Re-evaluating the Advancements of Heterophilic Graph Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Over the past decade, Graph Neural Networks (GNNs) have achieved great success on machine learning tasks with relational data. However, recent studies have found that heterophily can cause significant performance degradation of GNNs, especially on node-level tasks. Numerous heterophilic benchmark datasets have been put forward to validate the efficacy of heterophily-specific GNNs, and various homophily metrics have been designed to help recognize these challenging datasets. Nevertheless, there still exist multiple pitfalls that severely hinder the proper evaluation of new models and metrics: 1) lack of hyperparameter tuning; 2) insufficient evaluation on the truly challenging heterophilic datasets; 3) missing quantitative evaluation for homophily metrics on synthetic graphs. To overcome these challenges, we first train and fine-tune baseline models on 27 most widely used benchmark datasets, and categorize them into three distinct groups: malignant, benign and ambiguous heterophilic datasets. We identify malignant and ambiguous heterophily as the truly challenging subsets of tasks, and to our best knowledge, we are the first to propose such taxonomy. Then, we re-evaluate 11 state-of-the-arts (SOTA) GNNs, covering six popular methods, with fine-tuned hyperparameters on different groups of heterophilic datasets. Based on the model performance, we comprehensively reassess the effectiveness of different methods on heterophily. At last, we evaluate 11 popular homophily metrics on synthetic graphs with three different graph generation approaches. To overcome the unreliability of observation-based comparison and evaluation, we conduct the first quantitative evaluation and provide detailed analysis.

1 Introduction

As a generic data structure, graph is capable of modeling complex relations among objects in many real-world problems [15, 19, 24, 58, 59]. In the last decade, various Graph Neural Networks (GNNs) architectures have been proposed [7, 10, 18, 20, 26, 29, 41, 57, 61, 65] and shown to outperform traditional neural networks (NNs) in modeling graph-based real-world tasks [6, 25, 35, 50, 53, 68, 69].

The success of GNNs, especially on node-level tasks, is commonly believed to be attributed to the homophily principle [49], which means that connected nodes are more likely to have similar labels [52] or attributes [21]. This inductive bias is believed to be a major contributor to the superiority of GNNs over traditional NNs on various tasks [74]. On the other hand, heterophily, *i.e.,* the lack of homophily [34, 37], is often cited as the primary reason for the inferiority of GNNs on heterophilic graphs. In such graphs, edges typically connect nodes from different classes, which can lead to mixed and indistinguishable node embeddings during the message passing process [42, 74]. Recently, numerous models have been proposed to deal with heterophily [5, 9, 23, 31, 33, 36, 37, 39, 42, 52, 70, 72, 74] and many homophily metrics have been put forward to identify the graph datasets that are unfriendly to GNNs [37, 71]. (We provide a brief introduction in Appendix A)

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

However, to date, there is no work that strictly validate the recent advances in heterophilic graph learning. Upon examination, we empirically identify several pitfalls that can severely impede fair and accurate assessment of models and metrics:

- **Lack of Adequate Hyperparameter Tuning.** With careful hyperparameter tuning, it is empirically found that basic GNNs can actually outperform some heterophily-specific models on several heterophilic graphs [43, 44]. This indicates that there potentially exist a substantial amount of inaccurate and biased results reported in current literature, which hinder fair comparison and mislead our understanding of heterophily problem.
- **Insufficient Evaluation on Truly Challenging Heterophilic Datasets.** Based on the studies in [38, 40, 44], the real challenging heterophilic datasets are those where graph-aware models underperform graph-agnostic models, instead of those with small homophily values. With this criterion, the evaluation results on many commonly used benchmark datasets cannot adequately validate the effectiveness of models on heterophily.
- **Absence of Quantitative Evaluation Benchmark for Homophily Metrics.** The existing evaluation methods mainly depends on observing how well the homophily metric correlates with GNN performance on synthetic graphs. However, such observation-based comparison can easily lead to subjective and unreliable conclusions, and there is currently no quantitative evaluation of homophily metrics.

In this paper, we aim to address the above issues and our main contributions are:

- **Discover New Categorization of Heterophily and Identify the Challenging Ones.** To find out the real challenging heterophilic datasets, in Section 2.2, we fine-tune baseline graph-aware models and their corresponding graph-agnostic models on 27 mostly used benchmark datasets. We find that there exist three disjoint sets of heterophilic datasets, where graph-aware models: 1) consistently outperform graph-aware models; 2) consistently underperform graph-aware models; 3) have inconsistent performance against graph-aware models. Based on this discovery, we categorize them into three types of heterophilic graphs: malignant, benign and ambiguous, and we argue that the malignant and ambiguous datasets are the truly challenging ones which should be used to validate the effectiveness of models.
- **Reassess Popular Heterophily-specific Graph Models.** In Section 2.3, we reassess 11 state-of-the-arts (SOTA) GNNs with fine-tuned hyperparameters on each categories of the 27 benchmark datasets. Based on the results, the efficacy of some widely used methods is found questionable, *e.g.,* most SOTA GNNs are not significantly better than the best baseline models in any category of heterophilic datasets, and some of them actually compromise their performance on easy graphs to obtain better results on challenging graphs.
- **Quantitative Evaluation for Homophily Metrics.** In Section 3.1, we evaluate 11 popular homophily metrics on synthetic graphs with three graph generation approaches. To compare them strictly and accurately, in Section 3.2, we use Pearson correlation and Fréchet distance to assess the metrics quantitatively. We conduct detailed analysis and provide insights for future evaluation.

2 Categorization of Heterophily Datasets and Model Re-Evaluation

In this section, we conduct a series of experiments with fine-tuned hyperparameters for accurate assessment and fair comparison of GNNs built for heterophilic graphs. Specifically, in Section 2.1, we introduce the 27 benchmark datasets used in this paper and the experimental setups; in Section 2.2, based on the performance of fine-tuned graph-aware and graph-agnostic models, we classify the existing heterophily benchmark datasets into malignant, benign and ambiguous groups, and we argue that the real challenging tasks are on malignant and ambiguous datasets; in Section 2.3, we re-evaluate 11 popular SOTA models with fine-tuned hyperparameters on each group of heterophilic graphs to reassess their effectiveness and disclose their limitations on addressing heterophily.

2.1 Experimental Settings

We collect 27 mostly used benchmark datasets for heterophily research [32, 33, 52, 55, 56, 60, 73]. The dataset names and data splits are,

- *Cornell, Wisconsin, Texas, Film* are from [52], *Chameleon, Squirrel* are from [56], *Cora, CiteSeer, PubMed* are from [67]. We use the data processed by [52]. The models are trained on 10 random splits with 60%/20%/20% for train/validation/test, which follows [9].
- *Deezer-Europe, genius, arXiv-year, Penn94, pokec, snap-patents, twitch-gamers* are from [32, 33]. We train models on each dataset with five fixed 50%/25%/25% splits for train/validation/test, which is the same as [32, 33].
- *roman-empire, amazon-ratings, minesweeper, tolokers, questions, Chameleon-filtered, Squirrel-filtered* are from [55]. The models are trained on 10 fixed splits with 50%/25%/25% samples for train/validation/test, which is provided by [55].
- *BlogCatalog, Flickr, BGP, Wiki-cooc* are from [60, 73]. The splits for training/validation/test are 10 60%/20%/20% random splits, which is the same as [9].

For other experimental settings such as early stopping, optimizer, max number of training epochs, evaluation metrics, we all follow the original papers. The hyperparameter searching range is shown in Appendix A.4.

Computing Resources For all experiments on real-world and synthetic datasets, we use NVIDIA V100 GPUs with 16/32GB GPU memory. The software implementation is based on PyTorch and PyTorch Geometric [14].

