Heterophilic Methods on Multi-Label Graphs

How Do Methods Designed for Heterophilic Graphs Compare for Multi-Label Node Classification?

Cristian Turcan[1]
Responsible Professor: Megha Khosla[1] **Supervisor: 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
16 June 2026

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

Abstract

Graph neural networks for node classification usually assume homophily (that connected nodes share labels), and a promising family of methods has been developed for heterophilic graphs, where neighbours' labels instead tend to differ. These methods are almost always evaluated on multi-class datasets, in which each node has exactly one label. However, many real problems are instead multi-label, with each node carrying a set of labels. An intuitive question is then whether methods designed for heterophilic graphs remain effective for multi-label node classification. To find out, we run a comparison of two simple baselines against six heterophily-oriented GNNs across several real multi-label graphs and a multi-class control, complemented by two synthetic experiments, one that varies homophily directly and one that varies the number of labels per node, and a supplementary check that collapses the multi-label structure of real-world datasets into a multi-class one (which we report for completeness but treat cautiously, since the collapse changes the datasets too much for the two versions to count as the same problem). We find that performance appears to track how homophilous a graph is at least as much as how sophisticated a model is: when homophily is low, the heterophilic models rarely beat a plain feature-only baseline or a simple structure-only embedding, and the advantage of message passing tends to return only as homophily rises. The gains these methods report on multi-class benchmarks may therefore not transfer automatically to the low-homophily multi-label regime.

1 Introduction

Node classification is one of the most important tasks on graph data: given a graph whose nodes carry features, the goal is to assign each node a label using both its own features and the structure around it. It has many applications from categorising papers in citation networks and users in social networks to annotating proteins in biological interaction graphs. The dominant approach is the graph neural network (GNN), which predicts a node's label by iteratively aggregating information along its edges. Most GNNs, including GCN [5] and GraphSAGE [4], work on an implicit assumption of *homophily*: that a node's neighbours tend to share its label, so that smoothing (low-pass) aggregation reinforces the correct signal.

Unfortunately, many real graphs violate this assumption. Under *heterophily*, connected nodes often carry different labels, and simple smoothing then blends a node's own learning signal with unrelated neighbours and makes it less informative. A substantial family of heterophily-oriented methods has emerged in response [14], among them H2GCN [15], FAGCN [2], GPR-GNN [3], ACM-GCN [7], Ordered GNN [12] and LINKX [6]. These models try to learn when neighbours disagree through strategies such as signed message passing, learnable spectral filters, non-local neighbourhood expansion.

Almost all of these methods, however, are validated on *multi-class* benchmarks, where each node has exactly one label. Yet many of the node-classification tasks are *multi-label*: a protein can have several biological functions, and a social-network user can hold multiple interests at once. In multi-label graphs, however, the notion of homophily becomes harder to define. When nodes have sets of labels, a node is not necessary simlar to another if only one class is shared, but rather depends on the overlap between their label sets [13]. Whether the approaches that help heterophily-oriented GNNs on multi-class graphs carry over to this setting is unknown.

This motivates the central question of this project:

> *How do different methods designed for heterophilic graph datasets compare for multi-label node-classification datasets?*

To answer it, we run an empirical comparison of 2 baselines: a feature-only MLP and the structure-only embedding DeepWalk [10], and six heterophily-oriented GNNs across five real multi-label graphs and a multi-class heterophily control (Roman-Empire), as well as on two controlled synthetic experiments: one that varies homophily directly and one that varies the number of labels per node. We also include a supplementary binarization check that collapses each multi-label graph to a multi-class problem on the identical graph; we report it for completeness but, as we discuss, collapsing the labels changes the datasets so substantially that the binarized graphs cannot really be regarded as the same datasets, so we keep it out of our main claims. Every model passes through a single identical pipeline, the same data loading, preprocessing, splits, training budget and evaluation metrics, so that any difference in score is attributable to the model rather than to the experimental environment.

In short, we find that the model performance tends to track label homophily. On the lower-homophily multi-label graphs the heterophily-specialised architectures rarely beat the plain MLP or the structure-only DeepWalk embedding, on the high-homophily multi-label graph (DBLP) and on the multi-class control the expected message-passing advantage largely returns, with H2GCN strongest. A synthetic homophily sweep is consistent with this under more controlled conditions: every heterophily GNN's performance rises with homophily, and none gains an advantage as the graph grows more heterophilous within the range we tested. The same conclusion survives when we instead vary the number of labels per node: the disadvantage remains across cardinalities. A supplementary binarization check is at most consistent with this, collapsing to multi-class prediction helps only where it incidentally raises homophily, but we lean on it lightly, since the collapse alters the datasets too much to count as a clean control. Taken together, the advantages reported for these methods on multi-class benchmarks may not transfer automatically to the low-homophily multi-label datasets.

Contributions and relation to prior work. Two recent works motivate this study, and our contribution is best understood as the gap between them. The survey of Zheng et al. [14] organises the heterophily literature into the three families we adopt to group the models we evaluate (Section 2.4), but, like the methods it catalogues, it evaluates them almost entirely on multi-class benchmarks. Independently, Zhao et al. [13] introduced a benchmark for multi-label node classification (including the multi-label graphs we use and synthetic generator). Neither study closes the gap we target: the survey never runs its heterophily-specialised models on multi-label graphs, while Zhao et al. evaluate only classical GNNs and a *single* heterophily model (H2GCN). Our contribution is to test the methods while assessing if the performance remains in a multi-label setting as we test across several real and synthetic graphs, and to interpret the results in terms of the approaches that distinguish the heterophily-oriented models from each other and from the baselines.

2 Background

2.1 Node classification

We work with a single graph $G = (\mathcal{V}, \mathcal{E})$ of $n = |\mathcal{V}|$ nodes, described by an adjacency matrix $A \in \{0,1\}^{n \times n}$ and a node-feature matrix $X \in \mathbb{R}^{n \times d}$ whose i-th row x_i is the feature vector of node i. Each node is associated with labels drawn from C classes. In the *multi-class* case the label of node i is one class $y_i \in \{1, \ldots, C\}$. In the *multi-label* case node i is assigned a set of classes, encoded as a binary vector $y_i \in \{0,1\}^C$ in which several entries may equal

one at once. The task is: labels are observed on a training subset of nodes and must be predicted for the held-out nodes, with the full graph structure and all node features available throughout. Predicted label scores are scored with the threshold-free, class-imbalance-aware metrics described in Section 3.5.

2.2 Graph neural networks

Modern node classifiers are predominantly *message-passing* graph neural networks. Starting from $h_i^{(0)} = x_i$, each layer updates a node by aggregating its neighbours' representations and combining the result with its own:

$$h_i^{(l+1)} = \mathrm{UPDATE}\Big(h_i^{(l)},\ \mathrm{AGGREGATE}\big(\{\, h_j^{(l)} : j \in \mathcal{N}(i)\,\}\big)\Big), \tag{1}$$

where $\mathcal{N}(i)$ is the set of neighbours of node i. The canonical instance, GCN [5], aggregates with a symmetrically normalised, self-looped adjacency $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\,\tilde{D}^{-1/2}$ (with $\tilde{A} = A + I$ and $\tilde{D}$ its degree matrix), giving the layer

$$H^{(l+1)} = \sigma\big(\hat{A}\,H^{(l)}W^{(l)}\big), \tag{2}$$

for a learnable weight matrix $W^{(l)}$ and a non-linearity σ. Stacking such layers repeatedly smooths each node towards the average of its neighbourhood, acting as a low-pass filter on the graph signal, which is beneficial particularly when neighbours tend to share labels.

2.3 Homophily and heterophily

How well neighbour smoothing serves a task depends on how often an edge joins nodes of the same label. In the multi-class case this is captured by the *edge homophily ratio*

$$h = \frac{\big|\{(i,j) \in \mathcal{E} : y_i = y_j\}\big|}{|\mathcal{E}|}, \tag{3}$$

the fraction of edges connecting same-label nodes. A graph is *homophilic* when h is high (neighbours mostly agree, and a GCN's low-pass smoothing reinforces the correct signal) and *heterophilic* when h is low (neighbours mostly disagree, so smoothing blends a node's signal with unrelated ones and degrades it). The heterophily-oriented methods we study (Section 2.4) are built for the low-h regime, replacing pure smoothing with signed, high-pass, or non-local aggregation. When each node carries a *set* of labels, however, even this scalar notion of agreement is no longer well defined: two neighbours may share some labels and not others, which is one of several ways the multi-label setting departs from the standard.

