Property-Driven Comparison of GNNs on Multi-Label Graphs

Victor Paiu[1]
Supervisor(s): Megha Khosla[1]**, Elena Congeduti**[1]
[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 16, 2026

Name of the student: Victor Paiu
Final project course: CSE3000 Research Project
Thesis committee: Megha Khosla, Elena Congedutir, Christoph Lofi

Abstract

Multi-label node classification on graphs occurs in domains where entities can have several labels, such as biological, social, and recommendation networks. Most Graph Neural Networks (GNN) research focuses on multi-class graphs, so it remains unclear how dataset properties affect model performance in multi-label settings. This thesis studies how structural, feature, and label properties influence Graph Convolutional Network (GCN) and Heterophilic Graph Convolutional Network (H2GCN). These models were chosen because they are widely used and represent homophilous and heterophilous graph learning, respectively. Synthetic graphs are used to vary their properties in a controlled way, with real-world datasets used as validation points, and a pooled Ridge regression then tests how well each property predicts model performance in a joint setting. The results show that no single property explains performance solely by itself. Label imbalance reduces both models similarly, structural noise harms GCN more, unlabeled nodes degrade the performance of H2GCN more quickly, and cross-class neighbourhood similarity adds information beyond homophily. All code, seeds, and trained-graph properties are released publicly.[1]

1 Introduction

Graph-structured data is used in many contexts where the relationship between entities is important, such as social network analysis, recommendation systems, and drug discovery. In these settings, node classification is often used to infer missing information from both feature data and graph structure. Graph Neural Networks (GNNs) are widely used for this task because of their ability to learn from connected data (Kipf & Welling, 2017), while taking into account both dimensions.

Most existing graph classification research studies settings where each node can have only one label – multi-class graphs. This leaves a less explored but nonetheless important question: how do graph-based methods behave when nodes belong to multiple classes simultaneously? Prior work has shown that multi-label graph learning requires its own data, adapted learning methods, and evaluation metrics (Zhao et al., 2023). However, there is still a limited understanding of how graph or feature properties affect the performance of GNNs, and when one method performs better than another.

This thesis addresses that gap through a controlled analysis study of how existing methods respond when graph, feature, and label properties are changed. This is important because benchmark results alone do not show whether the performance difference is caused by graph structure, label imbalance, feature dimensionality, or a combination of these factors.

We focus on two GNN architectures with different inductive biases: GCN (Kipf & Welling, 2017), which works best when connected nodes share labels, and H2GCN (Zhu et al., 2020), designed for graphs where they often do not. This contrast, together with their widespread use, makes them a clean pair for testing.

The main research question is:

> **How do structural, feature, and label properties of multi-label graphs influence the performance of GCN and H2GCN?**

We decompose this question into four sub-questions:

[1] https://github.com/VictorP4/mlgnc-properties

SQ1. Which structural properties of multi-label graphs, such as homophily, density, clustering, label informativeness, and cross-class neighborhood similarity, show the strongest relationship with model performance?

SQ2. Which feature- and label-side properties, such as feature dimensionality, label dimensionality, mean label cardinality, per-label imbalance, and unlabeled fraction, show the strongest relationship with model performance?

SQ3. Under which property conditions does the performance gap between GCN and H2GCN widen or narrow, and does this align with their design assumptions?

SQ4. To what extent do trends observed on synthetic data carry over to real-world multi-label graphs, and which real-world phenomena are not captured by the synthetic generator?

To answer this question, this thesis uses synthetic multi-label graphs to vary specific properties in a controlled way, and then compares results with real-world multi-label graphs where possible. The experiments study feature and label dimensions, label imbalance, structural noise, homophily, and more. Across these studies, the focus is on the model performance of GCN and H2GCN, and the performance gap between these models. Finally, a pooled Ridge regression is used to analyse these properties jointly, testing how strongly each property predicts model performance while controlling for the others. This provides a broader view of which graph characteristics are most informative for explaining when each model performs well.

The results show that performance in multi-label graphs cannot be explained by a single property alone. Label imbalance degrades the performance of both GCN and H2GCN while keeping the performance gap similar, structural noise harms GCN more, and cross-class neighbourhood similarity provides additional information beyond homophily. These findings suggest that model choice in the context of multi-label graphs should consider combinations of dataset properties rather than just a single summary statistic.

The rest of the thesis is organized as follows. Section 2 introduces the notation and background. Section 3 describes the methodology and the experiment decisions and design. The following sections present the experiments and results, discussing responsible research considerations, interpret the findings, and finish with the limitations and suggestions for future work.

2 Background

2.1 Notations

Let $G = (V, E)$ be an undirected graph with $N = |V|$ nodes, $|E|$ edges, and ordered edge set E_{ord} of size $2|E|$. Each node u has feature vector $X_u \in R^{|F|}$ and binary label vector $y_u \in \{0, 1\}^{|C|}$; its label set is $\ell(u) = \{c : y_{uc} = 1\}$, with $|\ell(u)| \geq 1$ in the multi-label setting. We write $N(u)$ for the neighborhood of u, $V_c = \{v : c \in \ell(v)\}$ for the set of nodes carrying class c, and $\bar{\ell} = \frac{1}{N} \sum_u |\ell(u)|$ for the mean label cardinality.

2.2 Multi-Label Node Classification

Graphs are commonly used to represent connected data, where nodes represent entities and edges represent relationships between them. A standard machine learning task on graphs

is node classification, where the goal is to predict labels for unlabeled nodes using both node features and graph structure. This area of research has seen significant progress in recent years due to its usefulness across a multitude of scenarios: social network analysis, biological network analysis, drug discovery, recommendation systems, and more. The most well-known and used methods for solving this task are Graph Neural Networks (GNNs).

Most existing work focuses on the multi-class setting, where each node has exactly one label. However, many real-world graph datasets are naturally multi-label. For example, proteins can have multiple biological functions, and users in social or recommendation networks may belong to several interest categories at the same time. In such cases, a node can be assigned multiple labels, making the task more realistic but also more complex.

2.3 Properties for Multi-Label Graphs

The performance of graph-based methods often depends on dataset properties. In multi-class graphs, homophily usually means that connected nodes tend to have the same label. In multi-label graphs, this idea is more complex because two connected nodes may share only some of their labels. Therefore, low homophily in a multi-label graph does not necessarily mean that neighborhood information is useless.