2.2 Categorization of Heterophily Datasets

Categories		Datasets	#nodes	#edges	#feature dim	#classes	H_{edge}	H_{node}	Eval Metric	GCN	MLP-2	SGC-1	MLP-1	Literature
Heterophily Graphs	Malignant	Cornell	183	295	1,703	5	0.2983	0.2001	Accuracy	82.46 ± 3.11	91.30 ± 0.70	70.98 ± 8.39	93.77 ± 3.34	[52]
		Wisconsin	251	499	1,703	5	0.1703	0.0991	Accuracy	75.5 ± 2.92	93.87 ± 3.33	70.38 ± 2.85	93.87 ± 3.33	[52]
		Texas	183	309	1,703	5	0.0615	0.0555	Accuracy	83.11 ± 3.2	92.26 ± 0.71	83.28 ± 5.43	93.77 ± 3.34	[52]
		Film	7,600	33,544	931	5	0.2163	0.2023	Accuracy	35.51 ± 0.99	38.58 ± 0.25	25.26 ± 1.18	34.53 ± 1.48	[52]
		Deezer-Europe	28,281	92,752	31,241	2	0.5251	0.5299	Accuracy	62.23 ± 0.53	66.55 ± 0.72	61.63 ± 0.25	63.14 ± 0.41	[33]
		genius	421,961	984,979	12	2	0.6176	0.0985	Accuracy	83.26 ± 0.14	86.62 ± 0.08	82.31 ± 0.45	86.48 ± 0.11	[32]
		roman-empire	22,662	32,927	300	18	0.0469	0.046	Accuracy	48.92 ± 0.46	66.04 ± 0.71	44.60 ± 0.52	64.12 ± 0.61	[55]
		BlogCatalog	5,196	171,743	8,189	6	0.4011	0.3914	Accuracy	79.67 ± 1.06	92.97 ± 0.89	71.07 ± 1.15	91.86 ± 0.93	[73]
		Flickr	7,575	239,738	12,047	9	0.2386	0.2434	Accuracy	71.38 ± 1.00	90.24 ± 0.96	60.10 ± 1.21	89.91 ± 0.97	[73]
		BGP	63,977	174,803	287	8	0.2545	0.083	Accuracy	62.56 ± 0.94	65.56 ± 0.55	61.74 ± 0.73	64.67 ± 0.81	[60]
	Benign	Chameleon	2,277	36,101	2,325	5	0.2339	0.2467	Accuracy	64.18 ± 2.62	46.72 ± 0.46	64.86 ± 1.81	45.01 ± 1.58	[56]
		Squirrel	5,201	217,073	2,089	5	0.2234	0.2154	Accuracy	44.76 ± 1.39	31.28 ± 0.27	47.62 ± 1.27	29.17 ± 1.46	[56]
		Chameleon-filtered	890	8,854	2,325	5	0.2361	0.2441	Accuracy	41.46 ± 3.42	38.06 ± 3.98	44.00 ± 3.10	35.72 ± 2.23	[55]
		arXiv-year	169,343	1,166,243	128	5	0.2218	0.2778	ROC AUC	40 ± 0.26	36.36 ± 0.23	35.58 ± 0.22	34.11 ± 0.17	[32]
		amazon-ratings	24,492	93,050	300	5	0.3804	0.3757	Accuracy	50.05 ± 0.67	49.55 ± 0.81	40.69 ± 0.42	38.60 ± 0.41	[55]
		Wiki-cooc	10,000	2,243,042	100	5	0.3435	0.175	Accuracy	95.40 ± 0.41	89.38 ± 0.42	72.38 ± 0.78	48.86 ± 0.37	[73]
	Ambiguous	Squirrel-filtered	2,223	46,998	2,089	5	0.2072	0.1905	Accuracy	37.33 ± 1.88	38.30 ± 1.22	37.54 ± 2.13	30.14 ± 1.53	[55]
		Penn94	41,554	1,362,229	5	2	0.4704	0.4828	Accuracy	82.08 ± 0.31	74.68 ± 0.28	67.06 ± 0.19	73.72 ± 0.5	[32]
		pokec	1,632,803	30,622,564	65	2	0.4449	0.3931	Accuracy	70.3 ± 0.1	62.13 ± 0.1	52.88 ± 0.64	59.89 ± 0.11	[32]
		snap-patents	2,923,922	13,975,788	269	5	0.073	0.1857	Accuracy	35.8 ± 0.05	31.43 ± 0.04	29.65 ± 0.04	30.59 ± 0.02	[32]
		twitch-gamers	168,114	6,797,557	7	2	0.545	0.556	Accuracy	62.33 ± 0.23	60.9 ± 0.11	57.9 ± 0.18	59.45 ± 0.16	[32]
Homophily Graphs		Cora	2,708	5,429	1,433	7	0.8100	0.8252	Accuracy	87.78 ± 0.96	76.44 ± 0.30	85.12 ± 1.64	74.3 ± 1.27	[67]
		CiteSeer	3,327	4,732	3,703	6	0.7355	0.7062	Accuracy	81.39 ± 1.23	76.25 ± 0.28	79.66 ± 0.75	75.51 ± 1.35	[67]
		PubMed	19,717	44,338	500	3	0.8024	0.7924	Accuracy	88.9 ± 0.32	86.43 ± 0.13	86.5 ± 0.76	86.23 ± 0.54	[67]
		minesweeper	10,000	39,402	7	2	0.6828	0.6829	ROC AUC	72.34 ± 0.93	50.92 ± 1.25	82.04 ± 0.77	50.59 ± 0.83	[55]
		tolokers	11,758	519,000	10	2	0.5945	0.6344	ROC AUC	77.44 ± 1.32	74.58 ± 0.75	73.80 ± 1.35	71.89 ± 0.82	[55]
		questions	48,921	153,540	301	2	0.8396	0.898	ROC AUC	72.72 ± 1.93	69.97 ± 1.16	71.06 ± 0.92	70.33 ± 0.96	[55]

Table 1: Categorization of benchmark datasets. The cells marked by green are the better results for the comparison of graph-aware models vs. graph-agnostic models.

As stated in [38, 40, 44], heterophily does not always lead to inferior performance and homophily is not always necessary for GNNs. Therefore, empirical results are important to identify and distinguish the truly difficult and pseudo-difficult heterophily[1] datasets. Specifically, we conduct ablation study to evaluate the impact of graph structure on message passing (MP). For example, if we remove the MP step in GCN and the model performance decreases, this means MP with the graph structure is beneficial; otherwise, the MP with graph structure is harmful. In this way, we can reliably find out the easy and challenging heterophilic datasets. We have also compared the linear models SGC vs. MLP-1. Note that sufficient hyperparameter tuning for each baseline model is important to guarantee fair comparison. The statistics and experimental results are shown in Table 1. From the results, we have identified heterophilic datasets with distinct properties as follows,

- **Malignant and Benign Heterophily:** There exist a subset of heterophilic datasets where the graph-aware models consistently underperform their corresponding graph-agnostic models, *e.g., Cornell, Wisconsin, Texas, Film, Deezer-Europe, genius, roman-empire, BlogCatalog, Flickr* and *BGP*, which indicates that these heterophilic graph structure provides harmful information in feature aggregation step. On the other hand, there exist another class of

[1] In this paper, a dataset is considered as heterophilic if at least one of its H_{edge} or H_{node} value is smaller or close to 0.5, otherwise it is homophilic.

heterophilic graphs where the graph-aware models consistently outperform graph-agnostic models, *e.g., Chameleon, Squirrel, Chameleon-filtered, arXiv-year, amazon-ratings* and *Wiki-cooc*, which indicates that these heterophilic graph structures actually provide beneficial information for GNNs and do not cause any trouble to graph learning.[2] We call them **malignant and benign heterophilic datasets**, separately.

- **Ambiguous Heterophily:** Besides, we discover a third group of heterophily datasets, where there exists inconsistency between linear and non-linear graph-aware models compared with their coupled graph-agnostic models. For instance, on *Penn94, pokec, snap-patents* and *twitch-gamers*, GCN (non-linear model) outperforms MLP-2, while SGC-1 (linear model) underperforms MLP-1; on *Squirrel-filtered*, GCN underperforms MLP-2 but SGC-1 outperforms MLP-1. Such inconsistency indicates that the underlying synergy between graph structure and model non-linearity can influence GNN performance together. However, no theory can explain such relationship for now. Thus, we call this group of datasets the **ambiguous heterophilic dataset.**
- **Remark:** The tasks on malignant and ambiguous heterophilic dataset are considered as the truly challenging ones and benign heterophily is a pseudo challenge. Besides, some newly published heterophilic datasets are actually homophilic dataset as their homophily values are much larger than 0.5, *e.g.,minesweeper, tolokers, questions*. These homophilic and benign heterophilic datasets should not be used to evaluate GNNs on heterophily issue.