2.4 GNNs for heterophilic graphs

We group heterophily methods by the definitions of Zheng et al. [14], which separates methods that change *who* is in the neighbourhood from methods that change *how* messages are combined, with hybrids doing both. We use the same families for the models we evaluate (Section 3.1) and sketch each approach here, since these strategies are what we call to when interpreting the results.

Non-local extension (change *who*). These methods keep standard aggregation but alter which nodes effectively count as neighbours. *High-order mixing* aggregates from k-hop neighbours on the original graph, on the grounds that even when the 1-hop neighbourhood disagrees, 2- or 3-hop neighbours often share the ego node's label (e.g. MixHop [1]), *potential-neighbour discovery* instead re-wires the graph, adding edges to nodes similar in feature space but structurally distant (e.g. Geom-GCN [9]). The model we evaluate from this family, *Ordered GNN* [12], orders message passing by hop distance, writing nearer hops into earlier, non-overlapping blocks of the hidden vector, keeping information at different distances in separate feature dimensions rather than averaging it together.

Architecture refinement (change *how*). These methods keep the neighbour set fixed but change how messages are combined. *FAGCN* [2] learns a signed coefficient $\pm\alpha$ on each edge, so it can add similar neighbours (low-pass smoothing) and *subtract* dissimilar ones (high-pass sharpening) rather than only averaging. *ACM-GCN* [7] generalises this: at every layer it computes a low-pass, a high-pass and an identity (self) channel and learns per-node weights that select among them. *GPR-GNN* [3] propagates features for several hops and learns a signed weight on each step (a Generalised PageRank), letting it emphasise the most useful parts and reduce the impact of distant contributions. All three send a node's own features through neighbour aggregation, the property our results turn out to be sensitive to.

Hybrid methods (combine both). *H2GCN* [15] applies three heterophily design rules at once: it separates a node's own (ego) representation from its neighbour aggregate instead of mixing them, aggregates over higher-order hops, and concatenates the output of every layer. *LINKX* [6] keeps features and structure on separate paths, training one MLP on the node features and a second on each node's row of the adjacency matrix and combining the two only at the end.

A feature-structure spectrum. Two non-GNN baselines bracket these designs: an *MLP* that uses node features only and ignores the graph, and *DeepWalk* [10], which embeds nodes from random walks and so uses structure only. Ordering the models by how tightly they bind features to structure gives a spectrum: at one end the MLP and DeepWalk use a single source and so cannot let one corrupt the other, LINKX uses both but on separate paths, H2GCN keeps the ego features intact while still aggregating, Ordered GNN and FAGCN mix features across neighbours but with some protection (ordered hop blocks, signed edges), and at the other end ACM-GCN and GPR-GNN send a node's own features through unguarded neighbour aggregation. Under low homophily, where neighbours share few labels, models toward the entangling end have the most to lose, because aggregation blends a node's informative features with poorly-related neighbours before the classifier sees them. We return to this spectrum when interpreting the results (Section 7).

2.5 The multi-label setting

Multi-label node classification is not simply multi-class with extra target columns. Following the analysis of Zhao et al. [13], three properties set it apart.

1. **Homophily is not naturally defined.** Standard edge homophily is built around a single label per node [15]. Extending it to label sets is non-trivial: Jaccard overlap [13] and per-label agreement yield genuinely different measures of how "homophilic" an edge is.

2. **Label correlations may absorb the signal.** Co-occurring labels can already encode much of the structure that high-pass and signed filters are designed to recover, reducing the heterophily advantage one would expect.

3. **Labels are not mutually exclusive.** In multi-class classification, predicting "cat" usually implies "not dog." In multi-label, predicting one label does not rule out another.

3 Methodology

Our methodology has three parts: the set of models we compare (Section 3.1), the benchmark datasets (Section 3.2), and a single shared experimental pipeline applied identically to every model (Section 3.5). The families used to group the models follow the standard heterophily grouping of Zheng et al. [14], whose details we describe in the background (Section 2.4).

3.1 Models compared

We compare eight models grouped by family (Table 1): two baselines and six heterophily-oriented GNNs spanning architecture refinement, non-local extension, and hybrid designs. The six heterophily-oriented models: FAGCN [2], ACM-GCN [7], GPR-GNN [3], Ordered GNN [12], H2GCN [15] and LINKX [6], are drawn from the taxonomy of Zheng et al. [14]. Only H2GCN was previously evaluated on multi-label graphs by Zhao et al. [13]. The two baselines are a feature-only MLP and the random-walk embedding DeepWalk [10].

Table 1: The eight models under comparison, grouped by family.

Family	**Model**	**Key idea**
Baselines	MLP	Feature-only, ignores the graph
	DeepWalk [10]	Random-walk node embeddings
Architecture refinement	FAGCN [2]	Learns $\pm\alpha$ per edge (low/high-pass)
	ACM-GCN [7]	Low-pass, high-pass and identity channels
	GPR-GNN [3]	Learns per-layer combination weights
Non-local extension	Ordered GNN [12]	Orders message passing by hop into hidden blocks
Hybrid	H2GCN [15]	Ego/neighbour split + high-order mixing + combine
	LINKX [6]	MLP on node features + adjacency rows

3.2 Datasets

We evaluate on six node-classification benchmarks: five real multi-label graphs and one multi-class graph (Roman-Empire) included as a heterophily control (Table 2). The real multi-label graphs are those released by Zhao et al. [13], EukaryoteGO, HumanGO and BlogCatalog correspond to their EukLoc, HumLoc and BlogCat datasets respectively, and PCG and DBLP keep their original names. Roman-Empire is the multi-class heterophily benchmark introduced by Platonov et al. [11].

We deliberately reuse the multi-label graphs of Zhao et al. [13] rather than gathering new ones, for three reasons. First, they are the only published multi-label node-classification benchmarks with a documented notion of homophily we could find, so reusing them keeps

our results directly comparable to the single prior study that evaluated GNNs in this setting. Second, and most important for our question, they span almost the entire homophily axis, from strongly heterophilous (BlogCatalog, $h = 0.10$), through intermediate (PCG, HumanGO, EukaryoteGO), to strongly homophilous (DBLP, $h = 0.76$), which is close to the kind of variation needed to test whether performance tracks homophily. Third, they differ widely in feature richness: BlogCatalog has no native node features (forcing models to rely on structure alone), the Gene-Ontology graphs carry low-dimensional sequence embeddings, and DBLP has rich 300-dimensional features. Because we read each model along a feature/structure axis, this spread lets us watch how a method behaves as the balance between the two shifts.

Roman-Empire [11] is included as a *multi-class control*: it is one of the standard multi-class heterophily benchmarks these methods were designed and tuned for, and it acts as a positive control on our pipeline. If the heterophily models rank there as the literature reports, we can attribute any departure from that ranking on the multi-label graphs to the multi-label setting itself, rather than to a faulty implementation.

Table 2: Benchmark datasets. "Labels" is the number of distinct classes, for multi-label graphs each node may hold several of them. h is the Jaccard label homophily reported by Zhao et al. [13] for the multi-label graphs, for Roman-Empire it is the multi-class edge homophily reported by Platonov et al. [11].

