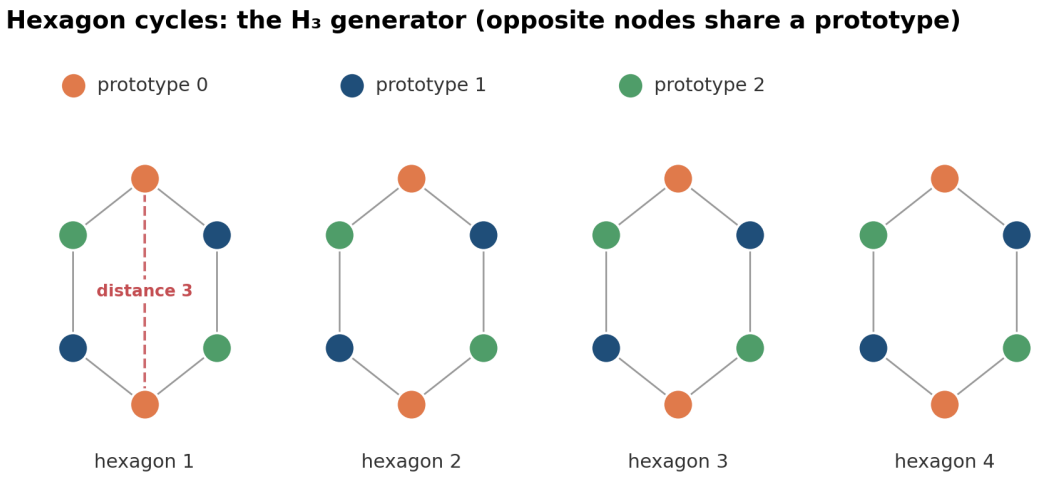

Figure 8: The hexagon-cycle generator: disjoint 6-cycles with three prototypes shared by position modulo 3. Opposite nodes (distance 3) share a prototype, lifting H_3; distance-1 and distance-2 pairs use different prototypes, so H_1 and H_2 stay near baseline (up to residual prototype overlap, Appendix B.4).

Figure 8 sketches the layout. **Algorithm 2. Hexagon-cycle generator (targets H_3).** Inputs: 2000 nodes, a purity knob π between 0 and 1, and $L = 10$ labels.

1. Build disjoint 6-cycles (hexagons). Opposite nodes of a hexagon are exactly three hops apart.
2. Create three prototype label sets shared across all hexagons, and lay them around each cycle in the repeating pattern 0, 1, 2, 0, 1, 2.
3. For each node, with probability π copy the prototype of its position, otherwise give it a fresh random label set.
4. Give every node 64 dimensions of Gaussian noise as features.

Opposite nodes sit three positions apart and so share the same prototype, so raising π drives the three-hop homophily H_3 toward 1; neighbouring and two-hop pairs sit at different positions with different prototypes, so H_1 and H_2 stay near baseline (up to small accidental overlap between the three prototypes, Appendix B.4). Sharing the three prototypes across all hexagons rather than per hexagon makes the planted signal transferable to held-out nodes.

B.4 Calibration

Because α maps to the realised H_k monotonically but not linearly, we calibrate: for each generator we sweep α over a wide purity grid, measure the achieved H_k at the targeted shell, and pick the α closest to each target $h \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. Calibration runs once at seed 1234 so all data seeds share the same α per target. Figures 9 and 10 show the achieved H_k tracking the target almost diagonally while the other distances stay near baseline.

