New Benchmarks for Learning on Non-Homophilous Graphs

Derek Lim*
dl772@cornell.edu
Cornell University
USA

Xiuyu Li*
xl289@cornell.edu
Cornell University
USA

Felix Hohne*
fmh42@cornell.edu
Cornell University
USA

Ser-Nam Lim
sernam@gmail.com
Facebook AI
USA

ABSTRACT

Much data with graph structures satisfy the principle of homophily, meaning that connected nodes tend to be similar with respect to a specific attribute. As such, ubiquitous datasets for graph machine learning tasks have generally been highly homophilous, rewarding methods that leverage homophily as an inductive bias. Recent work has pointed out this particular focus, as new non-homophilous datasets have been introduced and graph representation learning models better suited for low-homophily settings have been developed. However, these datasets are small and poorly suited to truly testing the effectiveness of new methods in non-homophilous settings. We present a series of improved graph datasets with node label relationships that do not satisfy the homophily principle. Along with this, we introduce a new measure of the presence or absence of homophily that is better suited than existing measures in different regimes. We benchmark a range of simple methods and graph neural networks across our proposed datasets, drawing new insights for further research. Data and codes can be found at https://github.com/CUAI/Non-Homophily-Benchmarks.

KEYWORDS

datasets, graphs, neural networks, homophily

ACM Reference Format:
Derek Lim, Xiuyu Li, Felix Hohne, and Ser-Nam Lim. 2021. New Benchmarks for Learning on Non-Homophilous Graphs. In *Workshop on Graph Learning Benchmarks, WWW '21, April 19–23, 2021, Ljubljana, Slovenia.*. ACM, New York, NY, USA, 10 pages.

1 INTRODUCTION

Various types of real-world data have natural graph structures, in which objects (nodes) and relationships (edges) are encoded as a graph G. As a result, numerous types of models have been developed for machine learning tasks on graph data. Much recent work in graph representation learning [22] has focused on using both node features and graph structure, especially in the general family of graph neural network (GNN) models.

To generate predictions, graph learning methods can leverage complex inductive biases relating to the topology of the graph [7]. One common property of real-world graphs that is often leveraged as a strong inductive bias is *homophily*, which means that connected nodes tend to be similar in certain attributes [4, 38]. Here, we refer to a graph as *homophilous* if connected nodes are much more likely to have the same class label than if edges were independent of labels. In a *non-homophilous graph*, connected nodes are not significantly more likely to have the same class label, and may even be less likely to have the same class label than if edges were independent of labels. For instance, nodes may have little to no preference for connecting to any particular class in aggregate (e.g. in friendship networks where the classes represent gender [4, 33]); or, nodes of a certain class may only be connected to a subset of the other classes (e.g. in citation networks where class labels are the year of publication of a paper).

Many GNNs explicitly assume homophily in their construction, and even those models that do not may perform poorly in non-homophilous graph settings [26, 60]. While some GNNs have been developed that work better in non-homophilous settings, their evaluation is often limited to a few graph datasets introduced by Pei et al. [44] that have certain undesirable properties such as small size, narrow range of application areas, and synthetic classes [10, 11, 36, 55, 59, 60].

In this work, we summarize issues with the non-homophilous graph datasets that are used in prior work and propose improved non-homophilous graph datasets that are substantially larger, span wider application areas, and capture different types of complex label-topology relationships. As previously-used homophily metrics have various shortcomings, we introduce a new metric that detects the presence or absence of homophily in graphs, and use it to analyze these graph datasets. We reintroduce strong simple methods for learning on graphs that have been overlooked in recent non-homophilous GNN work. Finally, we benchmark these simple methods along with general GNNs and non-homophilous GNNs on our proposed datasets. Our stronger empirical evaluation setup allows for a better understanding of graph learning methods in diverse settings.

*Authors contributed equally to this research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
WWW '21, April 19–23, 2021, Ljubljana, Slovenia.
© 2021 Copyright held by the owner/author(s).