Dataset	**Domain**	**Nodes**	**Labels**	h	**Task**
EukaryoteGO [13]	Gene Ontology	7 766	22	0.46	Multi-label
HumanGO [13]	Gene Ontology	3 106	14	0.42	Multi-label
BlogCatalog [13]	Social network	10 312	39	0.10	Multi-label
PCG [13]	Protein graph	3 233	15	0.17	Multi-label
DBLP [13]	Co-authorship	28 000	4	0.76	Multi-label
Roman-Empire [11]	Word graph	22 662	18	0.05	Multi-class

3.3 Controlled synthetic sweeps

The real benchmarks confound homophily with size, domain and feature richness, so to isolate one property at a time we also generate synthetic multi-label graphs in which a single target property is varied while the others are held as fixed as the generator allows. Two sweeps share the same generator. The *homophily sweep* (Section 4.1) fixes the labels per node and varies the target homophily h; the *cardinality sweep* (Section 4.2) does the reverse, fixing homophily and varying the number of labels per node k. Both run the same model suite as the real datasets through the identical pipeline below, as a multi-label task with no label binarization and each model using its full upstream defaults (Section 3.1). The two generators differ in one important respect: the cardinality-sweep generator calibrates each graph to both its target homophily and a fixed mean degree of 10, whereas the homophily-sweep generator targets h only and lets the mean degree co-vary with it. Mean degree is therefore a partial implication in the homophily sweep (Section 6).

Homophily sweep design. We generate one graph at each target homophily $h \in \{0.2, 0.3, \ldots, 1.0\}$, all on $N = 3000$ nodes with a label space of $L = 20$ classes (mean cardinality ≈ 3.2 labels per node) and 20-dimensional features, holding those quantities fixed while only h changes.

Because the generator targets h alone and leaves the mean degree free, density falls steeply as the graph becomes more homophilous, from a mean degree of ≈ 1499 at $h = 0.2$ to ≈ 32 at $h = 1.0$; this is the degree confound we flag in Section 6 and remove in the cardinality sweep. Each graph is loaded with a fresh 60/20/20 random split, and every model is trained on it over three seeds $\{42, 43, 44\}$ on that single split. The headline analysis (Section 4.1) reads the heterophilous-to-homophilous part of this grid, $h = 0.2$ to 0.7.

Cardinality sweep design. We sweep $k \in \{2, 4, 6, 8, 10, 12\}$ labels per node over a label space of $L = 60$ classes, on graphs of $N = 3000$ nodes with target mean degree 10 and 64-dimensional features, generating one graph per k (graph seed 0) and training every model on it over three seeds $\{42, 43, 44\}$ on a single split. Homophily is fixed at $h = 0.20$ for every cell.

3.4 Multi-class binarization of multi-label datasets

To test how much the multi-label structure itself impacts the results, we re-run the real datasets as a multi-class task by giving each node a single label, leaving the graph unchanged. The change is made at load time and touches only the labels y, the adjacency A and features X stay the same, so any change in score comes from the task and not from the graph. Each node keeps the most common of its labels, where most common means the label carried by the most nodes across the dataset (ties broken by lowest index). Labels that win for at least one node become the new classes, and nodes with no label are dropped from every split rather than given a fake class. Collapsing each node to one label usually raises homophily, since neighbours now only need to match on a single class instead of overlapping label sets, and this is the effect we point to when reading the change in performance. Picking the most common label is just one option, it helps popular labels, and a rarest-label or random-label rule would behave differently. Every model is run on both versions under the same pipeline and metrics, so the multi-label and multi-class scores can be compared side by side. Section 4.3 reports the result and discusses how far the two are really comparable. We stress up front that this is a limited supplementary check rather than a controlled experiment: collapsing every node to a single label discards most of the label structure that defines these as multi-label problems and, as we show, incidentally shifts the graph's effective homophily, so the binarized version is better thought of as a *different* task on the same wiring than as the same dataset re-scored. We report it for completeness and transparency, but read nothing about the models' standing on the multi-label task into it.

3.5 Experimental pipeline

Every model is run through an identical five-stage pipeline so that score differences reflect the model and not the pre-processing or random seed.

1. **Data.** Load adjacency, node features and (multi-label) targets.
2. **Preprocess.** Symmetric normalisation, self-loop handling, feature scaling.
3. **Split.** A stratified train/validation/test split, with the same indices reused for every model.
4. **Train/eval.** A fixed training budget per model, with early stopping on the validation set.

5. **Metrics.** Evaluation on the held-out test set with the metrics defined in Section 3.6.

3.6 Evaluation metrics

We summarise predictive quality with four metrics.

- **F1-micro** pools true positives, false positives and false negatives across all labels and nodes before computing a single F1 score. It is dominated by the frequent labels and reflects overall predictive accuracy.
- **F1-macro** computes F1 separately for each class and averages the per-class values without weighting, so a rare class counts as much as a common one. The gap between F1-micro and F1-macro therefore indicates how well a model handles minority labels rather than just the frequent ones.
- **ROC-AUC** per label it estimates the probability that a randomly chosen positive node is ranked above a randomly chosen negative one, averaged over labels.
- **Average precision (AP)** is the area under the precision-recall curvem it rewards correctly ranking the present labels and is far less forgiving for class imbalance. For this reason, and following Zhao et al. [13], we treat AP as the primary metric on the multi-label graphs and read the others as supporting evidence.

Reporting all four predictive metrics matters because they can disagree: a model may post a strong F1-micro by getting the common labels right while failing the rare ones (low F1-macro). Where the metrics disagree, that is itself part of the finding.

4 Experiments and Results

Tables 3-8 report test-set means for the five real multi-label benchmarks and the multi-class control, means are over the splits described in Section 3.5. Best value per metric is in **bold**. Throughout, it is useful to read the models along a feature-structure axis: the MLP uses node features only, DeepWalk uses structure only, LINKX keeps the two *separate* (an MLP on features and an MLP on adjacency rows) and combines them late, while the remaining GNNs *entangle* the two by passing features through neighbour aggregation. How each dataset rewards or punishes that merging is the recurring story.

Table 3: Results on EukaryoteGO (multi-label, $h = 0.46$).

Family	Model	F1-micro	F1-macro	ROC-AUC	AP
Baselines	MLP	0.450	0.110	0.698	0.138
	DeepWalk	0.347	0.099	0.656	0.108
Arch.	FAGCN	0.430	0.073	0.623	0.115
	ACM-GCN	0.166	0.044	0.504	0.059
	GPR-GNN	0.436	0.093	0.662	0.111
Non-local	Ordered GNN	0.359	0.045	0.543	0.073
Hybrid	H2GCN	**0.470**	**0.124**	0.714	0.142
	LINKX	0.451	0.121	**0.750**	**0.154**

EukaryoteGO. H2GCN posts the best F1-micro (0.470) and F1-macro (0.124), LINKX the best ROC-AUC (0.750) and AP (0.154), and the feature-only MLP sits just behind (0.450 F1-micro). We read this cluster as a sign that the 32-dimensional sequence-embedding features already carry much of the label information and that the graph adds relatively little here. LINKX's small edge on the ranking metrics is broadly in line with what its architecture would suggest, its adjacency-row information adds a little structural signal on top of an essentially feature MLP, though the margins are small enough that we would not read too much into the exact ordering.

Table 4: Results on HumanGO (multi-label, $h = 0.42$).

Family	Model	F1-micro	F1-macro	ROC-AUC	AP
Baselines	MLP	0.451	0.178	0.680	0.189
	DeepWalk	0.486	**0.276**	**0.728**	**0.260**
Arch.	FAGCN	0.454	0.151	0.670	0.197
	ACM-GCN	0.090	0.072	0.515	0.090
	GPR-GNN	0.414	0.130	0.562	0.147
Non-local	Ordered GNN	0.367	0.075	0.562	0.123
Hybrid	H2GCN	0.475	0.185	0.670	0.186
	LINKX	**0.513**	0.229	0.720	0.247

HumanGO. Here both channels appear to carry signal, and the split decision is suggestive of which. LINKX takes F1-micro and ROC-AUC by combining features and structure, while DeepWalk, which sees only structure, wins F1-macro and AP, which we read as the random-walk structure recovering some rare labels that the feature-only view misses. The tight cluster of five models within $\sim$0.07 F1-micro suggests that no single bias is decisive on this graph.

Table 5: Results on BlogCatalog (multi-label, $h = 0.10$).