To address this, Zhao et al., 2023 define **multi-label homophily** using the average Jaccard similarity between the label sets of connected nodes. In particular, the multi-label homophily of a graph is defined as:

$$h = \frac{1}{|E|} \sum_{(i,j)\in E} \frac{|\ell(i) \cap \ell(j)|}{|\ell(i) \cup \ell(j)|}.$$

They also introduce **Cross-Class Neighborhood Similarity** (CCNS) (Zhao et al., 2023), which compares the neighborhood label distributions of nodes that carry each pair of classes. For each pair of classes (c, c'),

$$\mathrm{CCNS}(c, c') = \frac{1}{|V_c|\,|V_{c'}|} \sum_{u\in V_c,\, v\in V_{c'}\, u\neq v} \frac{\cos\big(d(u), d(v)\big)}{|\ell(u)|\,|\ell(v)|},$$

where $V_c = \{v : c \in \ell(v)\}$ is the set of nodes carrying class c, $d(u) = \sum_{w\in N(u)} \ell(w)$ is the neighborhood label-sum vector of u. The factor $1/(|\ell(u)|\,|\ell(v)|)$ normalises the contribution of multi-labeled nodes so that a node with many labels does not dominate the entries of every class pair it belongs to. CCNS is useful because it captures neighborhood-level label structure that scalar homophily collapses into a single number.

A second alternative to scalar homophily is **label informativeness** (LI), which measures how much a node's label tells you about its neighbours' labels. Platonov et al., 2024 introduced LI in the multi-class setting and showed it predicts GNN accuracy better than scalar homophily. We adapt LI to the multi-label setting using the same per-edge weighting as CCNS: for each ordered class pair (c, c'),

$$p(c, c') = \frac{1}{2|E|} \sum_{(u,v)\in E_{\mathrm{ord}}} \frac{1\{c \in \ell(u)\}\, 1\{c' \in \ell(v)\}}{|\ell(u)|\,|\ell(v)|},$$

with marginal $\bar{\pi}(c) = \sum_{c'} p(c, c')$, and

$$\mathrm{LI} = -\frac{\sum_{c,c'} p(c, c') \log \frac{p(c,c')}{\bar{\pi}(c)\,\bar{\pi}(c')}}{\sum_c \bar{\pi}(c) \log \bar{\pi}(c)}.$$

The $1/(|\ell(u)|\,|\ell(v)|)$ factor prevents nodes with many labels from dominating the sum, and when every node has one label LI reduces to Platonov's original $\mathrm{LI}_{\mathrm{edge}}$.

Finally, we measure per-graph label imbalance using the **Mean Imbalance Ratio** (MeanIR) from Charte et al., 2015:

$$\mathrm{MeanIR} = \frac{1}{|C|} \sum_{k=1}^{|C|} \frac{\max_j \mathrm{count}(j)}{\mathrm{count}(k)},$$

the mean over labels of how many times rarer each label is than the most common one (MeanIR= 1 is perfect balance).

2.4 Generators and Network Architecture

Graph Neural Networks are commonly used for node classification because they learn node representations by aggregating information from neighboring nodes. The two architectures studied in this paper are **GCN**, the neighborhood-aggregation baseline, and **H2GCN**, a heterophily-targeted variant, both of which are described in detail in subsection 4.4.

Synthetic data generation is useful for multi-label learning because it allows datasets with known properties to be created. **MLDataGen** is a framework for generating synthetic multi-label datasets (Tomas et al., 2014). This framework is used as part of a synthetic multi-label graph generator, where multi-label data is first generated and graph edges are then constructed using a social distance attachment model (Zhao et al., 2023).

3 Methodology

3.1 Why Synthetic Data, and the Role of Real-World Data

Real-world multi-label graphs vary along many axes at once: a dataset such as Yelp differs from DBLP in homophily, label cardinality, feature dimension, clustering, and unlabeled fraction simultaneously, as seen in Table 1. Such multi-dimensional variation makes it difficult to attribute a performance difference to any single property. Synthetic graphs solve this problem because it gives us more control over the graph properties when generating them. We therefore use synthetic graphs as the primary tool to partially isolate the effect of each property. Real-world graphs serve a complementary role: they anchor the synthetic findings to actual data. When a synthetic trend extends to the real-world data points, we treat that as external validation, and when it does not, the mismatch itself becomes a finding about the limitations of the graph generator.

3.2 General Experiment Template

Each controlled experiment follows the same template:

1. We pick one property to vary and identify the others we want to hold constant.
2. We generate a batch of synthetic multi-label datasets using the MLDataGen hypersphere model (Tomas et al., 2014) and connect them with the social-distance-attachment graph generator from Zhao et al. (2023), sweeping the target property across at multiple levels.

3. We measure the realised properties on every generated graph, including homophily h, density, clustering, mean label cardinality, label informativeness, and unlabeled fraction.

4. We train GCN and H2GCN on each graph for three random seeds with the hyperparameters from Zhao et al., 2023 (Table 7).

For evaluation, we report mean macro average precision (AP) and macro F_1, all averaged over three random seeds and reported with their seed-level standard deviation. AP is our primary metric because it is more reliable than AUC–ROC on highly imbalanced and sparsely labeled multi-label data (Zhao et al., 2023), macro F_1 is reported alongside it for control. For the controlled experiment, the performance-versus-property trend is summarised using a Pearson correlation coefficient.

4 Experiments

4.1 Synthetic Multi-Label Data and Graph Construction

Among the strategies in the MLDataGen framework (Tomas et al., 2014), we use the hypersphere strategy, which places $|C|$ hyperspheres in a $|F|$-dimensional feature space and assigns to each node the labels of every hypersphere whose region contains its sampled position. The hypersphere strategy gives us independent geometric control over the three axes our experiments need to vary: the feature dimension $|F|$ is set by the ambient space, the label dimension $|C|$ by the number of spheres, and the marginal frequency of each label by its sphere radius. Each can be moved without disturbing the others, which is what makes the controlled property sweeps in this paper possible. Unless an experiment varies one of these axes explicitly, we use the following defaults: $N = 3000$ nodes, $|F| = 10$, $|C| = 20$, label noise 0.05, and 10 irrelevant features.

Two additions on top of MLDataGen were necessary for the label-imbalance experiment. First, we expose a per-label radii option so that the imbalance ratio of the resulting label matrix can be tuned through a vector of sphere radii rather than just a maximum and minimum value, which is what allows the synthetic imbalance range to span the real-world range. Second, a post-generation step suppresses labels for a fixed fraction of nodes, giving us an axis for the unlabeled fraction of nodes. Both additions leave the rest of the generation pipeline untouched and are skipped by every other experiment.

For every generated multi-label dataset we attach a graph with the social-distance-attachment (SDA) model of Zhao et al., 2023 (their Eq. 2). SDA assigns each pair (u, v) a connection probability

$$p_{uv} \;=\; \frac{1}{1 + \big(d(y_u, y_v)/b\big)^{\alpha}},$$

where $d(y_u, y_v)$ is the normalised Hamming distance between the binary label vectors of u and v, and b and α control the sharpness of the probability curve. To produce a graph at a target multi-label homophily we sweep over a grid of b values and binary-search the matching $\alpha \in [0, 50]$ that brings the measured Jaccard homophily within a small tolerance of the target.