2 PRIOR WORK

2.1 Graph Representation Learning

Graph neural networks [21, 30, 52] have demonstrated their utility on a variety of graph machine learning tasks. Most GNNs are constructed by stacking graph neural network layers that propagate transformed node features and then aggregate them via different mechanisms. The neighborhood aggregation used in many existing GNNs implicitly leverage homophily, so they often fail to generalize on non-homophilous graphs [5, 60]. Indeed, a wide range of GNNs operate as low-pass graph filters [5, 41, 53] that smooth features over the graph topology, which produces similar representations and thus similar predictions for neighboring nodes.

Due in part to the most common graph learning benchmarks exhibiting strong homophily, various graph representation learning methods have been developed that explicitly make use of an assumption of homophily in the data [8, 14, 24, 32, 53]. By leveraging this assumption, several simple, inexpensive models are able to achieve state-of-the-art performance on homophilic datasets [24, 53].

2.2 Non-Homophilous Methods

Various GNNs have been proposed to achieve higher performance in low-homophily settings. For instance, Geom-GCN [44] introduces a geometric aggregation scheme, MixHop [2] uses a graph convolutional layer that mixes powers of the adjacency matrix, GPR-GNN [11] uses learnable weights that can be positive and negative in feature propagation, and H_2GCN [60] shows that separation of ego and neighbor embeddings, aggregation in higher-order neighborhoods, and the combination of intermediate representations improves GNN performance in low-homophily. Also, various methods that only depend on graph topology have been proposed for non-homophilous settings, which are based on label propagation or supervised learning models [4, 12, 43, 57]. There are several recurring design decisions across these methods that appear to strengthen performance in non-homophilous settings: using higher-order neighborhoods, decoupling neighbor information from ego information, and combining graph information at different scales.

2.3 Real-world Datasets

Recently, the Open Graph Benchmark [23] has provided a series of datasets and leaderboards that improve the quality of evaluation in graph representation learning; however, most of the node classification datasets tend to be homophilous, as noted in past work [60] and expanded upon in Appendix A.1. A comparable set of high-quality benchmarks to evaluate non-homophilous methods does not currently exist.

The most widely used datasets to evaluate non-homophilous graph representation learning methods were presented by [44] (see our Appendix Table 4); however, these datasets have fundamental issues. First, they are very small — the Cornell, Texas, and Wisconsin datasets have between 180-250 nodes, and the largest of these datasets has 7,600 nodes. In analogy to certain pitfalls of graph neural network evaluation on small (homophilic) datasets discussed in [48], evaluation on the datasets in [44] is plagued by high variance across different train/test splits (see results in [60]). Also, the small size of these datasets may make models more prone to overfitting [17], and does not allow for experiments in scalability of GNNs designed for performance in non-homophilous settings. As a result, they are not satisfactory for evaluating the performance of node classification models for non-homophilous graphs, and larger, real-world datasets are necessary.

Peel [43] also studies node classification on network datasets with various types of relationships between edges and labels. However, they only study methods that act on node labels, and thus their datasets do not necessarily have node features. We take inspiration from their work, by testing on Pokec and Facebook networks with node features that we define, and by introducing other year-prediction tasks on citation networks that have node features.

2.4 Synthetic Data

Synthetically generated or synthetically modified non-homophilous graph data may also be used for evaluation of graph machine learning methods. Past works have taken various approaches to this [2, 11, 27, 28, 60]. When generating the graph, they control homophily through modifications of some generative models over graphs [1, 6, 15]. They tend to either sample node feature vectors from a real graph or draw them from multivariate Gaussian distributions. While synthetic data are useful for investigating properties of models in controlled settings, we focus on benchmarks from real-world data that exhibit diverse types of complex structure.