Family	Model	F1-micro	F1-macro	ROC-AUC	AP
Baselines	MLP	0.171	0.025	0.500	0.036
	DeepWalk	**0.354**	**0.207**	**0.708**	**0.203**
Arch.	FAGCN	0.172	0.026	0.475	0.038
	ACM-GCN	0.036	0.021	0.508	0.040
	GPR-GNN	0.174	0.026	0.476	0.036
Non-local	Ordered GNN	0.189	0.038	0.571	0.077
Hybrid	H2GCN	0.171	0.026	0.496	0.039
	LINKX	0.346	0.153	0.700	0.164

BlogCatalog. BlogCatalog has no node features, so every feature-consuming model is handed only an identity matrix, which makes it perhaps the cleanest separation of structure from features in the setup. DeepWalk, built purely from random walks, wins every metric, while the MLP and the message-passing GNNs that rely on feature aggregation fall to $\sim$0.17 F1-micro and chance-level ROC-AUC, consistent with there being little for a feature learning. Two models stand apart in a way that fits this reading. LINKX is a clear second,

because its adjacency-row MLP encodes structure directly and so still works when the feature are empty.

Table 6: Results on PCG (multi-label, $h = 0.17$).

Family	Model	F1-micro	F1-macro	ROC-AUC	AP
Baselines	MLP	0.380	0.198	0.533	0.163
	DeepWalk	**0.413**	**0.317**	**0.638**	**0.232**
Arch.	FAGCN	0.391	0.200	0.554	0.180
	ACM-GCN	0.238	0.197	0.497	0.137
	GPR-GNN	0.384	0.186	0.446	0.129
Non-local	Ordered GNN	0.388	0.183	0.507	0.138
Hybrid	H2GCN	0.398	0.251	0.597	0.218
	LINKX	0.408	0.270	0.588	0.225

PCG. PCG looks like an in-between case: structure is the single strongest information source (DeepWalk wins all four metrics), while the 32-dimensional features are only weakly predictive. No architecture seems to be a strong match for this graph, so the message-passing models neither gain much from aggregation nor are badly hurt by it, and end up around the same score.

Table 7: Results on DBLP (multi-label, $h = 0.76$).

Family	Model	F1-micro	F1-macro	ROC-AUC	AP
Baselines	MLP	0.802	0.785	0.913	0.820
	DeepWalk	0.671	0.633	0.793	0.671
Arch.	FAGCN	0.823	0.804	0.924	0.844
	ACM-GCN	0.811	0.784	0.858	0.751
	GPR-GNN	0.856	0.841	0.913	0.843
Non-local	Ordered GNN	0.466	0.302	0.782	0.583
Hybrid	H2GCN	**0.910**	**0.902**	**0.962**	**0.919**
	LINKX	0.877	0.867	0.938	0.891

DBLP. DBLP appears to invert the story. At $h = 0.76$ neighbours largely share labels, so aggregation seems to help rather than hurt, and the models whose design assumptions fit this do best.

Roman-Empire. This is the multi-class heterophily benchmark the methods were designed for, and the ranking is broadly consistent with the designs working as intended.

Across datasets. Figure 1 collects the AP column of the five multi-label tables onto a single homophily axis. Read this way the per-dataset observations line up into one trend: average precision for the GNNs is low on the heterophilous graphs and only rises on DBLP, the one strongly homophilous graph. We stress that this is a trend *across* graphs that differ in features, size and domain as well as homophily, not a controlled manipulation of any one axis, the synthetic sweeps below isolate the variables one at a time.

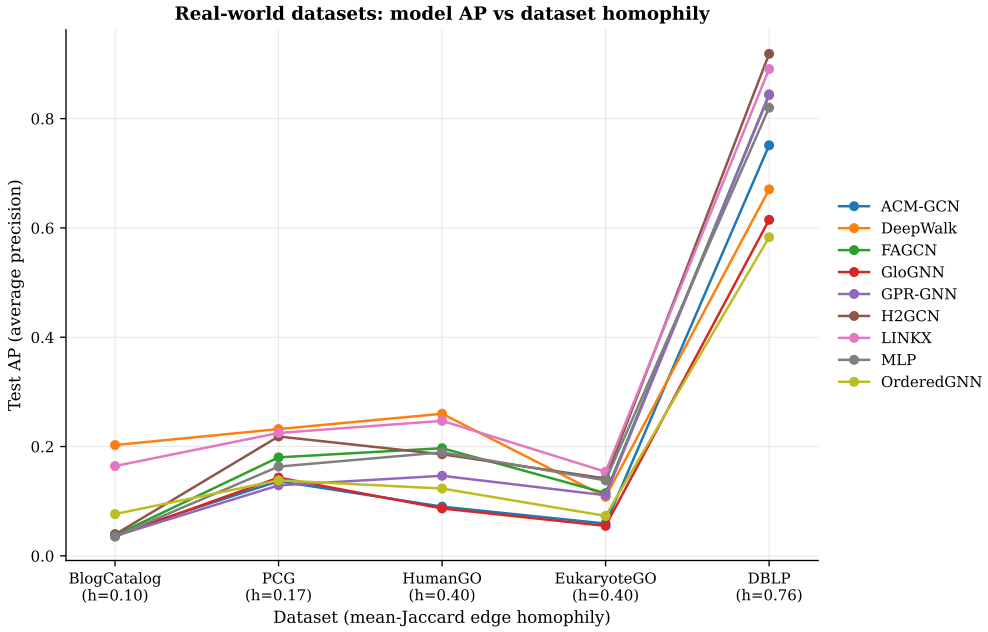

Figure 1: Test average precision for every model on the five real-world multi-label graphs, ordered left-to-right by increasing mean-Jaccard edge homophily. Each line is one model. The cross-dataset pattern mirrors Tables 3-7: the message-passing GNNs are competitive only on the most homophilous graph (DBLP), while toward the heterophilous end the structure-only DeepWalk. Homophily gaps between datasets are not to scale (positions are evenly spaced, the measured value is printed under each tick).

Table 8: Results on Roman-Empire (multi-class control, edge $h = 0.05$).

Family	Model	F1-micro	F1-macro	ROC-AUC	AP
Baselines	MLP	0.661	0.506	0.936	0.545
	DeepWalk	0.119	0.043	0.505	0.058
Arch.	FAGCN	0.620	0.420	0.932	0.515
	ACM-GCN	0.622	0.476	0.888	0.474
	GPR-GNN	0.645	0.464	0.939	0.575
Non-local	Ordered GNN	0.145	0.016	0.682	0.125
Hybrid	H2GCN	**0.774**	**0.719**	**0.974**	**0.757**
	LINKX	0.578	0.451	0.894	0.455

4.1 Controlled homophily sweep (synthetic)

The per-dataset comparison above is suggestive but hard to follow: the real graphs differ in node features, size and domain as well as in homophily, so the apparent homophily trend could be impacted by any of those. No real dataset lets us separate these factors, because homophily cannot be changed without also changing the graph it belongs to. To isolate homophily as the single axis we therefore turn to synthetic data, where a generator exposes homophily as a tunable parameter and the remaining properties can be controlled. Using the synthetic multi-label graph generator, we produce a family of graphs that vary their target homophily level h from 0.2 (strongly heterophilous) to 0.7 while holding node count, label space, label cardinality and features fixed, and re-run the six heterophily GNNs (ACM-GCN, FAGCN, GPR-GNN, H2GCN, LINKX and Ordered GNN) on each under the same pipeline. Figure 2 reports test average precision against h, with the feature-only MLP included as a baseline reference. The DeepWalk baseline is omitted from the plot because we are interested in the behavior of the heterophily-specialised models.

The overall trend is monotone: average precision rises (or stays flat) as the graph becomes more homophilous, and no model improves as heterophily increases, which supports the pattern we could observe in the real-world datasets. Against this background two models stand out. LINKX is the most dramatic: at the most heterophilous point ($h = 0.2$) its AP sits near the bottom of the field (0.19), but as soon as the graph is even mildly homophilous ($h \geq 0.3$) it jumps almost to the ceiling and stays there. The feature-only MLP, by contrast, is flat across h since it never sees the graph, so the heterophily GNNs pull clearly ahead of it only once the graph becomes homophilous.

4.2 Controlled label-cardinality sweep (synthetic)

The homophily sweep holds the number of labels per node fixed and varies h, here we do the opposite. Using the same synthetic generator (Section 3.3) we fix homophily at $h = 0.20$ and mean degree at 10 and sweep the number of labels per node k over $\{2, \ldots, 12\}$, re-running every model under the same pipeline (the figure shows $k \geq 4$, where the curves separate). This asks whether the heterophily GNNs' weakness in the heterophilous regime is an impact of one the label cardinalities. Figure 3 reports test average precision against k.