Methods			Baseline		Ego-Neighbor Sep.	Negative Message Passing			HP + Ada. Mixing	Selective Message Passing		Spectral GNN		Multi-hop GNN	
Categories		Datasets	Best	Worst	H_2GCN	GPRGNN	FAGCN	GloGNN*	ACM-GCN*	GBK-GNN	FSGNN	JacobiConv	BernNet	APPNP	LINKX
Heterophily Graphs	Malignant	Cornell	93.77 ± 3.34	70.98 ± 8.39	86.23 ± 4.71	91.36 ± 0.70	88.03 ± 5.6	86.32 ± 3.62	95.9 ± 1.83	80.26 ± 7.92	91.58 ± 4.68	89.74 ± 5.58	92.13 ± 1.64	87.37 ± 5.49	82.11 ± 4.53
		Wisconsin	93.87 ± 3.33	70.38 ± 2.85	87.5 ± 1.77	93.75 ± 2.37	89.75 ± 6.37	89.98 ± 2.63	97.5 ± 1.25	85.10 ± 5.49	89.22 ± 3.19	88.04 ± 4.15	87.25 ± 3.75	85.29 ± 5.77	83.53 ± 4.74
		Texas	93.77 ± 3.34	83.11 ± 3.2	85.90 ± 3.53	92.92 ± 0.61	88.85 ± 4.39	87.62 ± 4.89	96.56 ± 2	84.21 ± 6.12	90.26 ± 4.86	88.95 ± 5.86	93.12 ± 0.65	87.89 ± 6.02	84.21 ± 6.12
		Film	38.58 ± 0.25	25.26 ± 1.18	38.85 ± 1.17	39.30 ± 0.27	31.59 ± 1.37	39.65 ± 1.03	41.86 ± 1.48	38.47 ± 1.53	37.65 ± 0.79	37.48 ± 0.76	41.79 ± 1.01	37.68 ± 0.96	35.64 ± 1.36
		Deezer-Europe	66.55 ± 0.72	61.63 ± 0.25	67.22 ± 0.90	66.90 ± 0.50	66.86 ± 0.53	OOM	67.5 ± 0.53	OOM	OOM	66.71 ± 0.6	67.03 ± 0.55	66.03 ± 0.54	65.76 ± 0.41
		genius	86.62 ± 0.08	82.31 ± 0.45	87.67 ± 0.10	90.05 ± 0.31	90.03 ± 0.20	90.91 ± 0.13	91.37 ± 0.07	OOM	89.82 ± 0.03	86.00 ± 3.52	86.16 ± 0.35	87.61 ± 0.12	90.77 ± 0.27
		roman-empire	66.04 ± 0.71	44.60 ± 0.52	60.11 ± 0.52	64.85 ± 0.27	65.22 ± 0.56	59.63 ± 0.69	71.89 ± 0.61	74.57 ± 0.47	79.92 ± 0.56	71.14 ± 0.42	65.56 ± 1.34	65.87 ± 0.53	56.15 ± 0.93
		BlogCatalog	92.97 ± 0.89	71.07 ± 1.15	97.14 ± 0.50	97.07 ± 0.45	97.31 ± 0.46	OOM	97.38 ± 0.41	OOM	97.00 ± 0.55	96.84 ± 0.36	96.95 ± 0.52	96.01 ± 0.56	95.81 ± 0.69
		Flickr	90.24 ± 0.96	60.10 ± 1.21	92.46 ± 1.00	92.60 ± 0.70	93.50 ± 0.81	OOM	92.64 ± 0.67	OOM	93.39 ± 0.99	92.29 ± 0.99	92.71 ± 0.80	91.43 ± 0.67	90.69 ± 0.73
		BGP	65.56 ± 0.55	61.74 ± 0.73	66.40 ± 0.73	65.48 ± 0.77	66.06 ± 0.54	OOM	66.79 ± 0.81	66.92 ± 0.49	66.72 ± 0.62	65.51 ± 0.53	66.04 ± 0.66	65.66 ± 0.64	63.80 ± 0.62
	Benign	Chameleon	64.86 ± 1.81	45.01 ± 1.58	52.30 ± 0.48	67.48 ± 0.40	49.47 ± 2.84	71.98 ± 2.38	76.08 ± 2.13	50.57 ± 1.86	76.95 ± 1.03	72.11 ± 2.77	68.29 ± 1.58	48.55 ± 1.89	81.38 ± 1.41
		Squirrel	47.62 ± 1.27	29.17 ± 1.46	30.39 ± 1.22	49.93 ± 0.53	42.24 ± 1.2	59.56 ± 1.82	69.98 ± 1.53	34.92 ± 1.23	72.11 ± 2.66	55.86 ± 1.46	51.35 ± 0.73	34.08 ± 1.21	77.44 ± 1.69
		Chameleon-filtered	44.00 ± 3.10	35.72 ± 2.23	42.90 ± 3.91	41.95 ± 3.68	42.87 ± 5.01	OOM	42.73 ± 3.59	36.20 ± 4.37	40.96 ± 2.73	41.42 ± 2.67	40.90 ± 4.06	37.50 ± 3.69	42.34 ± 4.13
		arXiv-year	40 ± 0.26	34.11 ± 0.17	49.09 ± 0.10	45.07 ± 0.21	40.12 ± 0.44	54.79 ± 0.25	52.49 ± 0.23	OOM	50.62 ± 0.18	38.07 ± 0.21	35.62 ± 0.19	35.23 ± 0.16	56.00 ± 1.34
		amazon-ratings	50.05 ± 0.67	38.60 ± 0.41	36.47 ± 0.23	44.88 ± 0.34	44.12 ± 0.30	36.89 ± 0.14	52.49 ± 0.24	45.98 ± 0.71	52.74 ± 0.83	43.55 ± 0.48	44.64 ± 0.56	46.02 ± 0.73	52.66 ± 0.64
		Wiki-cooc	95.40 ± 0.41	48.86 ± 0.37	98.75 ± 0.15	92.58 ± 1.18	89.50 ± 0.92	OOM	99.32 ± 0.23	OOM	98.96 ± 0.15	90.22 ± 0.6	94.76 ± 0.31	88.96 ± 0.49	98.24 ± 0.37
	Ambiguous	Squirrel-filtered	38.30 ± 1.22	30.14 ± 1.53	42.77 ± 1.61	38.05 ± 1.44	42.37 ± 1.77	OOM	42.35 ± 1.97	35.07 ± 1.26	37.56 ± 1.12	42.71 ± 1.75	41.18 ± 1.77	35.12 ± 1.12	40.10 ± 2.21
		Penn94	82.08 ± 0.31	67.06 ± 0.19	81.31 ± 0.60	81.38 ± 0.16	79.87 ± 0.82	85.74 ± 0.42	86.08 ± 0.43	OOM	OOM	83.80 ± 0.33	82.88 ± 0.52	75.57 ± 0.26	84.71 ± 0.52
		pokec	70.3 ± 0.1	52.88 ± 0.64	OOM	78.83 ± 0.05	OOM	83.05 ± 0.07	81.07 ± 0.165	OOM	OOM	75.85 ± 0.06	OOM	61.82 ± 0.19	82.04 ± 0.07
		snap-patents	35.8 ± 0.05	29.65 ± 0.04	OOM	40.19 ± 0.03	OOM	62.09 ± 0.27	54.79 ± 0.616	OOM	OOM	40.87 ± 0.04	OOM	32.47 ± 0.11	61.95 ± 0.12
		twitch-gamers	62.33 ± 0.23	57.9 ± 0.18	OOM	61.89 ± 0.29	OOM	66.34 ± 0.29	66.24 ± 0.24	OOM	61.71 ± 0.24	61.98 ± 0.06	60.08 ± 0.29	60.36 ± 0.18	66.06 ± 0.19
Homophily Graphs		Cora	87.78 ± 0.96	74.3 ± 1.27	87.52 ± 0.61	79.51 ± 0.36	88.85 ± 1.36	87.67 ± 1.16	89.75 ± 1.16	87.09 ± 1.52	87.51 ± 1.21	89.61 ± 0.96	88.52 ± 0.95	88.29 ± 1.24	82.62 ± 1.44
		CiteSeer	81.39 ± 1.23	75.51 ± 1.35	79.97 ± 0.69	67.63 ± 0.38	82.37 ± 1.46	78.91 ± 1.75	81.87 ± 1.38	76.62 ± 0.84	76.59 ± 1.45	77.60 ± 1.12	80.09 ± 0.79	74.88 ± 1.27	69.92 ± 1.36
		PubMed	88.9 ± 0.32	86.23 ± 0.54	87.78 ± 0.28	85.07 ± 0.09	89.98 ± 0.54	90.32 ± 0.54	90.96 ± 0.62	88.88 ± 0.44	90.11 ± 0.43	89.99 ± 0.39	88.48 ± 0.41	90.02 ± 0.43	88.12 ± 0.47
		minesweeper	82.04 ± 0.77	50.59 ± 0.83	89.71 ± 0.31	86.24 ± 0.61	88.17 ± 0.73	51.08 ± 1.23	84.71 ± 0.85	90.85 ± 0.58	90.08 ± 0.70	89.66 ± 0.40	77.99 ± 0.95	69.62 ± 2.11	56.78 ± 2.47
		tolokers	77.44 ± 1.32	71.89 ± 0.82	73.35 ± 1.01	72.94 ± 0.97	77.75 ± 1.05	73.39 ± 1.17	74.95 ± 1.16	81.01 ± 0.67	82.76 ± 0.61	68.66 ± 0.65	77.00 ± 0.65	76.98 ± 1.03	81.15 ± 1.23
		questions	72.72 ± 1.93	69.97 ± 1.16	63.59 ± 1.46	55.48 ± 0.91	77.24 ± 1.26	65.74 ± 1.19	62.91 ± 2.10	74.47 ± 0.86	78.86 ± 0.92	73.88 ± 1.16	70.43 ± 1.38	64.77 ± 1.32	71.96 ± 2.07
Average Ranking		Overall	5.85	11.48	6.92	7.07	5.71	**5.55**	**2.85**	7.35	**4.39**	6.33	6.20	8.41	6.67
		Malignant Heterophily	6.10	12.10	6.50	5.30	5.60	6.33	**1.60**	7.33	**4.56**	7.30	**5.20**	7.90	9.80
		Benign Heterophily	5.67	12.00	7.33	6.67	8.00	5.75	**2.83**	9.25	**3.17**	7.17	7.50	9.83	**2.33**
		Ambiguous Heterophily	5.80	9.80	4.50	6.00	6.00	**1.25**	**2.60**	11.00	8.00	4.00	6.33	8.40	**3.20**
		Homophily	5.67	11.33	8.00	11.33	**3.50**	7.50	**5.17**	5.50	**4.17**	5.83	6.50	7.83	8.67