4.2 Real-World Datasets

We use the seven real-world multi-label graphs collected by Zhao et al., 2023, summarised in Table 1. Three of them come from biology (PCG, HumLoc, EukLoc cover protein functions and relationships), one from biological networks at a larger scale (OGB-Proteins, from the Open Graph Benchmark (Hu et al., 2021), where labels are protein functional annotations), one from co-authorship (DBLP, where labels are research areas), one from social relationships (BlogCat, where labels are bloggers' interest categories), and one from business-review graphs (Yelp, where labels are business categories). Their fields span computational biology, scientific publishing, and online social platforms, which gives the property table a wide spread in homophily, label cardinality, imbalance, and unlabeled fraction.

The properties in Table 1 are measured directly on the real-world graphs.

Table 1: Real-world multi-label graphs used in this paper. h is multi-label (Jaccard) homophily, LI is label informativeness, $\bar{\ell}$ is mean label cardinality, MeanIR follows Charte et al., 2015.

Dataset	N	$\lvert C\rvert$	h	LI	$\bar{\ell}$	MeanIR	Unlabeled nodes
BlogCat	10 312	39	0.10	0.010	1.40	15.4	0.0%
OGB-Proteins	132 534	112	0.15	0.006	12.75	6.4	40.3%
PCG	3 233	15	0.17	0.004	1.93	3.6	0.0%
Yelp	716 847	100	0.22	0.003	9.44	17.4	0.1%
HumLoc	3 106	14	0.42	0.079	1.19	15.3	0.0%
EukLoc	7 766	22	0.46	0.090	1.15	45.0	0.0%
DBLP	28 702	4	0.76	0.318	1.18	1.6	0.0%

4.3 Selected Methods: GCN and H2GCN

We compare GCN and H2GCN because they represent different approaches to using neighbourhood information: GCN aggregates neighbours directly, which works best when connected nodes share labels, while H2GCN separates the ego node representation from the one-hop and two-hop neighbourhood information, which are aggregated separately and concatenated together. This was specifically designed for graphs where neighbours often disagree (Zhu et al., 2020). Comparing them tells us when the one-hop neighbourhood is a good predictor of a node's labels, and each of our experiments varies a property that changes that. They are also the two architectures that Zhao et al., 2023 report on every real-world dataset in Table 1, so the published numbers give us an external anchor for our own runs. In the multi-class setting Zhu et al., 2020 reported that GCN's accuracy collapses below a homophily threshold of roughly 0.5 while H2GCN holds. The multi-label analogue of this collapse is one of the hypotheses we test directly in the homophily-sweep experiment.

We use the following hyperparameters for the GNNs: two layers, hidden dimension 256 for GCN and 64 for H2GCN, weight decay $5{\times}10^{-4}$, 300 epochs with early stopping (patience 30), and learning rate 0.01. Each model is trained with binary cross-entropy on a 60/20/20 random node split, repeated over three random seeds. Reported metrics are the seed-level mean with the standard deviation; macro AP is the primary number, with macro F_1 also included.

4.4 Summary of Controlled Experiments

We run five controlled experiments. Each varies one property axis at a time while attempting to hold the rest fixed, and reports the per-model AP as a function of the varied axis. The summary below lists the axis, fixed properties, hypothesis, and goal of each experiment. The full results are presented in section 5.

Homophily sweep on the Synthetic1 base. *Varies:* multi-label homophily $h \in [0.2, 1.0]$ over 9 levels. *Fixed:* $N = 3000$, 10 relevant, and 10 irrelevant features, $|C| = 20$, label noise 0.05. *Hypothesis:* GCN AP rises with h; H2GCN matches or exceeds GCN at low h and the gap narrows at high h. *Goal:* test whether the multi-class homophily collapse (Zhu et al., 2020) reproduces in the multi-label setting.

Random-edge addition with matched-h control. *Varies:* random-edge addition as a fraction of $|E|$, at five levels of 0, 10, 25, 50, and 100 percent, plus a matched-h SDA control. *Fixed:* pre-noise base at $h \approx 0.6$, $|F| = 10$, $|C| = 20$. *Hypothesis:* GCN collapses faster than H2GCN under random structural noise; clean SDA at the same realised h keeps performance higher than the noisy graph. *Goal:* separate scalar-h from neighbourhood-structure effects.

Label imbalance and unlabeled nodes. *Varies:* per-label imbalance from MeanIR 1.25 to 2.95, and unlabeled fraction $\in \{0, 20, 40\}\%$. *Fixed:* $h \approx 0.4$, $N = 3000$, $|C| = 20$. *Hypothesis:* imbalance hurts macro metrics more than micro; the H2GCN advantage is preserved under imbalance but eroded by unlabeled nodes. *Goal:* connect the synthetic gap behaviour to real-world datasets with high MeanIR (Yelp, BlogCat, EukLoc) and high unlabeled fraction (OGB-Proteins).

Label informativeness as a predictor. *Pool:* the nine synthetic graphs from the homophily sweep plus the seven real-world datasets. *Hypothesis:* LI carries information about neighbourhood label structure that scalar h does not summarise, so as a single predictor it should be at least as informative as h, replicating Platonov et al., 2024 in the multi-label setting. *Goal:* compare h and LI as single property predictors of AP.

Pooled Ridge regression across all graphs. *Varies:* predicts AP and macro F_1 from the standardised property vector $(h, \mathrm{LI}, \bar{\ell}, |C|, \log_{10}\text{density}, |F|, \text{unlabeled nodes}, \text{clustering})$ over every trained graph. *Fixed:* no controlled axis; the pool covers roughly 90 synthetic graphs and the 7 real-world datasets. *Hypothesis:* different properties survive as independent predictors of AP versus macro F_1, and the predictor ranking is model-specific. *Goal:* identify which graph properties carry independent predictive power once the others are accounted for.

The first four experiments are single-axis sweeps, and the fifth is the pooled meta-analysis that absorbs the correlation between axes that those sweeps cannot avoid. Each row of the table corresponds to one subsection of section 5, where we present the realised property values, the resulting performance curves, and the verdict on the hypothesis.