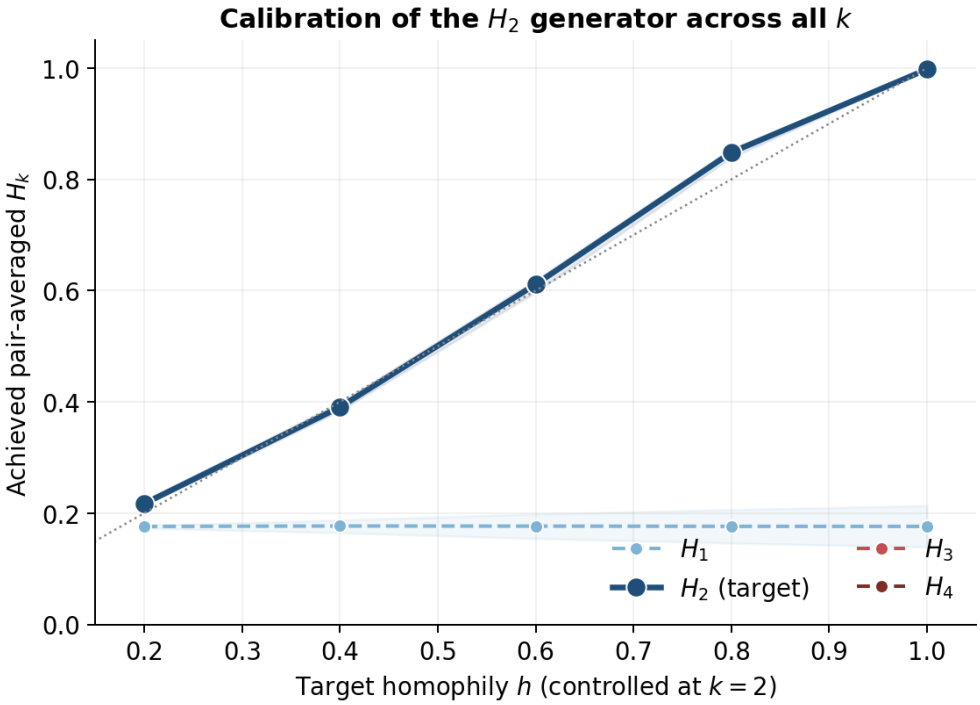

Figure 9: Hub-spoke calibration: achieved H_2 tracks the target while H_1 stays at baseline ≈ 0.18. Distances 3 and 4 do not arise within a cluster.

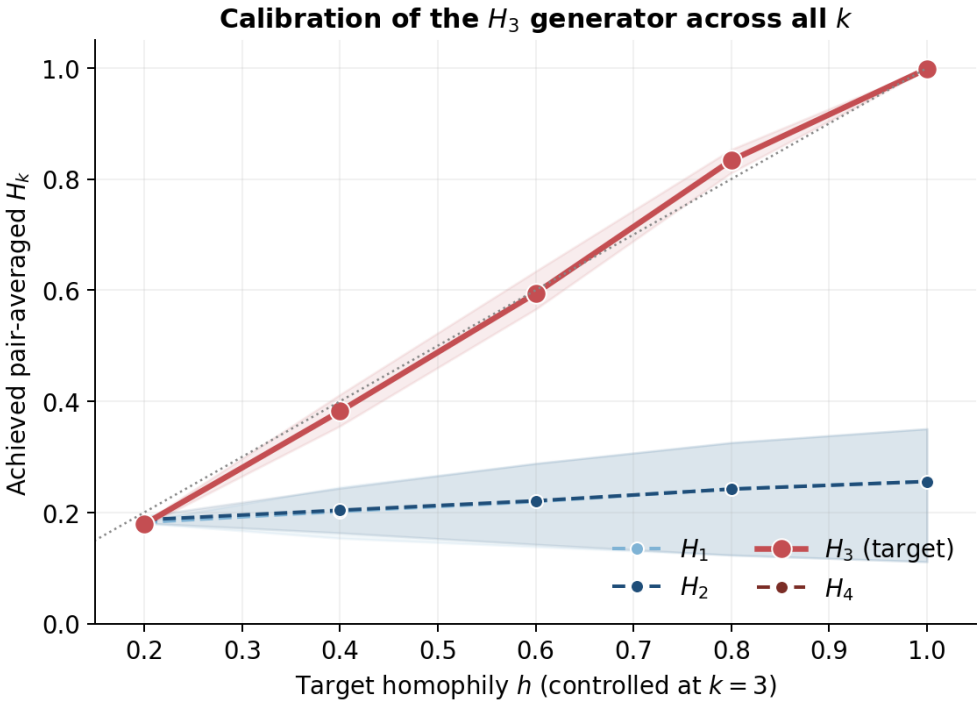

Figure 10: Hexagon-cycle calibration: achieved H_3 tracks the target. At the experiment seeds, residual overlap between the three shared prototypes lifts H_1 and H_2 to ≈ 0.30–0.35 at full purity for two of three seeds, a less clean control than the hub-spoke sweep.