Table 2: Re-evaluation and comparison of 11 SOTA models on different categories of datasets. The result or ranking is marked by red if it is worse than the best baseline models (GCN, SGC-1, MLP-2, MLP-1); the first, second and third best average ranking is marked by green, blue, orange. OOM is short for out-of-memory.

2.3 Reassessment of State-of-the-arts Models

Based on the new categorization of heterophilic datasets, our next question is: **how do current SOTA models perform on different groups of datasets, and are they really effective on heterophily?** In this subsection, we reassess 11 popular SOTA GNNs, covering six most popular methods for heterophily, with fine-tuned hyperparameters.[3] The methods and models include: ego-neighbor separation (H_2GCN [74]), negative message passing (GPRGNN [9], FAGCN [5], GloGNN* [31] [4]), high-pass filter with adaptive channel mixing (ACM-GCN* [39][5]), selective message passing (GBK-GNN [11], FSGNN [48]), spectral GNN (JacobConv [62], BernNet [23]), multi-hop GNN (APPNP [17], LINKX [33]). A model is considered effective for heterophily if it: 1) outperforms the best baseline model on malignant and ambiguous heterophily graphs, and 2) performs at least as good as the best baseline model in other categories. According to Table 2, we find,

- **Effectiveness of Heterophily-specific Methods:** In most tasks, the majority of SOTA GNNs do not significantly outperform the best baseline models. ACM-GCN*, FSGNN and

[2]This is consistent with the conclusions in [39, 40, 44]

[3]See Appendix A.4 for the hyperparameter searching range.

[4]GloGNN* indicates the best results of the variants of GloGNN.

[5]ACM-GCN has lots of variants, we report the best results of them as ACM-GCN*.

GloGNN* are the only three models that have better overall average ranking than the best baseline models. This means only high-pass filter with adaptive channel mixing, selective message passing and negative message passing methods are verified to have the potential to be effective for heterophily.

- **Detailed Analysis:** Besides the aforementioned three methods, others are only partially effective on one of the heterophily categories, but have bad overall rankings. For example, H_2GCN, JacobConv and LINKX perform well on ambiguous heterophily, BernNet is good at malignant heterophily. However, they still underperform the best baselines in general. This demonstrates the necessity of comprehensive evaluation for each category. Otherwise, the conclusions about effectiveness can be questionable and unreliable. Particularly, for multi-hop GNNs, LINKX (hop=2) works well for ambiguous and benign heterophily, but APPNP (hop=10) struggles on them. This echoes the over-globalizing phenomenon [64, 66], where the model focuses too much on distant nodes, but the long-range dependencies are found not to be quite informative for heterophilic graphs [64, 66]. In other words, we need to be fairly selective for the diffusion scopes of our model [36].
- **Imbalanced Performance:** Some GNNs sacrifice their abilities on easy (homophily or benign heterophily) graphs to gain performance enhancement on difficult (malignant or ambiguous heterophily) graphs, *e.g.,* every model with negative message passing yield subpar results on benign heterophily, H_2GCN, GPRGNN, GloGNN*, spectral GNNs and multi-hop GNNs show inferior results on homophily datasets. Such imbalanced performance implies that they are not universally effective methods. Surprisingly, GBK-GNN only performs well on homophily datasets, and such results is caused by the unrigorous and insufficient evaluation in the original paper.
- **Scalability Issue:** Some of the tested GNNs suffer from severe out-of-memory (OOM) problem, *e.g.,* GloGNN, GBK-GNN and FSGNN, which indicates that some heterophily-specific methods encounter scalability issue.

3 Quantitative Evaluation of Homophily Metrics on Synthetic Graphs

Homophily metrics are proposed to help people recognize the challenging heterophilic datasets without training models [40], and people usually evaluate the metrics by synthetic graphs (we introduce three most widely used synthetic graph generation methods, *i.e.,* regular graphs, preferential attachment and GenCat, in Appendix A.6). In Section 3.1, we unify the evaluation methods, compare 11 popular homophily metrics on synthetic graphs and illustrate the challenges of the observation-based evaluation approach; to compare the metrics strictly, in Section 3.2, we conduct quantitative comparisons among metrics based on Pearson correlation and Fréchet distance and provide detailed analysis.

3.1 Evaluation of Metrics and Observation-Based Comparison

Evaluation Process It includes the following steps: 1) generate synthetic graphs with different homophily-related hyperparameters, *e.g.,* H_{edge} for regular graphs, μ for PA model and β for GenCat; 2) split nodes randomly into 60%/20%/20% for train/validation/test; 3) each baseline model (GCN, SGC-1, MLP-2 and MLP-1) is trained on every synthetic graph with the same hyperparameter searching range as [39], the mean test accuracy and standard deviation of 10 runs are recorded; 4) calculate the corresponding metric values for each synthetic graph; 5) plot the metric curves and the model performance curves *w.r.t.* the homophily-related hyperparameters, observe their relations.

The model performance curves and the metric curves are shown in Figure 1 (a)(b)(c) and Figure 1 (d)(e)(f). A metric is considered superior to another if the shape of its curve aligns more closely with GNN performance curves.

Challenges of Observation-based Comparison It is hard to justify which metric is superior based merely on observation. For example, in PA generated graphs, GNN performance exhibits a bit of bouncing back in low-homophily area. However, H_{agg}, H_{neighbor} and LI all exhibit such shape and it is difficult to determine which is better. Therefore, quantitative evaluation is required for strict and reliable comparison and we will provide the first quantitative evaluation on homophily metrics in the next subsection.

3.2 Quantitative Evaluation for Homophily Metrics

Metrics/Graphs	RG				PA				GenCat				Average Ranking					
	Pearson		Fréchet		Pearson		Fréchet		Pearson		Fréchet		RG		PA		GenCat	
	GCN	SGC	GCN	SGC	GCN	SGC	GCN	SGC	GCN	SGC	GCN	SGC	Pearson	Fréchet	Pearson	Fréchet	Pearson	Fréchet
H_{edge}	0.75	0.78	0.53	0.60	0.86	0.87	0.19	0.17	0.99	0.99	0.27	0.21	8.00	4.00	5.00	3.00	**1.50**	5.00
H_{node}	0.76	0.79	0.51	0.58	0.86	0.87	0.19	0.17	0.99	0.99	0.27	0.21	6.00	2.00	5.00	4.00	**1.50**	4.00
H_{class}	0.62	0.69	0.63	0.69	0.94	0.95	0.07	0.15	1.00	0.98	0.15	0.21	9.50	6.50	3.00	**1.00**	2.50	2.50
H_{adj}	0.75	0.78	0.76	0.83	0.86	0.87	0.19	0.17	0.99	0.99	0.27	0.21	7.00	11.00	5.00	2.00	**1.50**	4.50
H_{GE}	0.26	0.30	0.64	0.65	0.87	0.88	0.94	0.92	0.42	0.46	0.26	0.21	11.00	7.50	4.00	11.00	11.00	3.00
H_{agg}	0.82	0.83	0.37	0.30	0.60	0.64	0.51	0.46	0.84	0.90	0.42	0.40	5.00	**1.00**	10.00	7.00	7.00	9.50
LI	0.62	0.69	0.61	0.68	0.99	0.98	0.28	0.34	0.95	0.91	0.14	0.12	9.50	5.50	**1.00**	5.00	5.00	**1.00**
$H_{neighbor}$	0.88	0.87	0.51	0.58	0.97	0.96	0.50	0.50	0.87	0.82	0.26	0.18	2.50	3.00	2.00	7.50	7.00	3.00
GNB	0.91	0.85	0.63	0.70	0.80	0.78	0.48	0.50	0.87	0.81	0.38	0.48	**1.50**	7.50	9.00	6.50	8.00	9.50
KR_L	0.91	0.85	0.63	0.70	-0.23	-0.22	0.80	0.77	0.69	0.77	0.88	0.79	**1.50**	7.50	11.00	10.00	10.00	11.00
KR_{NL}	0.91	0.85	0.63	0.70	0.83	0.84	0.67	0.60	0.74	0.82	0.36	0.27	**1.50**	7.50	8.00	9.00	8.00	8.00

Table 3: Quantitative comparison of homophily metrics on synthetic graphs. Cells marked by green and red are the best and worst ranked metrics.