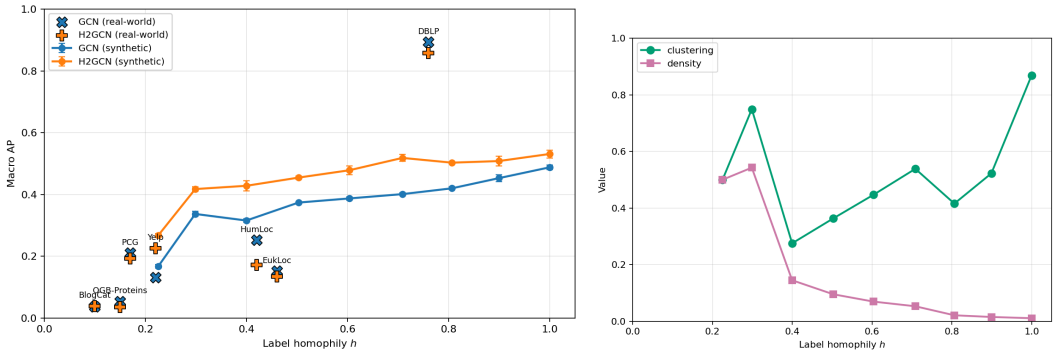
5 Results

We present the five experiments in the same order as the summary above. Each subsection states the setup in one sentence, shows the main figure or table, presents the headline

observation, and closes with the hypothesis verdict and the real-world connections, if there exists one.

5.1 Homophily Sweep on the Synthetic1 Base

We trained GCN and H2GCN on nine SDA graphs built on the Synthetic1 base of Zhao et al., 2023 at $h \in \{0.2, 0.3, \ldots, 1.0\}$, three seeds per cell. The realised h matches the target within ± 0.01 everywhere except $h = 0.2$, where it floors at ≈ 0.225.



(a) Macro AP versus h. Synthetic sweep (lines, circles) rises monotonically for both models; real-world datasets (crosses, plus signs) are overlaid at their measured h.

(b) Edge density and average clustering versus h on the same nine SDA graphs. Density drops $\sim 50\times$ across the sweep; clustering is non-monotonic.

Figure 1: Homophily sweep on the Synthetic1 base: (a) model performance and real-world overlay, (b) structural properties of the synthetic datasets

As shown in Figure 1a, both models rise monotonically with h and neither saturates. The Pearson correlation with h is $+0.90$ (GCN) and $+0.86$ (H2GCN). SDA couples density to h (Figure 1b, $\sim 50\times$ drop across the sweep), and density carries strong negative correlations (-0.79, -0.83), while clustering is non-monotonic and correlates weakly ($+0.32$, $+0.19$).

The H2GCN–GCN gap stays in $+0.04$ to $+0.12$ throughout, with no monotonic narrowing as h grows. This breaks with the multi-class pattern of Zhu et al., 2020, where GCN catches up to H2GCN above $h \approx 0.5$: the rise-with-h half of our hypothesis holds, but the gap-narrowing half does not. The real-world overlay (Figure 1a) is directionally consistent: DBLP at $h = 0.76$ sits near the top with both models above 0.85 AP; the low-h datasets cluster at the low end. Real-world H2GCN–GCN gaps stay within ± 0.10, matching the synthetic band.

5.2 Random-Edge Addition with Matched-h Control

Starting from a base SDA graph at $h \approx 0.6$, we added random edges equal to $\{0, 10, 25, 50, 100\}\%$ of $|E|$ and built clean SDA graphs at each realised h as a matched control. The added edges are sampled uniformly at random from node pairs not already connected. Features and labels are fixed throughout.

Random edge addition drives realised h from 0.618 to 0.379 and roughly doubles density. GCN macro AP collapses $0.573 \rightarrow 0.285$ (-50%) and H2GCN drops only $0.589 \rightarrow 0.507$

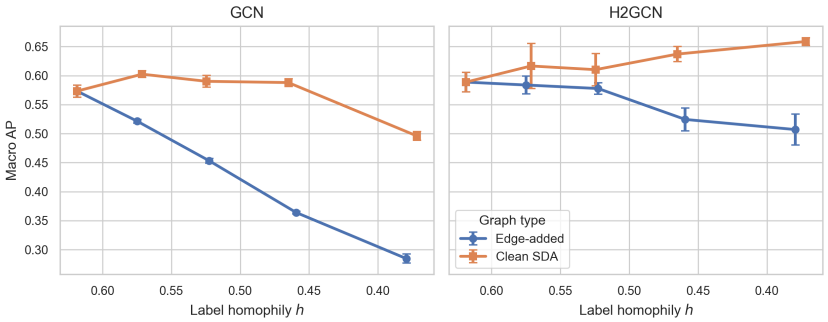


Figure 2: Macro AP versus measured h for edge-added graph and matched-h clean SDA graphs, GCN (left) and H2GCN (right). At matched h, clean SDA outperforms edge-added by a wide margin, especially for GCN.

(-14%). The H2GCN–GCN gap widens monotonically across the five levels ($+0.015 \rightarrow +0.222$).

The matched-h control (Figure 2) shows that scalar h alone does not predict performance: each edge-added graphs performs worse than the clean SDA graph at matching h. The hypothesis holds: GCN collapses faster under random noise, its 1-hop mean aggregation mixes the noisy neighbours directly into each node's representation, while H2GCN's ego/neighbour separation keeps the node's own signal intact even when the immediate neighbourhood is polluted, and the cross-class neighbourhood structure matters, not just scalar h. Real-world: BlogCat and OGB-Proteins (low h, both models low AP) match the noise regime, while Yelp (low h, H2GCN 0.226 vs GCN 0.131) matches the prediction that H2GCN is less damaged when 1-hop aggregation is unreliable.

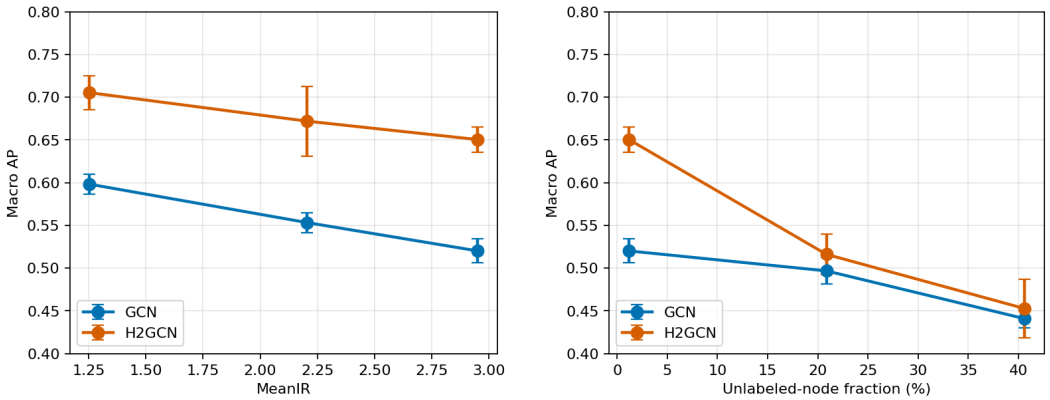
5.3 Label Imbalance and Unlabeled Nodes

We trained GCN and H2GCN on five SDA graphs at $h \approx 0.4$ that vary two label-side axes: per-label frequency skew (measured by MeanIR, defined in subsection 2.3) and the fraction of unlabeled nodes. Conditions 1–3 sweep MeanIR from 1.25 to 2.95 at $\approx 0\%$ unlabeled, and conditions 3–5 fix MeanIR ≈ 3 and vary unlabeled fraction from 0 to 40%.