The figure shows the six heterophily GNNs together with the feature-only MLP baseline. The MLP sits at the top across the entire swept range and no heterophily GNN overtakes it. Among the GNNs the ordering is stable: H2GCN sits just below the MLP, Ordered

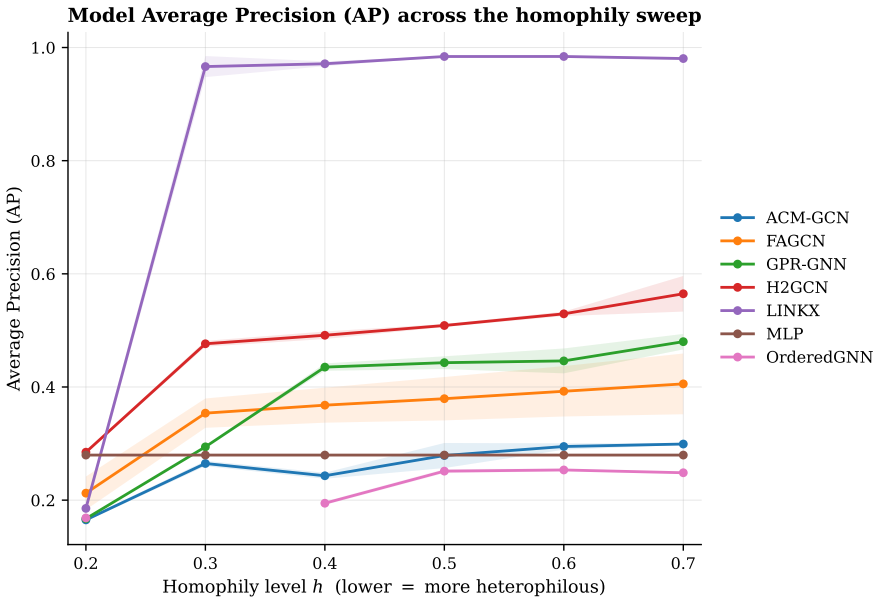

Figure 2: Average precision versus target homophily h on the synthetic sweep for the six heterophily GNNs and the feature-only MLP baseline. Node count, labels and features are held fixed across h. Shaded bands show variance across seeds.

GNN and FAGCN follow, and the two spectral message-passing models (ACM-GCN, GPR-GNN) stay pinned near the bottom at every cardinality, close to the identity-feature floor. A plausible reading is that these two would be that both apply a filter on top of the aggregated representation, so at $h = 0.20$, where adjacent nodes share few labels, the informative features are blended with largely unrelated neighbours before the classifier sees them. The same two models also collapse to near-chance AP on the heterophilous real-world graphs (Tables 5, 6), so this is not specific to the synthetic data. Two trends cut across this ordering. The absolute AP of the strongest models (the MLP and H2GCN) drifts *down* as k grows, which is expected since predicting more co-occurring labels per node is a harder problem, while Ordered GNN stays roughly flat. LINKX moves the other way, climbing from the bottom group at $k = 4$ towards the middle band by $k = 12$, plausibly because its adjacency-row information has more structure to exploit as the label sets grow.

4.3 Binarization of multi-label datasets

A natural objection to the picture above is that the message-passing GNNs are not really weak on these graphs. (Section 3.4) probes this: we collapse each real dataset to a multi-class problem on the identical graph and re-run every model. Figure 4 plots raw AP on each side, one slope per model. We flag that this is the weakest experiment in the study and report it mainly for completeness. Collapsing each node to its single dominant label does not just simplify the target, it changes the dataset so that the two sides are no longer the same problem: the multi-label structure that makes these benchmarks what they are is gone, and the collapse also moves the graph's effective homophily. So while we present the

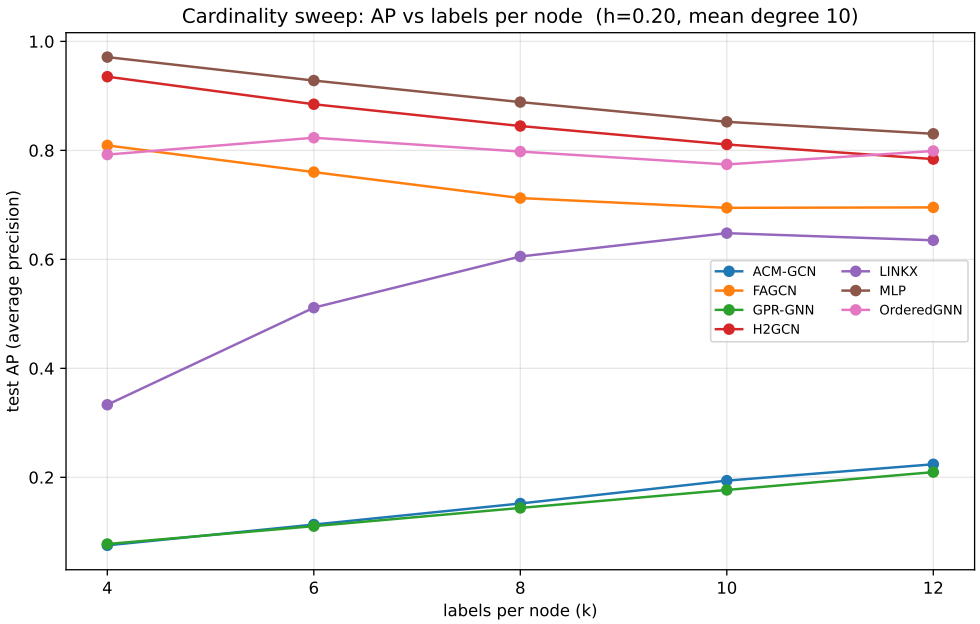

Figure 3: Average precision versus labels per node k on the synthetic cardinality sweep (homophily fixed at $h = 0.20$, mean degree 10), one line per model

numbers, we do not treat them as a clean before/after on the same datasets, and we keep them out of the main claims.

If the multi-label setting were the only thing holding the GNNs back, every panel should slope upward. It does not. The effect is overwhelmingly concentrated on **BlogCatalog**, where collapsing to multi-class lifts the message-passing models sharply, vaulting them from the MLP floor to well above it. Elsewhere the picture is flat or runs the other way: on **PCG** and **DBLP** binarizing nudges most models *down* and on EukaryoteGO and HumanGO the slopes are small and mixed.

Why this is not the result it appears to be. The BlogCatalog jump is tempting to read as "the GNNs were fine all along." We think that reading is wrong, and we report the experiment mainly to rule it out rather than to support it. First, the comparison is *confounded by homophily*: the dominant-label collapse leaves the graph fixed but, by reducing each node to its single most frequent tag, raises the multi-class edge homophily above the multi-label Jaccard homophily of the same graph (Section 3.4). On a near-zero-homophily graph like BlogCatalog that is a large jump, so the GNNs improve because they are handed a *more homophilous* graph. Second, AP is now computed over a *different and easier target*: binarization shrinks the label space to the set of surviving dominant labels and asks for one label per node instead of a set, so a larger number need not mean a method is better at the problem we care about. Third, and most basic, the collapse *discards the multi-label structure* that is the entire object of this study, whatever the binarized score measures, it is no longer multi-label node classification and the datasets loose learning information from the other nodes. For these reasons we treat the experiment as a diagnostic, it confirms

the GNNs *can* exploit these graphs once homophily is restored, and not as evidence about their standing on the multi-label task, and we do not fold it into the main comparison.