3 DATASETS AND METRICS

3.1 Measuring Homophily

Various metrics have been proposed to measure the homophily of a graph in a single scalar quantity. However, these metrics are sensitive to the number of classes and the number of nodes in each class. Thus, we introduce a metric that better captures the presence or absence of homophily. Our metric does not distinguish between different non-homophilous settings (such as heterophily or independent edges); we argue that there are too many degrees of freedom in non-homophilous settings for a single scalar quantity to be able to distinguish them all.

Let $G = (V, E)$ be a graph with n nodes, none of which are isolated. Further let each node $x \in V$ have a class label $k_x \in \{0, 1, \ldots, C-1\}$ for some number of classes C, and denote by C_k the set of nodes in class k. In recent non-homophilous graph representation learning work, the edge homophily [60] has been defined as the proportion of edges that connect two nodes of the same class:

$$h = \frac{|\{(x, v) \in E : k_x = k_v\}|}{|E|}. \tag{1}$$

Another related measure is what we call the node homophily [44], defined as $\frac{1}{|V|}\sum_{x \in V} \frac{d_x^{(k_x)}}{d_x}$, in which d_x is the number of neighbors of node x, and $d_x^{(k_x)}$ is the number of neighbors of x that have the same class label. We focus on the edge homophily (1) in this work, but find that node homophily tends to have similar qualitative behavior in experiments.

The sensitivity of edge homophily to the number of classes and size of each class limits its utility. Note that if edges were rewired randomly independently of node labels, a node $x \in V$ would be

expected to have $d_x^{(k_x)}/d_x \approx |C_k|/n$ as the proportion of nodes of the same class that they connect to [4]. For a dataset with C balanced classes, we would thus expect the edge homophily to be around $\frac{1}{C}$, so the interpretation of the measure depends on the number of classes. Also, if classes are imbalanced, then the edge homophily may be misleadingly large. For instance, if 99% of nodes were of one class, then most edges would likely be within that same class, so the edge homophily would be high.

We define a homophily measure that alleviates these shortcomings. Our measure is given as:

$$\hat{h} = \frac{1}{C-1} \sum_{k=0}^{C-1} \left[h_k - \frac{|C_k|}{n} \right]_+, \tag{2}$$

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

$$h_k = \frac{\sum_{x \in C_k} d_x^{(k_x)}}{\sum_{x \in C_k} d_x}. \tag{3}$$

Note that $\hat{h} \in [0, 1]$, with a fully homophilous graph (in which every node is only connected to nodes of the same class) having $\hat{h} = 1$. Since each class-wise homophily metric h_k only contributes positive deviations from the null expected proportion $|C_k|/n$, the class-imbalance problem is substantially mitigated. Also, graphs in which edges are independent of node labels are expected to have $\hat{h} \approx 0$, for any number of classes. Our measure $\hat{h}$ measures presence of homophily, but does not distinguish between the many types of possibly non-homophilous relationships. This is reasonable given the diversity of non-homophilous relationships. For example, non-homophily can imply independence of edges and classes, extreme heterophily, connections only among subsets of classes, or certain chemically / biologically determined relationships. Indeed, these relationships are very different, and are better captured by more than one scalar quantity, such as the compatibility matrices that are discussed in Appendix A. Figure 1 compares our measure $\hat{h}$ with the edge homophily ratio h.

On certain datasets where previous measures are misleading, our measure shows its advantages. For example, some of our proposed datasets are class-imbalanced (e.g. YelpChi and ogbn-proteins), so they have high edge homophily, but our measure $\hat{h}$ captures that they are indeed non-homophilous (Table 1). Moreover, as seen in Appendix Table 4, the edge homophily of Chameleon, Actor, and Squirrel are approximately the same, but the graph structures (Appendix Figure 5) and performance of different methods on these datasets vary significantly [60]. According to our measure, Chameleon is more homophilous than Squirrel, which is in turn more homophilous than Actor, and past work has shown that models tend to perform better on Chameleon than Squirrel and better on Squirrel than Actor [11, 44, 60]. Our measure suggests one possible axis of variation that may help explain this divergence, but we emphasize that there are many possible confounders. Further discussion is given in Appendix A.