Table 2: Imbalance and unlabeled-fraction sweep at $h \approx 0.4$. Mean over 3 seeds.

Condition	MeanIR	unlabeled nodes	GCN		H2GCN	
			macro AP	micro AP	macro AP	micro AP
balanced	1.25	1.6%	0.598	0.590	0.705	0.693
mild skew	2.21	1.3%	0.553	0.614	0.672	0.732
strong skew	2.95	1.2%	0.520	0.611	0.650	0.755
strong skew, 20% unlab	2.93	20.9%	0.497	0.557	0.516	0.603
strong skew, 40% unlab	3.00	40.6%	0.441	0.499	0.453	0.524

The skew axis (Table 2, rows 1–3 and Figure 3a) demonstrates the rare label signature collapse: as MeanIR grows, macro AP drops for both models while micro AP stays flat or



(a) Macro AP versus MeanIR. H2GCN keeps its ~ 0.10 advantage across the skew range.

(b) Macro AP versus unlabeled-node fraction at fixed MeanIR ≈ 3. H2GCN drops twice as fast as GCN; the architectural advantage essentially disappears by 40%.

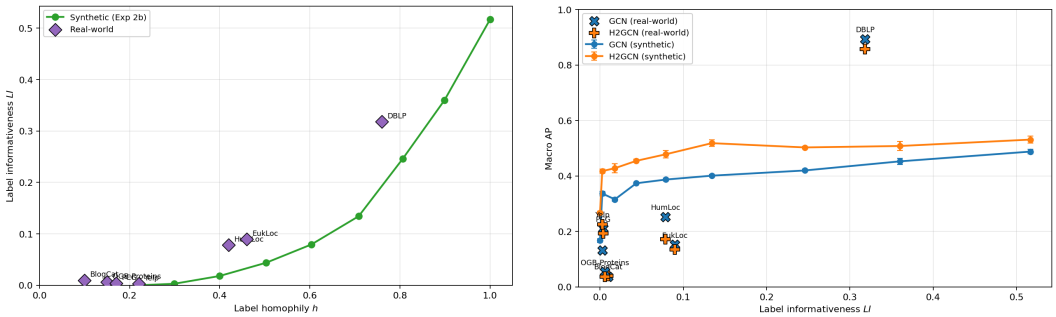
Figure 3: Label imbalance and unlabeled supervision at $h \approx 0.4$: (a) AP versus MeanIR, (b) AP versus unlabeled-node fraction.

rises. Macro penalises failure on rare labels equally, while micro is dominated by common labels that both models still learn well, so the gap between the two metrics widens with skew and indicates how much rare-label performance is being lost. The H2GCN–GCN gap holds and slightly widens across the skew range, suggesting that H2GCN's ego representation preserves rare-label signal that GCN's neighbourhood averaging dilutes.

The unlabeled axis (Table 2, rows 3–5 and Figure 3b) is the more interesting finding: H2GCN's macro-AP advantage *collapses entirely* by 40% unlabeled, dropping roughly twice as fast as GCN. The likely mechanism is parameter count: H2GCN's ego/1-hop/2-hop concatenation has $\sim 4\times$ more parameters than GCN, so fewer labelled nodes leaves the richer architecture undertrained. The real-world equivalent is similar: OGB-Proteins (40.3% unlabeled) is the one dataset where H2GCN underperforms GCN on macro AP, and HumLoc / EukLoc (MeanIR 15 and 45, far above our skew ceiling) show H2GCN behind GCN.

5.4 Label Informativeness as a Predictor

We evaluate label informativeness (LI), defined in subsection 2.3, as a predictor by reading it off the nine synthetic graphs from the homophily sweep (subsection 5.1) and the seven real-world datasets, and comparing how well h and LI track AP on each pool.



(a) h vs LI on the nine synthetic homophily-sweep graphs (line) and the seven real-world datasets (markers). DBLP, HumLoc, and EukLoc sit above the synthetic trend, while the low-h real-world points lie close to it.

(b) Macro AP vs LI on the same pool: synthetic sweep (lines) and real-world datasets (markers).

Figure 4: Label informativeness on the homophily-sweep graphs and real-world datasets: (a) h vs LI, (b) AP vs LI.

Figure 4a shows the coupling between h and LI: across the synthetic sweep they correlate at Pearson +0.67, but the real-world points do not fall on the synthetic line. DBLP, HumLoc, and EukLoc sit clearly above it, meaning that at matched h the real-world neighbourhoods carry more label information than the SDA-generated ones. That offset is what makes LI worth evaluating separately from h even though the two are correlated.

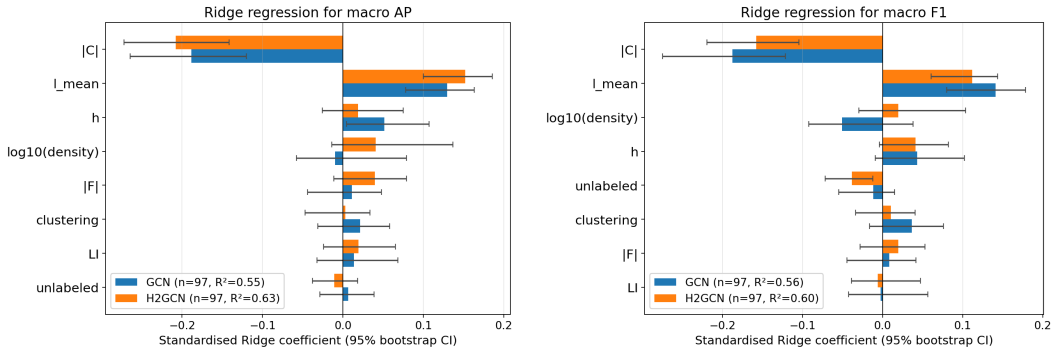
On the nine synthetic graphs, h is the stronger predictor of AP, with Pearson $\text{corr}(h, \text{AP}) = +0.90$ vs $\text{corr}(\text{LI}, \text{AP}) = +0.78$ for GCN and +0.86 vs +0.66 for H2GCN. This is expected because h is the controlled axis of the sweep, so its variance is artificially clean while the variance of LI has irregular intervals. On the seven real-world datasets the ranking flips: $\text{corr}(\text{LI}, \text{AP}) = +0.96$ vs $\text{corr}(h, \text{AP}) = +0.92$ for GCN, and +0.93 vs +0.87 for H2GCN. The flip replicates the multi-class finding of Platonov et al., 2024 in the multi-label setting on a small sample, suggesting that in some cases LI captures neighbourhood structure beyond what scalar h summarises once the controlled-axis advantage is removed.