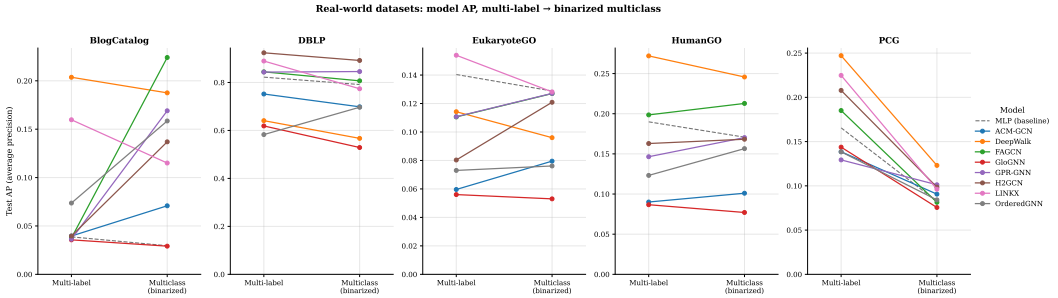

Figure 4: Binarization experiment: test AP for each model on the five real multi-label datasets in the original multi-label task (left of each panel) and after collapsing to multi-class on the same graph (right of each panel). One slope per model

Excluded benchmark. We also attempted the large Yelp multi-label graph of Zhao et al. [13], but the runs are incomplete: several models did not finish, and H2GCN failed on every split, so we exclude Yelp from the comparison.

5 Related Work

Heterophilic graph learning. The finding that message passing degrades under heterophily has produced a large and still-growing family of specialised architectures, surveyed and taxonomised by Zheng et al. [14] (whose families we adopt in Section 2.4). This literature is almost entirely multi-class: the methods are proposed, tuned and benchmarked on datasets where each node carries exactly one label. Platonov et al. [11] further show that even there the reported gains can be fragile, with simple baselines competitive once the evaluation is standardised, our Roman-Empire control is taken from that critique.

Measuring homophily. A parallel thread asks how homophily should even be quantified. Edge homophily [15] and similar multi-class ratios assume one label per node, while finer measures such as Cross-Class Neighbourhood Similarity [8] aim to characterise *when* heterophily actually harms a GNN rather than treating low h as uniformly damaging. None of these transfers cleanly to set-valued labels, where two neighbours may agree on some labels and disagree on others - the ambiguity laid out in Section 2.

Multi-label node classification. Multi-label node classification has received comparatively little attention in the heterophily literature. Zhao et al. [13] are the closest prior work: they curate the multi-label benchmarks we reuse, propose a Jaccard-based label homophily for them, and report that simple baselines can match or beat classical GNNs and the heterophily model H2GCN. Their study pairs a broad set of datasets with a narrow set of models, however, only a single heterophily-specialised method, and does not ask whether the wider heterophily toolkit catalogued by Zheng et al. [14] behaves any differently on multi-label graphs.

This work. We address that gap empirically. We evaluate six heterophily methods spanning all three families of the taxonomy under one fixed pipeline, on the real multi-label benchmarks alongside feature- and structure-only baselines and a multi-class heterophily control, and then isolate homophily and label cardinality on synthetic graphs where each axis can be varied in turn. To our knowledge this is the first systematic test of whether designs built for multi-class heterophily carry over to the multi-label setting.

6 Responsible Research

This is an empirical methods-comparison study on public benchmark data, so the main responsibility considerations are the reproducibility of the comparison and the integrity with which we report its limitations. We address each in turn and close with a short note on ethics.

6.1 Reproducibility

The study is designed so that every reported number can be regenerated. All models are run through a single shared pipeline (Section 3.5) with fixed random seeds and a fixed set of train/validation/test splits reused identically across every model, so that score differences reflect the model and not the data handling or seed luck. We deliberately do *not* tune hyper-parameters: each model uses its full published upstream defaults, recorded in version-controlled configuration files (one per model/dataset cell), which removes tuning effort and makes the exact settings inspectable. The real datasets are the public benchmarks of Zhao et al. [13] and Platonov et al. [11], reused unchanged, the synthetic graphs are produced by deterministic generators with a fixed graph seed and analytically calibrated parameters (Section 3.3), so each sweep cell is reproducible. The code, configuration files and plotting scripts that produce every table and figure are released with the paper.

6.2 Integrity and threats to validity

We have tried to report the comparison honestly, including where it is weak, rather than only its cleanest parts.

- **Correlational claims on real data.** On the real datasets homophily cannot be separated from size, domain and feature richness, so we phrase the homophily trend as a correlation ("performance appears to track homophily") and lean on the synthetic sweeps for the controlled version, rather than claiming causation from the real graphs.
- **A density confound in the homophily sweep.** Our homophily-sweep generator targets the homophily level h but does not pin the mean degree, which co-varies strongly with h (falling from roughly 1.5×10^3 at $h = 0.2$ to about 160 at $h = 0.7$). The low-homophily graphs are therefore also much denser, so part of the structure-reading models' behaviour there reflects graph density rather than homophily alone, and the sweep does not isolate homophily as cleanly as a fully controlled design would. The cardinality sweep, whose generator calibrates each graph to a fixed mean degree of 10, is the more tightly controlled of the two synthetic experiments.
- **Uneven statistical support.** Three real datasets are averaged over multiple splits, but EukaryoteGO and HumanGO use a single predefined split and so carry no cross-split variance. The homophily sweep also has thin cells: Ordered GNN is missing at

$h = 0.3$ and has a single seed at $h = 0.7$. We plot such cells as-is rather than hiding them, and read fragile points cautiously.

- **Incomplete runs reported as such.** The large Yelp benchmark did not finish for several models (H2GCN failed on every split), so we exclude it explicitly (Section 4) rather than presenting a partial table as if complete.
- **Diagnostics kept separate.** The binarization check (Section 4.3) is confounded by construction: collapsing each node to a single label discards the multi-label structure and shifts the effective homophily, so the binarized graphs are not really the same datasets. We report it for completeness and transparency but present it only as a supplementary diagnostic, deliberately kept out of the main claims.

Several of these limitations reappear as concrete next steps in Section 7.1.

6.3 Ethical considerations

The work raises little ethical concerns. All datasets are established, publicly released benchmarks (protein-function, gene-ontology, citation, word and blog-tag graphs) used here only to compare model accuracy, none is collected by us, and none contains sensitive personal data beyond what those public benchmarks already anonymise. The study builds no deployed system and makes no individual-level predictions, so the risk of direct harm is low. The main material cost is compute: training the full model suite across many datasets, seeds and synthetic cells consumes energy, which we contain by keeping to modest model sizes, stopping early on validation loss, and running each cell only as many times as the seed count requires.

7 Discussion

Across the multi-label graphs, the results appear to track label homophily. On the three lowest-homophily multi-label graphs: BlogCatalog ($h = 0.10$), PCG ($h = 0.17$) and HumanGO ($h = 0.42$), the best model is LINKX (an MLP on features plus adjacency rows) or DeepWalk (a structural embedding), with a plain MLP often competitive at a fraction of the runtime, and ACM-GCN the most frequent low point. EukaryoteGO ($h = 0.46$) sits at the boundary: there the field is tight and H2GCN edges ahead on the F1 metrics, so we read it as a transition point rather than a clean win for either side. On the high-homophily multi-label graph DBLP ($h = 0.76$) the ordering largely reverses: H2GCN and the other message-passing models come out ahead, much as on the multi-class control Roman-Empire, where H2GCN leads and the methods behave broadly as the literature reports [14].

One reading that would be consistent with these is the following. The models that do well tend to be the ones that do *not* force node features and graph structure to be used together to make a prediction. DeepWalk relies on structure alone and the MLP on features alone, so each leans on whichever source is informative and is unaffected by the other. LINKX uses both but processes them separately, an MLP on the features and an MLP on each node's row of the adjacency matrix, and only combines them at the end. Because none of these models reads a node's features *through* its neighbours, an uninformative neighbourhood cannot wash the feature signal out, and they may degrade more gracefully when one of the two sources is uninformative. The models that do worse tend to be those that *entangle* the two by passing features through neighbour aggregation: when neighbourhoods are only

weakly label-consistent (low h), aggregation mixes a node's own feature signal with poorly-related neighbours and can wash it out, and the more aggressive the approach (for instance ACM-GCN's spectral channel selection) the more capacity there is to fit noise instead. This is consistent with the hypothesis from Section 2: in the low-homophily multi-label setting, label co-occurrence may already absorb much of the structure that signed and high-pass filters are designed to recover, so the expected heterophily advantage could shrink or disappear while node features carry most of the usable signal. When label homophily is high, as in DBLP, that structure is informative again, aggregation appears to stop hurting, and the message-passing models tend to recover their advantage. The synthetic homophily sweep (Section 4.1, Fig. 2) offers more direct, though still limited, support for reading this: holding every graph property but h fixed, the average precision of the heterophily GNNs falls (or, for Ordered GNN, stays flat) as the graph grows more heterophilous, and none gains an edge in the heterophilous part of the range we tested. LINKX makes the point most sharply, collapsing from near-perfect AP to the bottom of the field at the most heterophilous setting.