To compare the metrics quantitatively, we measure the similarity between the metric and GNN (GCN and SGC) curves by Pearson correlation and Fréchet distance. Pearson correlation [51] measures linear correlation between two sets of data; Fréchet distance measures the similarity between two arbitrary curves and it can be approximately calculated by the discrete Fréchet distance[6] [2, 12]. A smaller distance value indicates a higher similarity. We calculate the average ranking of the homophily metrics *w.r.t.* graph generation methods and similarity metrics. From the results in Table 3 we can see that,

- **Classic Metrics Are Still Strong** Although some new proposed metrics can reveal the rebounding phenomenon in extremely low homophily area, *e.g.,* H_{agg}, the old homophily metrics, *e.g.,*H_{edge}, H_{node}, H_{class}, still show very strong overall performance among different scenarios. On the other hand, many new metrics are unstable and exhibit unsatisfactory results in some cases, *e.g.,* H_{agg}, LI and the hypothesis testing based metrics.
- **Results Highly Depends on Similarity Measurements** For example, in RG generated graphs, the hypothesis testing based metrics show strong results with Pearson correlation but poor results with Fréchet distance. This is because Pearson correlation does not consider the actual spatial distance between points, but Fréchet distance considers spatial arrangement of points. The hypothesis testing based metrics are mainly designed to find out the threshold value (p-value) for good and bad graphs. Therefore, they might only capture the correlation instead of the spatial position of GNN curves. In the future, if people aim to design metrics with threshold values, we suggest using Pearson correlation as similarity measurement.
- **Discrepancy Between Different Synthetic Graphs** Given the same similarity measurement, the average ranking on different synthetic graphs can give quite different conclusions. For example, with Pearson correlation, KR_L rank the best in RG generated graphs, but rank the worst in PA generated graphs. Similar problem happens to H_{agg}, LI and GNB. In the future, to get a more reliable conclusion, we suggest people to take a more comprehensive view based on results from all the three synthetic graphs instead on just one of them.

4 Conclusion and Limitation

In this paper, we reveal and overcome three pitfalls in the model and metric evaluation for heterophilic graph representation learning: 1) lack of hyperparameter tuning; 2) insufficient model evaluation on the real challenging heterophilic datasets; 3) absence of quantitative evaluation benchmark for homophily metrics on synthetic graphs. We fine-tune the baseline models to discover three different types of heterophilic graphs among 27 mostly used benchmark datasets, *i.e.,* malignant, benign and ambiguous heterophily datasets. We identify the real challenging tasks and reassess 11 popular SOTA models with fine-tuned hyperparameters. At last, we conduct quantitative evaluation for 11 popular metrics based on Pearson correlation and Fréchet distance and analyze performance.

In the future, more research on ambiguous heterophily datasets is necessary for understanding the synergy of graph structure and model nonlinearity. A corresponding homophily metrics can be designed based on such synergy.

[6]We use the Python implementation for the calculation of discrete Fréchet distance provided by [13]. The code is from `https://pypi.org/project/frechetdist/`.

References

[1] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pages 21–29. PMLR, 2019.

[2] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.

[3] S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

[4] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[5] D. Bo, X. Wang, C. Shi, and H. Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3950–3957, 2021.

[6] P. Bongini, M. Bianchini, and F. Scarselli. Molecular generative graph neural networks for drug discovery. *Neurocomputing*, 450:242–252, 2021.

[7] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and deep locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[8] J. Chen, S. Chen, J. Gao, Z. Huang, J. Zhang, and J. Pu. Exploiting neighbor effect: Conv-agnostic gnn framework for graphs with heterophily. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[9] E. Chien, J. Peng, P. Li, and O. Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2021.

[10] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[11] L. Du, X. Shi, Q. Fu, X. Ma, H. Liu, S. Han, and D. Zhang. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proceedings of the ACM Web Conference 2022*, pages 1550–1558, 2022.

[12] Efrat, Guibas, S. Har-Peled, and Murali. New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete & Computational Geometry*, 28:535–569, 2002.

[13] T. Eiter and H. Mannila. Computing discrete fréchet distance. 1994.

[14] M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[15] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE transactions on Neural Networks*, 9(5):768–786, 1998.

[16] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison. Deep convolutional networks as shallow gaussian processes. In *International Conference on Learning Representations*, 2018.

[17] J. Gasteiger, A. Bojchevski, and S. Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2018.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[19] C. Goller and A. Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 347–352. IEEE, 1996.

[20] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[21] W. L. Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[22] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[23] M. He, Z. Wei, H. Xu, et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34, 2021.

[24] C. Hua, Y. Liu, D. Zhang, O. Zhang, S. Luan, K. K. Yang, G. Wolf, D. Precup, and S. Zheng. Enzymeflow: Generating reaction-specific enzyme catalytic pockets through flow matching and co-evolutionary dynamics. *arXiv preprint arXiv:2410.00327*, 2024.

[25] C. Hua, S. Luan, M. Xu, Z. Ying, J. Fu, S. Ermon, and D. Precup. Mudiff: Unified diffusion for complete molecule generation. In *Learning on Graphs Conference*, pages 33–1. PMLR, 2024.

[26] C. Hua, B. Zhong, S. Luan, L. Hong, G. Wolf, D. Precup, and S. Zheng. Reactzyme: A benchmark for enzyme-reaction prediction. *Advances in Neural Information Processing Systems*, 37:26415–26442, 2024.

[27] D. Jin, R. Wang, M. Ge, D. He, X. Li, W. Lin, and W. Zhang. Raw-gnn: Random walk aggregation based graph neural network. In *31st International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 2108–2114. International Joint Conferences on Artificial Intelligence, 2022.

[28] F. Karimi, M. Génois, C. Wagner, P. Singer, and M. Strohmaier. Homophily influences ranking of minorities in social networks. *Scientific reports*, 8(1):11077, 2018.

[29] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

[30] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

[31] X. Li, R. Zhu, Y. Cheng, C. Shan, S. Luo, D. Li, and W. Qian. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning*, pages 13242–13256. PMLR, 2022.

[32] D. Lim, F. Hohne, X. Li, S. L. Huang, V. Gupta, O. Bhalerao, and S. N. Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems*, 34:20887–20902, 2021.

[33] D. Lim, X. Li, F. Hohne, and S.-N. Lim. New benchmarks for learning on non-homophilous graphs. *arXiv preprint arXiv:2104.01404*, 2021.

[34] C. Lozares, J. M. Verd, I. Cruz, and O. Barranco. Homophily and heterophily in personal networks. from mutual acquaintance to relationship intensity. *Quality & Quantity*, 48:2657–2670, 2014.

[35] Q. Lu, S. Luan, and X.-W. Chang. Gcepnet: Graph convolution-enhanced expectation propagation for massive mimo detection. *arXiv preprint arXiv:2404.14886*, 2024.

[36] Q. Lu, J. Zhu, S. Luan, and X.-W. Chang. Flexible diffusion scopes with parameterized laplacian for heterophilic graph learning. In *The Third Learning on Graphs Conference*, 2024.

[37] S. Luan, C. Hua, Q. Lu, L. Ma, L. Wu, X. Wang, M. Xu, X.-W. Chang, D. Precup, R. Ying, et al. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618*, 2024.

[38] S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

[39] S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375, 2022.

[40] S. Luan, C. Hua, M. Xu, Q. Lu, J. Zhu, X.-W. Chang, J. Fu, J. Leskovec, and D. Precup. When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability. *Advances in Neural Information Processing Systems*, 36:28748–28760, 2023.

[41] S. Luan, M. Zhao, X.-W. Chang, and D. Precup. Break the ceiling: Stronger multi-scale deep graph convolutional networks. *Advances in neural information processing systems*, 32, 2019.

[42] S. Luan, M. Zhao, C. Hua, X.-W. Chang, and D. Precup. Complete the missing half: Augmenting aggregation filtering with diversification for graph convolutional networks. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*, 2022.

[43] Y. Luo, L. Shi, and X.-M. Wu. Classic gnns are strong baselines: Reassessing gnns for node classification. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[44] Y. Ma, X. Liu, N. Shah, and J. Tang. Is homophily a necessity for graph neural networks? In *International Conference on Learning Representations*, 2021.