B.5 Receptive-field cross-check

A one-layer GNN sees only one hop, so it should stay flat as the targeted two- or three-hop homophily varies, while deeper models respond. Figure 6 (Section 4.5) confirms this for the hub-spoke GCN: the $L = 1$ line is flat at ≈ 0.47 while $L \geq 2$ models climb. Figure 11 shows the hexagon result, where the shallow/deep separation is softer because a 6-cycle's local structure is already weakly position-identifying, but deeper models still track H_3.

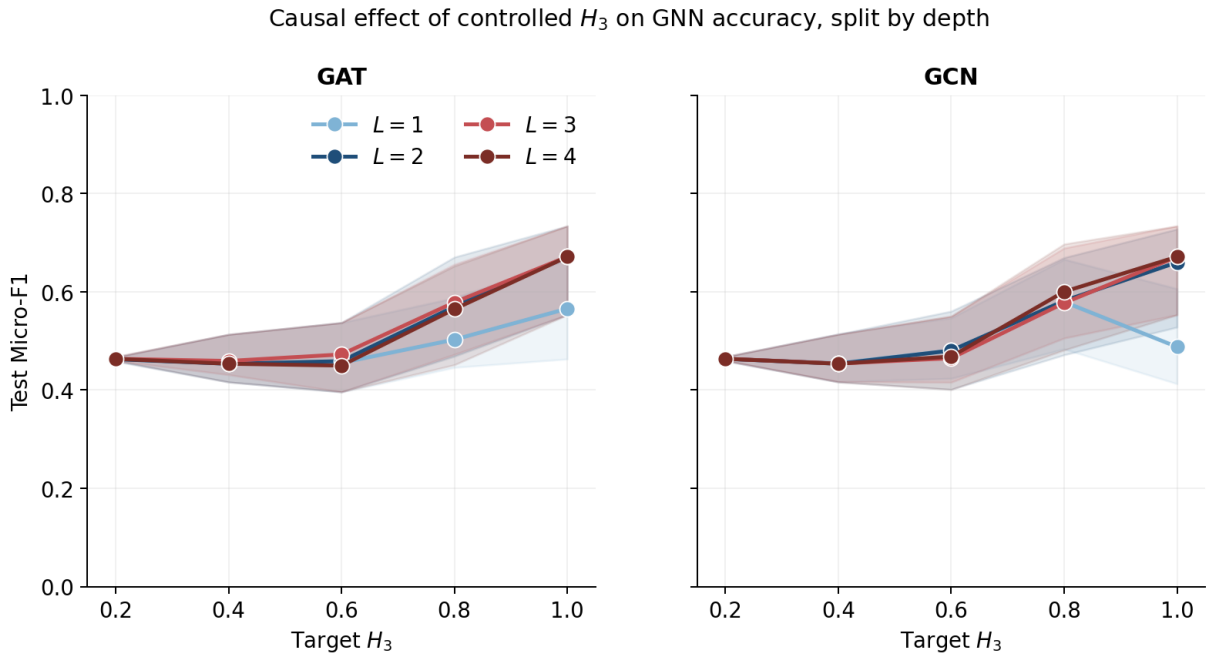

Figure 11: Test Micro-F1 versus the controlled three-hop homophily H_3, by architecture and depth L. Both GCN and GAT climb with H_3 at $L \geq 3$; shallow models climb more than in the hub-spoke sweep because a 6-cycle already carries some positional signal. Higher H_3 gives higher accuracy, confirming a causal effect.

B.6 Reproduction

A single command runs both sweeps end to end:

```
python MLGNC/causal_sweep.py --only both
```

It writes `causal_h2_results.csv` and `causal_h3_results.csv` (per-configuration H_1–H_4, the chosen α, the threshold, and Micro-F1/macro-F1/mean AP) plus the figures above; `--only h2` or `--only h3` restricts to one sweep.