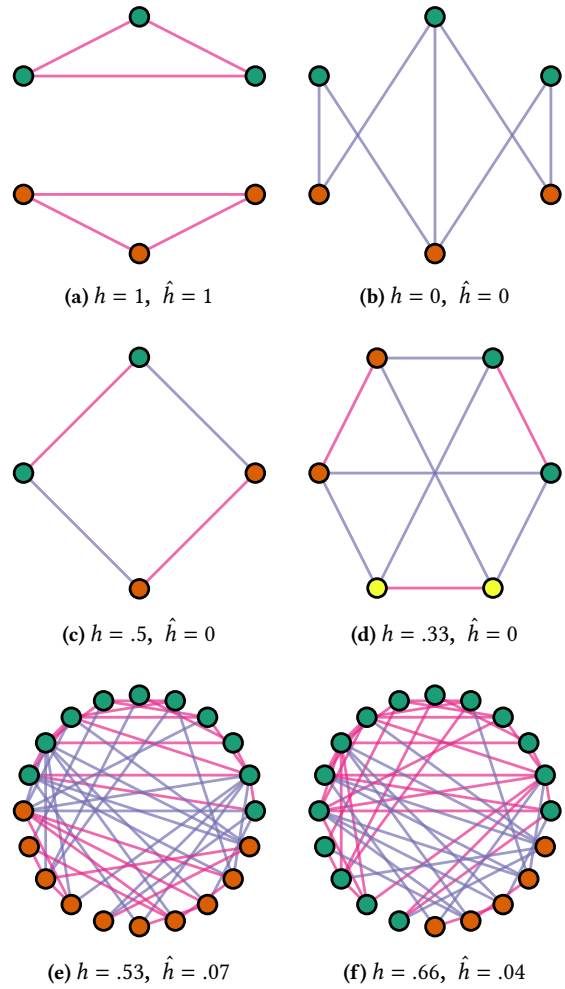

Figure 1: Examples of graphs with different label-topology relationships and comparison of our measure $\hat{h}$ with the edge homophily ratio h. The node classes are labeled by color. Pink edges link nodes of the same class, while purple edges link nodes of different classes. (a,b) Pure homophily and pure heterophily. Both measures equal 1 in homophily and 0 in heterophily. (c,d) Graphs where each node is connected to one member of each class. Edge homophily depends on the number of classes, while our measure $\hat{h}$ does not. (e,f) Random Erdős Rényi graphs in which edges are independent of labels. Edge homophily is sensitive to class imbalance, while our measure $\hat{h}$ is not.

3.2 Proposed Datasets

Here, we detail the non-homophilous datasets that we propose for graph machine learning evaluation. These datasets come from a variety of contexts, and some have been used for evaluation of graph machine learning models in past work; in certain cases, we make adjustments such as modifying node labels and adding node features. We define node features for Facebook100, Pokec, and snap-patents, while we redefine node labels for arXiv-year and snap-patents. Basic dataset statistics are given in Table 1. The datasets are as follows:

Table 1: Statistics of our proposed non-homophilous graph datasets. # C is the number of distinct node classes. # G/T is the number of different graphs or tasks. When there are multiple graphs or tasks, the minimum and maximum statistics are listed, with a hyphen "-" in between.

Dataset	# G/T	# Nodes	# Edges	# Node Feat.	# C	Class types	Edge hom.	$\hat{h}$ (ours)
Twitch-explicit	7	1,912 - 9,498	31,299 - 153,138	2,545	2	explicit content	.556 - .632	.049-.146
YelpChi	1	45,954	3,846,979	32	2	fake reviews	.773	.052
deezer-europe	1	28,281	92,752	31,241	2	gender	.525	.030
FB100	100	769-41,536	16,656-1,590,655	5	2	gender	.434 - .917	.000 - .246
Pokec	1	1,632,803	30,622,564	65	2	gender	.445	.000
ogbn-proteins	112	132,534	39,561,252	8	2	function	.623 - .940	.090 - .260
arXiv-year	1	169,343	1,166,243	128	5	pub year	.222	.272
snap-patents	1	2,923,922	13,975,788	269	5	time granted	.073	.100

Twitch-explicit contains 7 networks where Twitch users are nodes, and mutual friendships between them are edges [46]. Node features are games liked, location and streaming habits. Each graph is associated to users of a particular region. The class labels denote whether a streamer uses explicit language. We solely train and test on Twitch-DE, which has 9,498 nodes, 76,569 edges, edge homophily of .632, and $\hat{h}$ of .142.