The pooled Ridge regression in the next subsection is the more definitive test: it weighs LI against h and the other properties jointly across all 97 graphs.

5.5 Pooled Ridge Regression Across All Graphs

To absorb the multi-axes correlation that single-axis selection cannot remove, we fit a Ridge regression on every trained graph in this paper. This is methodologically similar to the empirical performance models of Hutter et al., 2013, who fit regression models (including Ridge regression) on vectors of instance features to predict combinatorial-solver runtime; we apply the same shape to GNNs, with graph properties as features and macro AP and macro F_1 as targets: the synthetic sweeps for each experiment, the seven real-world datasets, and an additional 28 *gap-filler* graphs generated outside the sweeps to cover corners of the property space the controlled axes leave undersampled (mid-$|C|$ at low h, higher $|F|$, higher $\bar{\ell}$, and extra unlabeled-fraction points), for a total of 97 graphs. For each model we predict macro AP and macro F_1 from the standardised vector (h, LI, $\bar{\ell}$, $|C|$, $\log_{10}$ density, $|F|$, unlabeled nodes, clustering). The penalty α is chosen by leave-one-out cross-validation and

pinned for a 1000-iteration bootstrap, from which we report the 95% confidence intervals. With only 97 graphs and the goal of learning from the coefficients how each property drives model performance rather than building a performance predictor, we use no held-out test or validation set, each bootstrap resample is fit on full-size data. Real-world AP and macro F_1 come from Zhao et al., 2023 Tables 3 and 9.



(a) Macro AP ($n = 97$).

(b) Macro F_1 ($n = 97$).

Figure 5: Standardised Ridge coefficients with 95% bootstrap confidence intervals for GCN (blue) and H2GCN (orange) on (a) macro AP and (b) macro F_1.

$|C|$ and $\bar{\ell}$ dominate every cell of Figure 5. More labels make per-label prediction harder, and more labels per node compensate. The magnitudes agree within 30% across AP and F_1, so neither is an artefact of AP's averaging. Two other coefficients are significant. First, h is a weaker predictor of performance for H2GCN than for GCN under both metrics, with the gap larger under macro AP than under macro F_1. Second, the unlabeled fraction harms H2GCN more than GCN, mirroring the label-imbalance experiment. LI, density, clustering, and $|F|$ carry lesser predictive power once the other axes are taken into account.

The h asymmetry is the clearest cross-metric architectural finding. GCN AP rises with h, while H2GCN's dependence on h is weaker, consistent with its ego and 2-hop separation removing the sensitivity to whether neighbours agree on labels. The unlabeled effect matches the label-imbalance finding: it has a stronger negative impact on H2GCN compared to GCN. The LI coefficient closer to zero follows from h and LI correlating at $+0.66$ in the pool, so Ridge most probably spreads their shared variance.

6 Discussion

6.1 Combined picture

Across the five experiments, no single scalar property explains GCN versus H2GCN performance in the multi-label setting. Homophily explains most variance within a controlled sweep. However, the matched-h control graphs of the Random-Edge Addition experiment indicate that the cross-class neighbourhood structure plays a more important role, but it can't be easily interpreted. Another important observation is that label-side properties (imbalance and unlabeled fraction) carry as much weight as graph-side ones, and label informativeness ranks above h as a single predictor on the real-world graphs. The pooled Ridge regression

confirms this joint view: $|C|$, $\bar{\ell}$, and unlabeled fraction all carry independent weight beyond h and LI.

These per-experiment readings have to be interpreted against a recurring obstacle: the SDA generator binds several axes together. Targeting low h raises density and the average degree, targeting many classes caps the achievable h, and label informativeness moves with h. The experiment choices are therefore partly consequences of which axes can or cannot be fixed, while varying others. The pooled Ridge regression is the natural complement: it separates effects through regression rather than selection, which is the appropriate tool when the generator cannot hold the other axes fixed.

Despite those caveats, the per model differences still track the design assumptions of their architectures. GCN's degree-weighted averaging assumes connected nodes share labels. When that assumption fails because of random noise, GCN collapses fastest. H2GCN's ego separation is designed for heterophily and keeps a small but persistent advantage across homophily levels, but its richer parameterisation can be a disadvantage: at higher fractions of unlabeled nodes, its performance worsens faster than GCN.

6.2 Limitations

We compare only two GNNs, GCN and H2GCN, on seven real-world datasets and synthetic graphs of $N = 3000$ nodes. Other architectures could behave differently: GAT (Velickovic et al., 2018) replaces GCN's fixed degree-based weights with learned attention weights, and GraphSAGE (Hamilton et al., 2018) samples a fixed-size neighbour subset and applies a generic aggregator. The real-world set is also small and biology-heavy, with four of the seven datasets covering protein-related tasks, so a broader selection can help tighten any potential correlations.

On the data side, the SDA generator couples homophily with density and the average degree, so our single-axis sweeps cannot fully isolate one property at a time. A generator that can vary homophily, density, clustering, and neighbourhood structure independently and reach the imbalance and the other extremes we observe in the wild, would let future work cleanly separate the effects.

7 Responsible Research

7.1 Code of conduct and Research Integrity

This work follows the TU Delft Code of Conduct[2] and its DIRECT values (Diversity, Integrity, Respect, Engagement, Courage, Trust). In practice, this meant honest reporting of inconclusive experiments alongside the conclusive ones, and openly discussing intermediate results with supervisors and peers within the project group.

7.2 Reproducibility

We release the full code, the random seeds for every dataset and training run, and the model hyperparameters at the repository linked in the abstract. The repository is published under the MIT licence, the most permissive of the common open-source licences, so that anyone can reuse, modify, or redistribute the code without restrictive obligations. Every synthetic graph is fully regenerable from its seed, and the 97 trained graphs with their measured

[2]https://www.tudelft.nl/en/about-tu-delft/strategy/integrity-policy/tu-delft-code-of-conduct

properties are released as a single CSV. Each (graph, model) cell is trained over three random seeds, and reported metrics are seed-level means with standard deviation. After generation, each graph is measured for the properties of interest (homophily, density, clustering, label informativeness, mean label cardinality, unlabeled fraction), and the measured values, not the target parameters, enter the analysis. This matters because the SDA generator cannot always hit the requested target exactly.