A further control reinforces rather than complicates this reading. Holding homophily fixed at $h = 0.20$ and instead varying the number of labels per node (Section 4.2, Fig. 3) leaves the ordering essentially unchanged: no heterophily-specialised design overtakes the feature-only MLP at any label cardinality we tested, so the disadvantage in the heterophilous regime is not related to label density.

7.1 Future work

The remaining work falls into four threads.

1. **Deepen the homophily analysis.** We report Zhao et al.'s Jaccard label homophily per dataset, the next step is to compute the per-class Cross-Class Neighbourhood Similarity (CCNS) [13, 8] under our pipeline and relate it to where the heterophily methods help or fail.

2. **Strengthen the statistics.** Extend multi-split runs to all datasets (currently EukaryoteGO and HumanGO are single-split), report variance/confidence intervals.

3. **Probe the co-occurrence hypothesis.** Test directly whether label correlations explain the heterophily advantage, e.g. by controlling for co-occurrence structure.

4. **Broaden coverage.** Complete the large Yelp benchmark (on which several models, including H2GCN, currently fail) and add further heterophily models from the survey's taxonomy [14] to confirm the patterns generalise.

8 Conclusion

We presented a controlled comparison of two baselines and six heterophily-oriented GNNs across five real multi-label graphs and one multi-class control, run through a single shared pipeline so that score differences are largely attributable to the models. The results appear to track label homophily: on the lower-homophily multi-label graphs the heterophily-specialised message-passing architectures rarely improved on a plain MLP or a structure-only DeepWalk embedding, with LINKX often competitive and ACM-GCN the most frequent low point, whereas on the high-homophily multi-label graph (DBLP) and the multi-class control the expected ordering largely returned, with H2GCN leading. A synthetic control points

the same way: holding homophily fixed while varying the number of labels per node leaves the ordering unchanged, collapsing the multi-label task to multi-class prediction restores the GNNs' advantage only where it incidentally raises homophily, but we lean on it lightly: the collapse discards the multi-label structure that defines these datasets, so the binarized graphs are not really the same datasets and we report the check only for completeness. Taken together, these results seem to extend to the heterophily-specialised methods a pattern that prior work observed for classical GNNs [13], and they suggest that the advantages reported for heterophily methods on multi-class benchmarks [14] may not transfer automatically to the low-homophily multi-label regime.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *International Conference on Machine Learning (ICML)*, 2019.

[2] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *AAAI Conference on Artificial Intelligence*, 2021.

[3] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations (ICLR)*, 2021.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[6] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[7] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[8] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2022.

[9] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[10] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

[11] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *International Conference on Learning Representations (ICLR)*, 2023.

[12] Yunchong Song, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. Ordered gnn: Ordering message passing to deal with heterophily and over-smoothing. In *International Conference on Learning Representations (ICLR)*, 2023.

[13] Tianqi Zhao, Ngan Thi Dong, Alan Hanjalic, and Megha Khosla. Multi-label node classification on graph-structured data. *Transactions on Machine Learning Research*, 2023.

[14] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2024.

[15] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

A Training dynamics

For completeness, Figures 5-11 show train and validation loss against epoch for the seven gradient-trained models (the feature-only MLP and the six GNNs). DeepWalk is omitted, as it produces unsupervised random-walk embeddings and has no train/validation loss in this sense. Within each figure, every panel is one dataset, shaded bands show ± 1 std over splits (where more than one split is available) and the green marker is the early-stopping (best-validation) epoch.

One thing is worth noting. Several models (ACM-GCN and Ordered GNN on Roman-Empire and DBLP, LINKX on PCG and Human-GO) show validation loss turning sharply upward after an early minimum, the best-validation marker indicates that early stopping selects that dip rather than the over-fit tail.

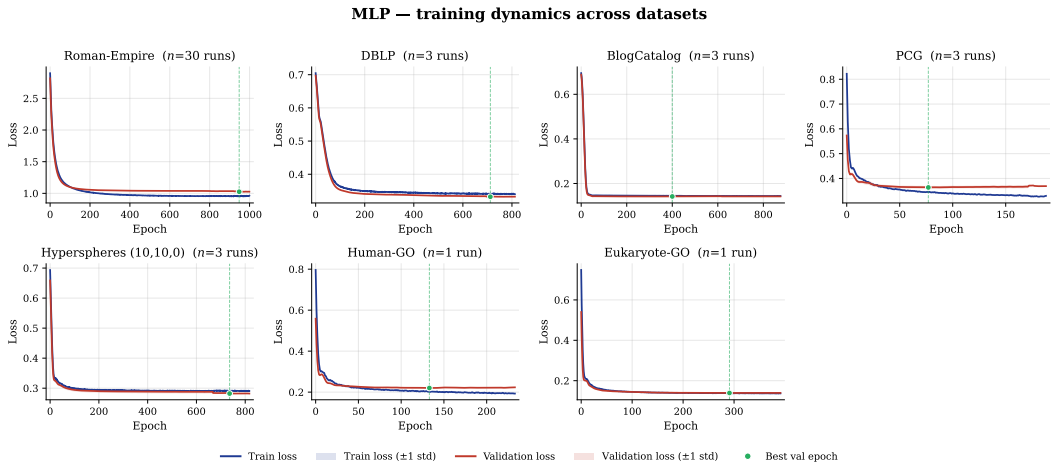

Figure 5: MLP training dynamics across the six datasets. Loss converges cleanly with a small train-validation gap on every dataset.

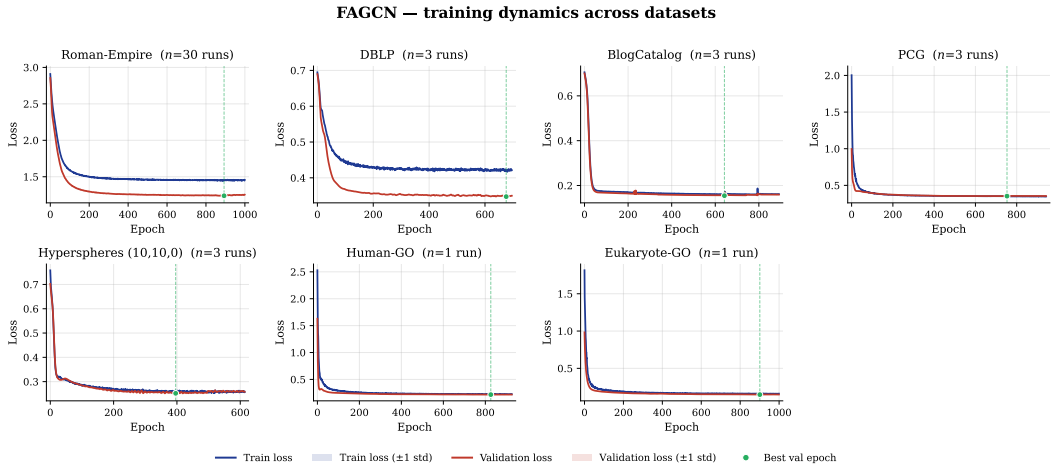

Figure 6: FAGCN training dynamics across the six datasets. Training is stable throughout.