YelpChi [39] is a graph in which the nodes are reviews for hotels and restaurants in the Chicago area, and the class labels are fraudulent reviews and recommended reviews. The 32 node features are taken from [45]. We take the topology from [16]; while there are different relations captured by edges, we treat them all as the same single edge type.

deezer-europe [47] is a social network of users on Deezer from European countries, where edges represent mutual follower relationships. The node features are based on artists liked by each user. Nodes are labeled with reported gender.

Facebook100 [51] consists of 100 Facebook friendship network snapshots from 2005, each of which has as nodes the Facebook users from a given American university. Each node is labeled with the reported gender of the user. The node features are major, second major/minor, dorm/house, year, and high school. We solely train and test on Penn94, which has 41,554 nodes, 1,362,229 edges, edge homophily of .470, and $\hat{h}$ of .046.

Pokec is the friendship graph of a Slovak online social network, where nodes are users and edges are directed friendship relations [35]. Nodes are labeled with reported gender. The node features are derived from profile information, such as geographical region, registration time, and age.

ogbn-proteins [23] has proteins as nodes and different biological relationships between proteins as edges. There are 112 tasks, in which each protein is given a binary label. Also, there are no separate node features — instead, the means of incoming edge features give 8 dimensional node features.

arXiv-year [23] is the ogbn-arxiv network with class labels given by the year that the paper is posted, instead of by subject areas. The nodes are arXiv papers, and directed edges connect a paper to other papers that it cites. The node features are averaged word2vec token features of both the title and abstract of the paper. The five classes are chosen by partitioning the posting dates so that class ratios are approximately balanced: 2013 and prior, 2014-2015, 2016-2017, 2018, and 2019-2020.

snap-patents [34, 35] is a dataset of utility patents granted between 1963 to 1999 in the US. Each node is a patent, and edges connect patents that cite each other. Node features are derived from patent metadata. The task is to predict the time at which a patent was granted. The five classes are: 1971 and prior, 1972-1980, 1981-1988, 1989-1994, and 1995-1999.

3.3 General Non-homophilous Settings

Different settings in which non-homophilous relationships are prevalent have been identified in the literature and are represented by our proposed datasets:

- Gender relations in social or interaction networks [4, 12, 26] (deezer, FB100, Pokec).
- Biological structures such as in food webs [19] and protein interactions [40] (ogbn-proteins).
- Technological and internet relationships, such as in web page connections [40, 44].
- Malicious or fraudulent nodes, such as in auction networks [9, 42] (YelpChi).
- Publication time in citation networks [43] (arXiv-year, snap-patents).

While not all example graph data from these contexts are non-homophilous, a diverse range are. In order to succeed in future applications in these contexts, it may be important to develop methods that are able to handle non-homophilous structures.

4 EXPERIMENTS

4.1 Experimental Setup

We include both methods that are graph-agnostic and node-feature-agnostic as simple baselines; the node-feature-agnostic models of two-hop label propagation [43] and LINK (logistic regression on the adjacency matrix) [57] have been found to perform well in various non-homophilous settings, but they have often been overlooked by recent graph representation learning work. Also, we include SGC [53] and C&S [24] as simple methods that perform well on homophilic datasets. We include a two-hop propagation variant of C&S in analogy with two-step label propagation. In addition to representative general GNNs, we also include three GNNs recently proposed for non-homophilous settings. The full list of methods is:

Table 2: Experimental results. Test accuracy is displayed for most datasets, while Twitch-DE, YelpChi, and ogbn-proteins display test ROC AUC†. Standard deviations are over 5 train/val/test splits, except for ogbn-proteins, which has a fixed split. The three best results per dataset are highlighted in blue. (M) denotes some (or all) hyperparameter settings run out of memory.