[45] S. Maekawa, K. Noda, Y. Sasaki, et al. Beyond real-world benchmark datasets: An empirical study of node classification with gnns. *Advances in Neural Information Processing Systems*, 35:5562–5574, 2022.

[46] S. Maekawa, Y. Sasaki, G. Fletcher, and M. Onizuka. Gencat: Generating attributed graphs with controlled relationships between classes, attributes, and topology. *Information Systems*, 115:102195, 2023.

[47] A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *In International Conference on Learning Representations*, 2018.

[48] S. K. Maurya, X. Liu, and T. Murata. Simplifying approach to node classification in graph neural networks. *Journal of Computational Science*, 62:101695, 2022.

[49] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[50] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[51] K. Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London Series I*, 58:240–242, Jan. 1895.

[52] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

[53] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2020.

[54] O. Platonov, D. Kuznedelev, A. Babenko, and L. Prokhorenkova. Characterizing graph datasets for node classification: Beyond homophily-heterophily dichotomy. *Advances in Neural Information Processing Systems*, 36, 2023.

[55] O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*, 2022.

[56] B. Rozemberczki, C. Allen, and R. Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[57] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[58] A. Sperduti. Encoding labeled graphs by labeling raam. *Advances in Neural Information Processing Systems*, 6, 1993.

[59] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[60] Y. Sun, H. Deng, Y. Yang, C. Wang, J. Xu, R. Huang, L. Cao, Y. Wang, and L. Chen. Beyond homophily: structure-aware path aggregation graph neural network. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pages 2233–2240, 2022.

[61] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[62] X. Wang and M. Zhang. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*, pages 23341–23362. PMLR, 2022.

[63] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[64] Y. Xing, X. Wang, Y. Li, H. Huang, and C. Shi. Less is more: on the over-globalizing problem in graph transformers. In *International Conference on Machine Learning*, pages 54656–54672. PMLR, 2024.

[65] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.

[66] N. Yadati. Localformer: Mitigating over-globalising in transformers on graphs with localised training. *Transactions on Machine Learning Research*, 2025.

[67] Z. Yang, W. Cohen, and R. Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[68] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

[69] M. Zhao, Z. Liu, S. Luan, S. Zhang, D. Precup, and Y. Bengio. A consciousness-inspired planning agent for model-based reinforcement learning. *Advances in neural information processing systems*, 34:1569–1581, 2021.

[70] Y. Zheng, X. Li, S. Luan, X. Peng, and L. Chen. Let your features tell the differences: Understanding graph convolution by feature splitting. In *The Thirteenth International Conference on Learning Representations*, 2025.

[71] Y. Zheng, S. Luan, and L. Chen. What is missing for graph homophily? disentangling graph homophily for graph neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[72] Y. Zheng, Z. Zhang, Z. Wang, X. Li, S. Luan, X. Peng, and L. Chen. Rethinking structure learning for graph neural networks. *arXiv preprint arXiv:2411.07672*, 2024.

[73] Z. Zhou, S. Zhou, B. Mao, X. Zhou, J. Chen, Q. Tan, D. Zha, Y. Feng, C. Chen, and C. Wang. Opengsl: A comprehensive benchmark for graph structure learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[74] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33, 2020.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: last section

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We put it in the main paper

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We have carefully reevaluated our project and we do not have such negative impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We have carefully reevaluated our project and we do not have such risk.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: We do not have such problem

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: We have carefully reevaluated our project and we do not have such risk.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM only for grammar check.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Preliminaries

A.1 Notation

We define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, N\}$ is the set of nodes and $\mathcal{E} = \{e_{ij}\}$ is the set of edges without self-loops. The adjacency matrix of $\mathcal{G}$ is denoted by $A = (A_{i,j}) \in \mathbb{R}^{N\times N}$ with $A_{i,j} = 1$ if there is an edge between nodes i and j, otherwise $A_{i,j} = 0$. The diagonal degree matrix of $\mathcal{G}$ is denoted by D with $D_{i,i} = d_i = \sum_j A_{i,j}$. The neighborhood set $\mathcal{N}_i$ of node i is defined as $\mathcal{N}_i = \{j : e_{ij} \in \mathcal{E}\}$. A graph signal is a vector in $\mathbb{R}^N$, whose i-th entry is a feature of node i. Additionally, we use $X \in \mathbb{R}^{N\times F_h}$ to denote the feature matrix, whose columns are graph signals and i-th row $X_{i,:} = \boldsymbol{x}_i^\top$ is the feature vector of node i (we use **bold** font for vectors). The label encoding matrix is $Y \in \mathbb{R}^{N\times C}$, where C is the number of classes, and its i-th row $Y_{i,:}$ is the one-hot encoding of the label of node i. We denote $y_i = \arg\max_j Y_{i,j} \in \{1, 2, \ldots C\}$. The indicator function $\mathbf{1}_B$ equals 1 when event B happens and 0 otherwise.

For nodes $i, j \in \mathcal{V}$, if $y_i = y_j$, they are termed *intra-class nodes*; if $y_i \neq y_j$, they are termed *inter-class nodes*. Similarly, an edge $e_{i,j} \in \mathcal{E}$ is termed an *intra-class edge* if $y_i = y_j$, and an *inter-class edge* if $y_i \neq y_j$.

The affinity matrices can be derived from the adjacency matrix, *e.g.,* $A_{\text{rw}} = D^{-1}A$ and $A_{\text{sym}} = D^{-1/2}AD^{-1/2}$. After applying the renormalization trick [29], we have $\hat{A}_{\text{sym}} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ and $\hat{A}_{\text{rw}} = \tilde{D}^{-1}\tilde{A}$, where $\tilde{A} \equiv A + I$ and $\tilde{D} \equiv D + I$. The renormalized affinity matrix essentially adds a self-loop to each node. The affinity matrices are commonly used as aggregation operators in GNNs.

A.2 Graph-aware Models and Graph-agnostic Models

A network that incorporates feature aggregation based on graph structure is referred to as a graph-aware model [40], *e.g.,* GCN [29], SGC [63]; and a network that does not use graph structure information in each layer is called graph-agnostic model, such as MLP-2 (Multi-Layer Perceptron with 2 layers) and MLP-1. A graph-aware model is always coupled with a graph-agnostic model, as when the aggregation step is removed, the graph-aware model becomes exactly the same as its coupled graph-agnostic model, *e.g.,* GCN is coupled with MLP-2 and SGC-1 is coupled with MLP-1 as shown below:

$$\begin{aligned}&\textbf{GCN: } \text{Softmax}(\hat{A}_{\text{sym}}\ \text{ReLU}(\hat{A}_{\text{sym}}XW_0)\ W_1),\ \ \textbf{MLP-2: } \text{Softmax}(\text{ReLU}(XW_0)\ W_1),\\ &\textbf{SGC-1: } \text{Softmax}(\hat{A}_{\text{sym}}XW_0),\ \ \textbf{MLP-1: } \text{Softmax}(XW_0),\end{aligned} \tag{1}$$

where $W_0 \in \mathbb{R}^{F_0\times F_1}$ and $W_1 \in \mathbb{R}^{F_1\times O}$ are learnable parameter matrices. A node classification task on graph is considered as real challenging if a graph-aware model underperforms its coupled graph-agnostic counterpart on it [40]. Numerous homophily metrics have been proposed to recognize the difficult graphs and the most commonly used ones will be introduced in the next subsection.

A.3 Homophily Metrics

There are mainly four ways to define homophily metrics [37]. We will introduce their calculations briefly in this subsection.

Graph-Label Consistency There are four commonly used homophily metrics that are based on the consistency between node labels and graph structures, including edge homophily [1, 74], node homophily [52], class homophily [33] and adjusted homophily [54], defined as follows:

$$\begin{aligned}&\mathrm{H}_{\text{edge}}(\mathcal{G}) = \frac{\left|\{e_{uv} \mid e_{uv} \in \mathcal{E}, y_u = y_v\}\right|}{|\mathcal{E}|};\ \ \mathrm{H}_{\text{node}}(\mathcal{G}) = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\frac{\left|\{u \mid u \in \mathcal{N}_v, y_u = y_v\}\right|}{d_v};\\ &\mathrm{H}_{\text{class}}(\mathcal{G}) = \frac{1}{C-1}\sum_{k=1}^{C}\left[h_k - \frac{\left|\{v \mid Y_{v,k}=1\}\right|}{N}\right]_+,\ \ h_k = \frac{\sum_{v\in\mathcal{V}, Y_{v,k}=1}\left|\{u \mid u \in \mathcal{N}_v, y_u = y_v\}\right|}{\sum_{v\in\{v\mid Y_{v,k}=1\}} d_v};\\ &\mathrm{H}_{\text{adj}}(\mathcal{G}) = \frac{\mathrm{H}_{\text{edge}} - \sum_{c=1}^{C}\bar{p}_c^2}{1 - \sum_{c=1}^{C}\bar{p}_c^2},\ \ \bar{p}_c = \frac{\sum_{v:y_v=c} d_v}{2|\mathcal{E}|},\end{aligned} \tag{2}$$

where $[a]_+ = \max(a, 0)$, h_k is the class-wise homophily metric [33].