7.3 Wider Implications and Limitations

Most of our experiments run on synthetic multi-label graphs generated from random seeds, which carry no personally identifiable information and no inherited social bias. The seven real-world datasets we use as anchors are public benchmarks reused from Zhao et al., 2023 without re-collection, and the work involves no human participants. Multi-label node classification is nonetheless deployed in domains that affect people and resources, such as drug discovery, content tagging, and recommendation, so even the small AP gaps we report can matter once a model is in production. Choosing the right architecture for the dataset can reduce wasted training runs, which is relevant given the energy and water cost of training graph neural networks at scale.

Regarding limitations, the SDA generator couples homophily with density, so the homophily sweeps also vary density, the achievable per-label imbalance (MeanIR up to 3) is narrower than the real-world range (up to 45), and the real-world panel is small (seven datasets) and biology-heavy. We flag these limitations explicitly in the Discussion.

8 Conclusions and Future Work

8.1 Conclusions

We investigate how structural, feature, and label properties of multi-label graphs influence the performance of two opposing graph neural networks, GCN and H2GCN. We answer through five controlled experiments on synthetic multi-label graphs with tunable properties, seven real-world multi-label datasets, and a pooled Ridge regression across all 97 trained graphs that ties the per-experiment results together. The pooled regression is, to our knowledge, the first multi-property predictor of GNN performance on multi-label graphs, and is paired with a multi-label adaptation of label informativeness that generalises the multi-class metric of Platonov et al., 2024 using the per-edge weighting introduced for cross-class neighbourhood similarity by Zhao et al., 2023.

Our main conclusion is that no single property fully predicts which model wins. Homophily is the strongest correlate within the controlled sweeps, and the pooled Ridge regression separately identifies the size of the label space $|C|$ and the mean label cardinality $\bar{\ell}$ as comparably strong predictors. On top of these, the mechanism that produces low homophily, the per-label balance of the data, and the fraction of unlabeled nodes all carry independent weight. On synthetic graphs H2GCN holds a small advantage almost everywhere, while on the seven real-world datasets neither model dominates, with gaps staying within ± 0.1 in mean average precision. In practice, GCN tends to be the safer choice when supervision is sparse or the label space is small. H2GCN tends to win when neighbourhoods are unreliable, but labels are abundant.

8.2 Future Work

Three directions follow from our limitations. First, the comparison should be broadened to attention-based and sampling-based GNNs such as GAT and GraphSAGE, so the patterns we observe can be tested for inductive-bias generality. Second, our synthetic graph generator couples homophily with density; a generator that varies homophily, density, clustering, and neighbourhood structure independently would let future work decouple effects that ours can only entangle. Third, a larger real-world panel from social and citation domains, paired with our own GCN and H2GCN runs on every dataset, would tighten the cross-pool correlations and allow stronger conclusions about generalisation from synthetic to real data.

References

Charte, F., Rivera, A. J., del Jesus, M. J., & Herrera, F. (2015). Addressing imbalance in multilabel classification: Measures and random resampling algorithms [Recent Advancements in Hybrid Artificial Intelligence Systems and its Application to Real-World Problems Progress in Intelligent Systems Mining Humanistic Data]. *Neurocomputing*, *163*, 3–16. https://doi.org/https://doi.org/10.1016/j.neucom.2014.08.091

Hamilton, W. L., Ying, R., & Leskovec, J. (2018). Inductive representation learning on large graphs. https://arxiv.org/abs/1706.02216

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., & Leskovec, J. (2021). Open graph benchmark: Datasets for machine learning on graphs. https://arxiv.org/abs/2005.00687

Hutter, F., Xu, L., Hoos, H. H., & Leyton-Brown, K. (2013). Algorithm runtime prediction: Methods evaluation. https://arxiv.org/abs/1211.0906

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. https://arxiv.org/abs/1609.02907

Platonov, O., Kuznedelev, D., Babenko, A., & Prokhorenkova, L. (2024). Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. https://arxiv.org/abs/2209.06177

Tomas, J. T., Spolaor, N., Cherman, E. A., & Monard, M. C. (2014). A framework to generate synthetic multi-label datasets [Proceedings of the XXXIX Latin American Computing Conference (CLEI 2013)]. *Electronic Notes in Theoretical Computer Science*, *302*, 155–176. https://doi.org/https://doi.org/10.1016/j.entcs.2014.01.025

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. https://arxiv.org/abs/1710.10903

Zhao, T., Dong, T. N., Hanjalic, A., & Khosla, M. (2023). Multi-label node classification on graph-structured data. *Transactions on Machine Learning Research*. https://openreview.net/forum?id=EZhkV2BjDP

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., & Koutra, D. (2020). Beyond homophily in graph neural networks: Current limitations and effective designs. https://arxiv.org/abs/2006.11468

A Appendix

This appendix lists, for every experiment in the paper, the properties of each generated graph together with the per-model macro average precision (AP) and macro F_1, reported as mean $\pm$ standard deviation over the three training seeds. For the label-imbalance experiment we also include micro AP.

A.1 Homophily Sweep on the Synthetic1 Base

Table 3: Homophily sweep on the Synthetic1 base: realised properties and per-model performance (mean $\pm$ std, 3 seeds).

h	density	clust.	LI	GCN AP	H2GCN AP	GCN F_1	H2GCN F_1
0.23	0.500	0.50	0.000	$0.167_{\pm 0.006}$	$0.267_{\pm 0.006}$	$0.251_{\pm 0.007}$	$0.312_{\pm 0.007}$
0.30	0.543	0.75	0.003	$0.337_{\pm 0.008}$	$0.417_{\pm 0.008}$	$0.250_{\pm 0.007}$	$0.316_{\pm 0.020}$
0.40	0.145	0.28	0.018	$0.315_{\pm 0.006}$	$0.428_{\pm 0.017}$	$0.255_{\pm 0.006}$	$0.338_{\pm 0.014}$
0.50	0.095	0.36	0.044	$0.374_{\pm 0.003}$	$0.454_{\pm 0.005}$	$0.282_{\pm 0.007}$	$0.375_{\pm 0.017}$
0.60	0.069	0.45	0.079	$0.387_{\pm 0.005}$	$0.478_{\pm 0.014}$	$0.299_{\pm 0.008}$	$0.426_{\pm 0.014}$
0.71	0.053	0.54	0.134	$0.401_{\pm 0.004}$	$0.518_{\pm 0.011}$	$0.316_{\pm 0.008}$	$0.470_{\pm 0.026}$
0.81	0.021	0.42	0.246	$0.420_{\pm 0.005}$	$0.503_{\pm 0.003}$	$0.337_{\pm 0.011}$	$0.469_{\pm 0.016}$
0.90	0.015	0.52	0.360	$0.453_{\pm 0.011}$	$0.508_{\pm 0.016}$	$0.386_{\pm 0.013}$	$0.485_{\pm 0.016}$
1.00	0.011	0.87	0.517	$0.488_{\pm 0.008}$	$0.531_{\pm 0.013}$	$0.408_{\pm 0.012}$	$0.577_{\pm 0.015}$