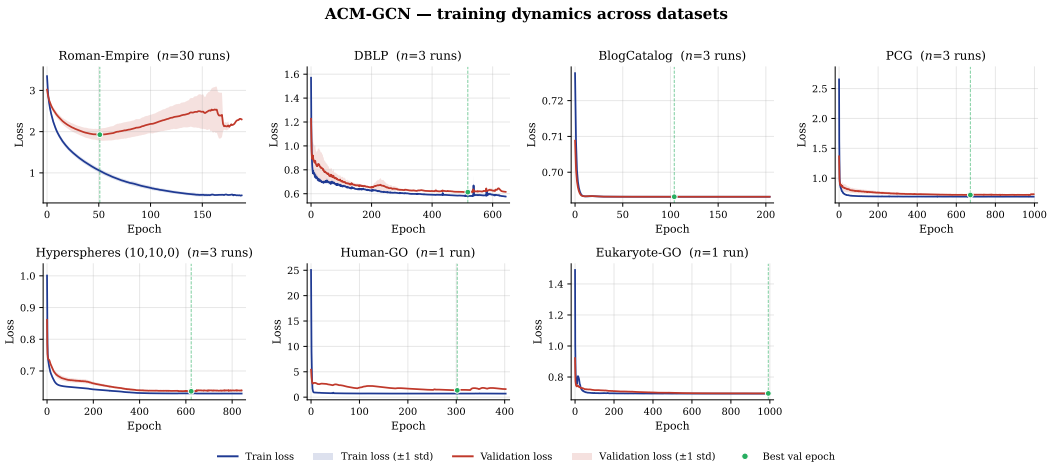

Figure 7: ACM-GCN training dynamics across the six datasets. On Roman-Empire the validation loss rises after the best epoch (over-fitting), which early stopping cuts off.

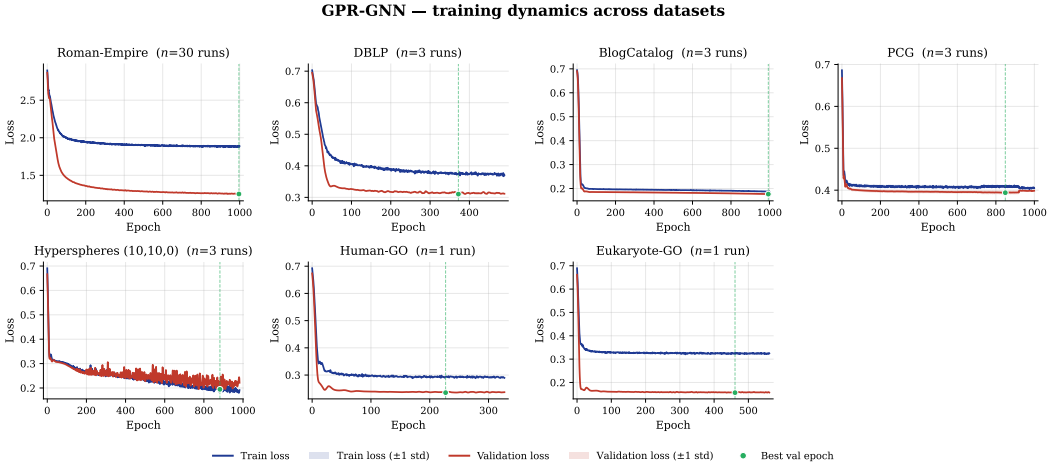

Figure 8: GPR-GNN training dynamics across the six datasets.

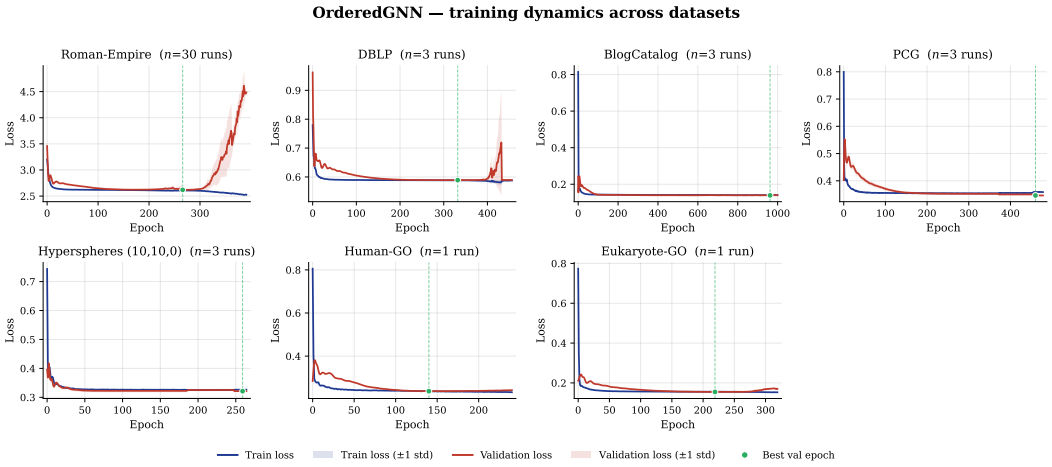

Figure 9: Ordered GNN training dynamics across the six datasets. Validation loss diverges late on Roman-Empire and DBLP, early stopping selects the earlier minimum.

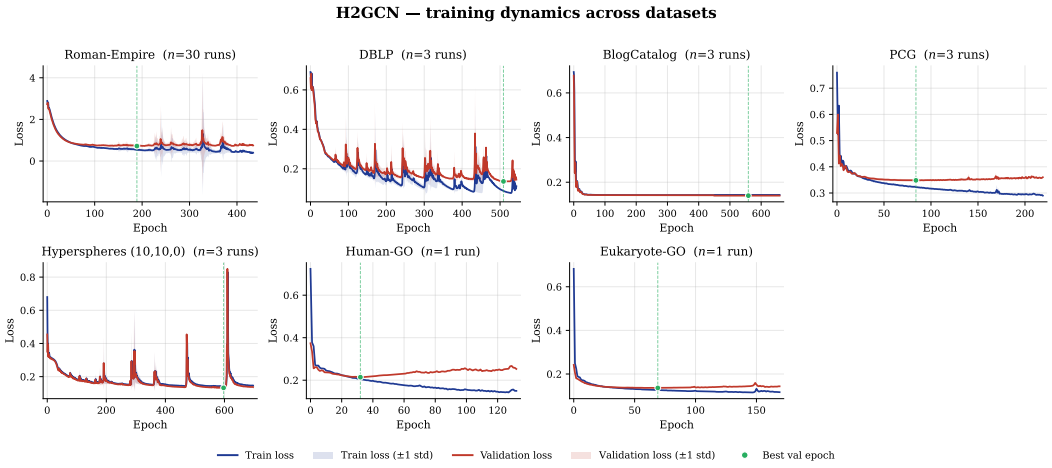

Figure 10: H2GCN training dynamics across the six datasets. Periodic loss spikes appear on several graphs, but training recovers after each.

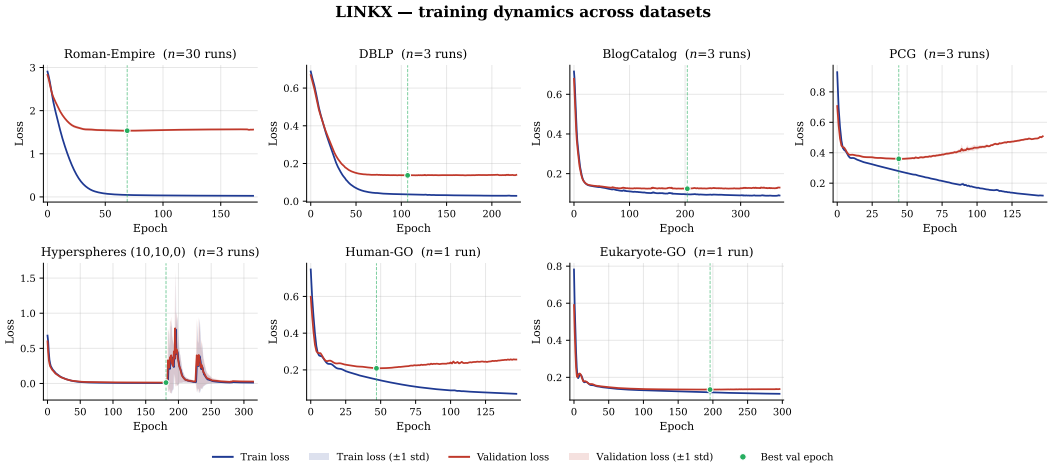

Figure 11: LINKX training dynamics across the six datasets. On PCG and Human-GO the validation loss turns upward after an early minimum, early stopping selects the dip.