Similarity-Based Metrics Generalized edge homophily [27] and aggregation homophily [39] leverages similarity functions to define the metrics:

$$\mathrm{H_{GE}}(\mathcal{G}) = \frac{\sum_{(i,j)\in\mathcal{E}} \cos(\boldsymbol{x}_i, \boldsymbol{x}_j)}{|\mathcal{E}|};\ \mathrm{H_{agg}}(\mathcal{G}) = \frac{1}{|\mathcal{V}|} \times \left|\left\{v \,\middle|\, \mathrm{Mean}_u\big(\{S(\hat{A},Y)_{v,u}^{y_u=y_v}\}\big) \geq \mathrm{Mean}_u\big(\{S(\hat{A},Y)_{v,u}^{y_u\neq y_v}\}\big)\right\}\right|, \quad (3)$$

where $\mathrm{Mean}_u\left(\{\cdot\}\right)$ takes the average over u of a given multiset of values or variables and $S(\hat{A},Y) = \hat{A}Y(\hat{A}Y)^\top$ is the post-aggregation node similarity matrix. These two metrics are feature-dependent.

Neighborhood Identifiability/Informativeness Label informativeness [54] and neighborhood identifiability [8] use the neighbor distribution instead of pairwise comparison to define the metrics:

$$\mathrm{LI} = 2 - \frac{\sum_{c_1,c_2} p_{c_1,c_2} \ln \frac{p_{c_1,c_2}}{\bar{p}_{c_1}\bar{p}_{c_2}}}{\sum_c \bar{p}_c \log \bar{p}_c};\ \mathrm{H_{neighbor}}\left(\mathcal{G}\right) = \sum_{k=1}^{C} \frac{n_k}{N} \mathrm{H}^k_{\text{neighbor}},\ \mathrm{H}^k_{\text{neighbor}} = \frac{-\sum_{i=1}^{C} \tilde{\sigma}_i^k \ln\big(\tilde{\sigma}_i^k\big)}{\ln(C)} \quad (4)$$

where $p_{c_1,c_2} = \sum_{(u,v)\in\mathcal{E}} \frac{\mathbf{1}\{y_u=c_1, y_v=c_2\}}{2|\mathcal{E}|}$ for $c_1, c_2 \in \{1,\ldots,C\}$; n_k is the number of nodes with the label k; and $\tilde{\sigma}_i^k$ will be defined immediately. Let $A^k \in \mathbb{R}^{n_k\times C}$ be a class-level neighborhood label distribution matrix for class $k = 1,\ldots,C$, *i.e.,* for a node i from class k, $(A^k)_{i,c}$ is the proportion of the neighbors of node i belonging to class c, and let $\sigma_1^k, \sigma_2^k, \ldots, \sigma_C^k$ denote the singular values of A^k, and they are normalized such that $\sum_{c=1}^{C} \tilde{\sigma}_c^k = 1$, i.e., $\tilde{\sigma}_c^k = \sigma_c^k / \sum_{c=1}^{C} \sigma_c^k$.

Hypothesis Testing Based Performance Metrics Classifier-based performance metric (CPM) [40] uses the p-value of the following hypothesis testing as a metric to measure the node distinguishability of the aggregated features H compared with the original features X.

$$\mathrm{H}_0 : \mathrm{Acc}(\mathrm{Classifier}(H)) \geq \mathrm{Acc}(\mathrm{Classifier}(X));\ \mathrm{H}_1 : \mathrm{Acc}(\mathrm{Classifier}(H)) < \mathrm{Acc}(\mathrm{Classifier}(X)), \quad (5)$$

where Acc is the prediction accuracy of the given classifier. To capture the feature-based linear or non-linear information efficiently, Luan *et al.* [40] choose Gaussian Naïve Bayes (GNB) [22] and Kernel Regression (KR) with Neural Network Gaussian Process (NNGP) [3, 16, 30, 47] as the classifiers, which do not require iterative training.

Overall, $\mathrm{H_{adj}}$ can assume negative values, while other metrics all fall within the range of $[0,1]$. Except for $\mathrm{H_{neighbor}}(\mathcal{G})$, where a smaller value indicates more identifiable,[7] the other metrics with higher values indicate strong homophily, implying that graph-aware models are more likely to outperform their coupled graph-agnostic model, and vice versa. These metrics will be compared in Section 3.

A.4 Hyperparameter Searching Range

For every models and datasets, we perform a grid search for learning rate $\in \{0.01, 0.05, 0.1\}$, weight decay $\in \{0, 5e-7, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2\}$, dropout $\in \{0, 0.1, 0.3, 0.5, 0.7\}$ with the Adam optimizer. We use hidden unit $= 128$ for wiki-cooc, 512 for roman-empire, amazon-ratings, minesweeper, tolokers, questions, Squirrel-filtered, Chameleon-filtered, and 64 for all the other datasets. These settings are used for GCN, SGC-1, MLP-2, MLP-1, and shared by other GNN models. Specific hyperparameters for are listed as follows

- GPRGNN: the weight is initialized by their Personalized PageRank, $\alpha \in \{0.1, 0.2, 0.5, 0.9\}$ and $K = 10$ power of the adjacency is used.
- BernNet: the propagation steps $K = 10$.
- FAGCN: $\epsilon \in \{0.3, 0.4, 0.5\}$
- LINKX: the number of layers of $\mathrm{MLP}_{\mathbf{A}}$ and $\mathrm{MLP}_{\mathbf{X}}$ are in $\{1, 2\}$.
- ACM-GCN: "structure_info" $\in \{0, 1\}$, "variant" $\in \{0, 1\}$, with "ACM-GCN+" and "ACM-GCN++".
- GBK-GNN: we set $\lambda = 30$ and use the model based on GraphSage.
- FSGNN: 3-hop configuration under "all-feature" settings.
- APPNP: $\alpha \in \{0.1, 0.2, 0.5, 0.9\}$ and $K = 10$ power of the adjacency is used.

Datasets/Models	GCN		MLP-2		SGC-1		MLP-1	
	w/o	w	w/o	w	w/o	w	w/o	w
Squirrel-filtered	30.35 ± 1.71	37.33 ± 1.88	26.20 ± 1.46	38.30 ± 1.22	30.81 ± 1.69	37.54 ± 2.13	28.78 ± 1.38	30.14 ± 1.53
Chameleon-filtered	39.39 ± 3.81	41.46 ± 3.42	29.24 ± 3.17	38.06 ± 3.98	37.64 ± 2.95	44.00 ± 3.10	29.54 ± 3.77	35.72 ± 2.23
roman-empire	41.77 ± 0.51	48.92 ± 0.46	65.14 ± 0.60	66.04 ± 0.71	28.48 ± 1.01	44.60 ± 0.52	52.67 ± 1.41	64.12 ± 0.61
amazon-ratings	45.28 ± 0.77	50.05 ± 0.67	43.13 ± 0.95	49.55 ± 0.81	38.00 ± 0.64	40.69 ± 0.42	36.46 ± 0.58	38.60 ± 0.41
minesweeper	71.73 ± 1.09	72.34 ± 0.93	50.10 ± 0.84	50.92 ± 1.25	49.54 ± 18.79	82.04 ± 0.77	49.88 ± 1.30	50.59 ± 0.83
tolokers	63.74 ± 3.30	77.44 ± 1.32	70.73 ± 1.07	74.58 ± 0.75	46.91 ± 15.67	73.80 ± 1.35	45.64 ± 11.00	71.89 ± 0.82
questions	55.21 ± 1.52	72.72 ± 1.93	70.95 ± 1.20	69.97 ± 1.16	51.59 ± 3.97	70.06 ± 0.92	51.70 ± 3.14	70.33 ± 0.96

Table 4: Comparison of baseline models with (w) and without (w/o) hyperparameter tuning. The results are highlighted in red if hyperparameter tuning significantly improve the model performance. The un-tuned models use the hyperparameters provided in [55].