A.2 Random-edge Addition and Matched-h Control

Table 4: Random-edge addition: realised properties and per-model performance (mean $\pm$ std, 3 seeds).

added	h	density	avg deg	GCN AP	H2GCN AP	GCN F_1	H2GCN F_1
0%	0.618	0.0080	24.1	$0.573_{\pm 0.010}$	$0.589_{\pm 0.017}$	$0.488_{\pm 0.010}$	$0.561_{\pm 0.011}$
10%	0.575	0.0088	26.5	$0.521_{\pm 0.003}$	$0.584_{\pm 0.015}$	$0.421_{\pm 0.005}$	$0.531_{\pm 0.013}$
25%	0.523	0.0100	30.1	$0.453_{\pm 0.004}$	$0.578_{\pm 0.010}$	$0.329_{\pm 0.009}$	$0.518_{\pm 0.009}$
50%	0.459	0.0120	36.1	$0.364_{\pm 0.002}$	$0.524_{\pm 0.020}$	$0.263_{\pm 0.008}$	$0.473_{\pm 0.022}$
100%	0.379	0.0161	48.1	$0.285_{\pm 0.008}$	$0.507_{\pm 0.027}$	$0.229_{\pm 0.005}$	$0.452_{\pm 0.024}$

Table 5: Matched-h clean SDA control: realised properties and per-model performance (mean $\pm$ std, 3 seeds).

h	density	avg deg	GCN AP	H2GCN AP	GCN F_1	H2GCN F_1
0.618	0.0080	24.1	$0.573_{\pm 0.010}$	$0.589_{\pm 0.017}$	$0.488_{\pm 0.010}$	$0.561_{\pm 0.011}$
0.571	0.0190	57.0	$0.603_{\pm 0.006}$	$0.617_{\pm 0.039}$	$0.539_{\pm 0.016}$	$0.584_{\pm 0.036}$
0.524	0.0226	67.8	$0.590_{\pm 0.010}$	$0.610_{\pm 0.028}$	$0.509_{\pm 0.011}$	$0.576_{\pm 0.038}$
0.465	0.0289	86.8	$0.588_{\pm 0.007}$	$0.637_{\pm 0.013}$	$0.484_{\pm 0.007}$	$0.589_{\pm 0.016}$
0.372	0.0439	131.6	$0.496_{\pm 0.008}$	$0.659_{\pm 0.007}$	$0.368_{\pm 0.005}$	$0.579_{\pm 0.013}$

A.3 Label Imbalance and Unlabeled Nodes

Table 6: Imbalance and unlabeled-fraction sweep at $h \approx 0.4$: realised properties and per-model performance (mean ± std, 3 seeds). Micro AP included.

			macro AP		micro AP		macro F_1	
condition	MeanIR	unlab.	GCN	H2GCN	GCN	H2GCN	GCN	H2GCN
balanced	1.25	1.6%	$0.598_{\pm 0.012}$	$0.705_{\pm 0.020}$	$0.590_{\pm 0.010}$	$0.693_{\pm 0.022}$	$0.615_{\pm 0.011}$	$0.688_{\pm 0.023}$
mild skew	2.21	1.3%	$0.553_{\pm 0.012}$	$0.672_{\pm 0.041}$	$0.614_{\pm 0.012}$	$0.732_{\pm 0.044}$	$0.499_{\pm 0.015}$	$0.638_{\pm 0.056}$
strong skew	2.95	1.2%	$0.520_{\pm 0.014}$	$0.650_{\pm 0.015}$	$0.611_{\pm 0.021}$	$0.755_{\pm 0.012}$	$0.418_{\pm 0.015}$	$0.626_{\pm 0.038}$
strong, 20% unlab.	2.93	20.9%	$0.497_{\pm 0.016}$	$0.516_{\pm 0.024}$	$0.557_{\pm 0.009}$	$0.603_{\pm 0.019}$	$0.238_{\pm 0.013}$	$0.281_{\pm 0.016}$
strong, 40% unab.	3.00	40.6%	$0.441_{\pm 0.011}$	$0.453_{\pm 0.034}$	$0.499_{\pm 0.017}$	$0.524_{\pm 0.041}$	$0.131_{\pm 0.007}$	$0.144_{\pm 0.014}$

A.4 Pooled Ridge regression coefficients

Tables 7 and 8 report the standardised Ridge coefficients with 95% bootstrap confidence intervals (1000 iterations) that underlie Figure 5. The penalty α is selected per model and per target via leave-one-out cross-validation on a logarithmic grid (10^{-3} to 10^{3}, 25 points) and pinned for the bootstrap.

Table 7: Pooled Ridge coefficients for macro AP ($n = 97$). 95% bootstrap CIs in brackets.

predictor	GCN	H2GCN
h	0.051 [0.004, 0.107]	0.019 [−0.025, 0.075]
LI	0.014 [−0.032, 0.068]	0.019 [−0.024, 0.065]
$\bar{\ell}$	0.129 [0.078, 0.163]	0.152 [0.100, 0.186]
$\lvert C\rvert$	−0.188 [−0.264, −0.120]	−0.208 [−0.272, −0.142]
$\log_{10}$ density	−0.010 [−0.058, 0.079]	0.041 [−0.014, 0.137]
$\lvert F\rvert$	0.011 [−0.044, 0.048]	0.040 [−0.011, 0.079]
unlabeled	0.006 [−0.028, 0.039]	−0.011 [−0.038, 0.018]
clustering	0.022 [−0.031, 0.058]	0.003 [−0.047, 0.034]

Table 8: Pooled Ridge coefficients for macro F_1 ($n = 97$). 95% bootstrap CIs in brackets.

predictor	GCN	H2GCN
h	0.043 [−0.009, 0.102]	0.041 [−0.004, 0.082]
LI	−0.002 [−0.043, 0.056]	−0.006 [−0.039, 0.047]
$\bar{\ell}$	0.141 [0.080, 0.178]	0.112 [0.061, 0.143]
$\lvert C\rvert$	−0.188 [−0.275, −0.121]	−0.158 [−0.219, −0.105]
$\log_{10}$ density	−0.051 [−0.092, 0.038]	0.020 [−0.030, 0.104]
$\lvert F\rvert$	0.009 [−0.044, 0.042]	0.020 [−0.028, 0.053]
unlabeled	−0.012 [−0.055, 0.015]	−0.038 [−0.072, −0.012]
clustering	0.037 [−0.016, 0.076]	0.010 [−0.034, 0.041]