A.5 Hyperparameter Fine-Tuning for Fair and Reliable Comparison

To demonstrate the importance of hyperparameter fine-tuning, following [40], we first fine-tune two baseline GNNs, GCN [29] and 1-hop SGC (SGC-1) [63], and their coupled graph-agnostic models, MLP-2 and MLP-1 [8] on the datasets where the baseline models are claimed to be quite robust to hyperparameter values [55]. From the experimental results shown in Table 4, we have identified a serious pitfall for model evaluation on heterophilic datasets, *i.e.,* there exits a huge discrepancy between the model performance with and without (w/o) hyperparameter fine-tuning, even on the 'hyperparameter-robust' datasets. In Table 4, we can see that in 19 out of 28 cases, hyperparameter fine-tuning can significantly improve model performance. This implies that a large amount of reported results in existing literature are potentially unreliable if there is no fine-tuning or the hyperparameter searching range is not large enough. This pitfall significantly hinders the fair model comparison and disrupt our way to discover the real challenging heterophilic datasets and really effective models. (See Appendix A.4 for our hyperparameter searching range.)

A.6 Generation Methods for Synthetic Graphs

There are mainly three ways to generate synthetic graphs for homophily metric evaluation.

Regular Graph (RG) Luan *et al.* [39] proposed to generate regular graphs as follows: 1) 10 graphs are generated for each of the 28 edge homophily levels, from 0.005 to 0.95, with a total of 280 graphs; 2) Every generated graph has five classes, with 400 nodes in each class. For nodes in each class, 800 random intra-class edges and $[\frac{800}{\mathrm{H}_{\text{edge}}(\mathcal{G})} - 800]$ inter-class edges are uniformly generated ; 3) The features of nodes in each class are sampled from node features in the corresponding class of the base datasets, *e.g.,* Figure 1 (a)(d) are based on the node features from *Cora*.

Preferential Attachment (PA) [4] Karimi *et al.* [28] incorporate homophily as an additional parameter to Preferential Attachment (PA) model and Abu-El-Haija *et al.* [1] extend it to multi-class settings, which is widely used in graph machine learning community. The process are as follows.

Suppose graph $\mathcal{G}$ has a total number of N nodes, C classes, and a homophily coefficient μ, the generation begins by dividing the N nodes into C equal-sized classes. Then, $\mathcal{G}$ (initially empty) is updated iteratively. At each step, a new node v_i is added, and its class y_i is randomly assigned from the set $\{1, \ldots, C\}$. Whenever a new node v_i is added to $\mathcal{G}$, a connection between v_i and an existing node v_j in $\mathcal{G}$ is established based on the probability $\bar{p}_{ij}$, which is calculated as follows,

$$p_{ij} = \begin{cases} d_j \times \mu, & \text{if } y_i = y_j \\ d_j \times (1-\mu) \times w_{d(y_i,y_j)}, & \text{otherwise} \end{cases}, \text{ and } \bar{p}_{ij} = \frac{p_{ij}}{\sum_{k:v_k \in \mathcal{N}(v_i)} p_{ik}} \tag{6}$$

where y_i and y_j are class labels of node i and j respectively, and $w_{d(y_i,y_j)}$ [9] denotes the "cost" of connecting nodes from two distinct classes with a class distance of $d(y_i, y_j)$. [10] The weight

[7] To compare with other metrics more easily, in this paper, we use $1 - \mathrm{H}_{\text{neighbor}}(\mathcal{G})$ for quantitative analysis.

[8] Note that for fair evaluation, we remove all tricks in model architectures, such as residual connection and batch normalization, and only keep the original models for tests.

[9] The code for calculating $w_{d(y_i,y_j)}$ is not open-sourced and we obtain the code from the authors of [1].

[10] The distance between two classes simply implies the shortest distance between the two classes on a circle where classes are numbered from 1 to C. For instance, if $C = 6$, $y_i = 1$ and $y_j = 5$, then the distance between y_i and y_j is 2.

exponentially decreases as the distance increases and is normalized such that $\sum_d w_d = 1$. For a larger μ, the chance of connecting with a node with the same label increases. Lastly, the features of each node in the output graph are sampled from overlapping 2D Gaussian distributions. Each class has its own distribution defined separately.

GenCat GenCat [45, 46] generates synthetic graphs based on a real-world graph and a hyper-parameter β controlling the homophily/heterophily property of the generated graph. According to base graph and β, class preference mean $M^{(\beta)} \in \mathbb{R}^{C\times C}$, class preference deviation $D^{(\beta)} \in \mathbb{R}^{C\times C}$, class size distribution and attribute-class correlation $H \in \mathbb{R}^{F\times C}$ are calculated, which are then used to create three latent factors: node-class membership proportions $U \in [0,1]^{N\times C}$, node-class connection proportions $U' \in [0,1]^{N\times C}$, and attribute-class proportions $V \in [0,1]^{F\times C}$, where C, F and N are the numbers of classes, features and nodes of the base graph, respectively. Finally, the synthetic graph is generated using these latent factors.

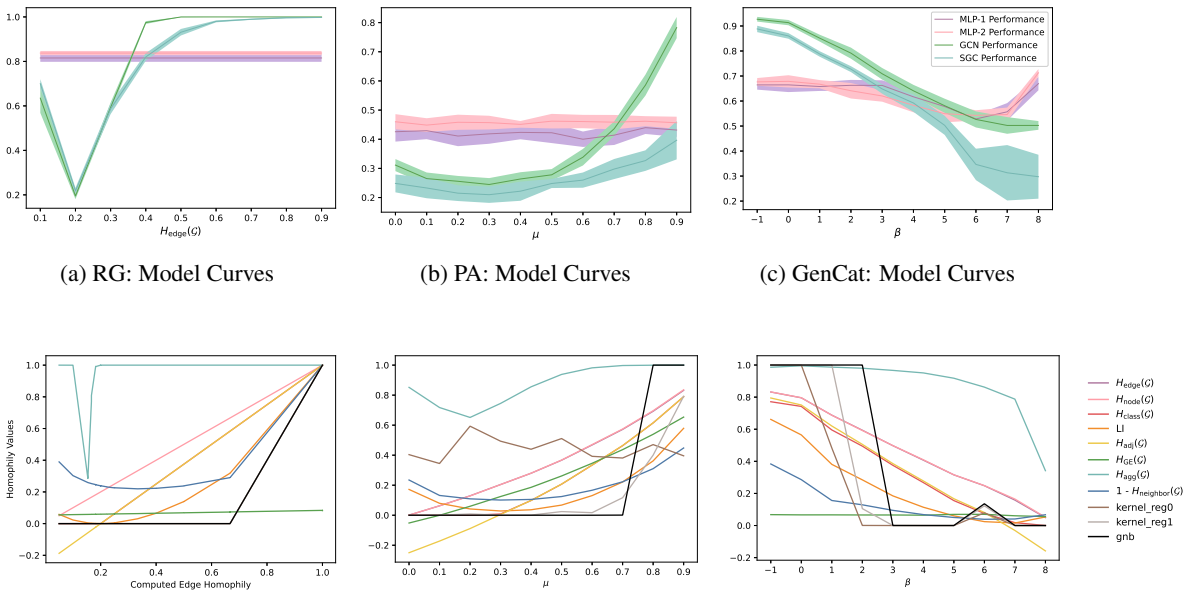
The class preference mean between class c_1 and class c_2 is initially calculated as:

$$M_{c_1,c_2} = \frac{1}{|\Omega_{c_1}|} \sum_{i\in\Omega_{c_1}} \left(\sum_{j\in\Omega_{c_2}} A_{ij} / \sum_j A_{ij} \right),$$

where $\Omega_{c_k} = \{v | Z_{v,k} = 1\}$ is the set of nodes in class c_k. Then, M_{c_1,c_2} is adjusted by β as follows,

$$M^{(\beta)}_{c_1,c_2} = \begin{cases} \max(M_{c_1,c_2} - 0.1 * \beta, 0) & (c_1 = c_2) \\ M_{c_1,c_2} + 0.1 * \beta/(C-1) & (c_1 \neq c_2) \end{cases}.$$

For a larger β, fewer edges would be generated later between nodes within the same class, thus corresponding to a more heterophilic graph. The range of β is $\{\lfloor 10M_{\text{avg}} \rfloor - 9, \lfloor 10M_{\text{avg}} \rfloor - 8, \ldots, \lfloor 10M_{\text{avg}} \rfloor\}$. The average of intra-class connections is calculated as $M_{\text{avg}} = \frac{1}{C}\sum_{c_i} M_{c_i,c_i}$.



(a) RG: Model Curves (b) PA: Model Curves (c) GenCat: Model Curves

(d) RG: Metric Curves (e) PA: Metric Curves (f) GenCat: Metric Curves

Figure 1: Comparison of metrics on synthetic graphs with different generation methods. Note that H_{node} overlaps with H_{edge} in Figure (e) and (f). In Figure (e), $H_{\text{class}}(\mathcal{G})$ overlaps with $H_{\text{adj}}(\mathcal{G})$. KR_L, KR_{NL} and GNB overlaps in Figure (d).