	Twitch-DE†	YelpChi†	deezer	Penn94 (FB100)	pokec	ogbn-proteins†	arXiv-year	snap-patents
MLP	69.20 ± 0.62	87.94 ± 0.52	66.55 ± 0.72	73.61 ± 0.40	62.37 ± 0.02	73.43 ± 0.12	36.70 ± 0.21	31.34 ± 0.05
L Prop (1 hop)	71.00 ± 1.01	63.64 ± 0.73	56.50 ± 0.41	63.21 ± 0.39	53.09 ± 0.05	75.14 ± 0.00	43.42 ± 0.17	30.28 ± 0.09
L Prop (2 hop)	72.27 ± 0.78	62.34 ± 0.78	56.96 ± 0.26	74.13 ± 0.46	76.76 ± 0.03	65.79 ± 0.00	46.07 ± 0.15	38.61 ± 0.07
LINK	72.42 ± 0.57	63.44 ± 1.07	57.71 ± 0.36	80.79 ± 0.49	80.54 ± 0.03	63.49 ± 0.02	53.97 ± 0.18	60.39 ± 0.07
SGC (1 hop)	72.30 ± 0.22	58.62 ± 0.85	59.73 ± 0.12	66.79 ± 0.27	53.61 ± 0.17	50.58 ± 1.08	32.83 ± 0.13	30.31 ± 0.06
SGC (2 hop)	73.65 ± 0.40	57.18 ± 0.75	61.56 ± 0.51	76.09 ± 0.45	62.81 ± 1.42	49.48 ± 1.45	32.27 ± 0.06	29.09 ± 0.09
C&S (1 hop)	68.54 ± 1.25	87.47 ± 0.50	64.60 ± 0.57	72.55 ± 0.63	63.23 ± 0.07	71.13 ± 0.69	42.08 ± 0.27	37.41 ± 0.07
C&S (2 hop)	69.39 ± 0.85	87.40 ± 0.44	64.52 ± 0.62	72.47 ± 0.73	77.22 ± 0.61	69.31 ± 0.62	42.17 ± 0.27	44.62 ± 0.14
GCN	74.07 ± 0.68	63.62 ± 1.00	62.23 ± 0.53	82.47 ± 0.27	75.45 ± 0.17	72.03 ± 0.32	46.02 ± 0.26	45.65 ± 0.04
GAT	73.13 ± 0.29	81.42 ± 2.12 (M)	61.09 ± 0.77	81.53 ± 0.55	71.77 ± 6.18 (M)	(M)	46.05 ± 0.51	45.37 ± 0.44 (M)
GCN+JK	70.57 ± 2.22	64.35 ± 0.86	60.99 ± 0.14	81.63 ± 0.54	77.00 ± 0.14	76.88 ± 0.48	46.28 ± 0.29	46.88 ± 0.13
GAT+JK	73.06 ± 0.49	90.04 ± 0.61 (M)	59.66 ± 0.92	80.69 ± 0.36	71.19 ± 6.96 (M)	(M)	45.80 ± 0.72	44.78 ± 0.50 (M)
APPNP	72.79 ± 0.29	75.60 ± 0.48	67.21 ± 0.56	74.95 ± 0.45	59.90 ± 0.11	68.96 ± 0.09	36.11 ± 0.79	31.78 ± 0.05
H_2GCN	72.67 ± 0.65	88.48 ± 0.21	67.22 ± 0.90	(M)	(M)	(M)	49.09 ± 0.10	(M)
MixHop	73.23 ± 0.99	87.02 ± 0.50	66.80 ± 0.58	83.47 ± 0.71	81.07 ± 0.16	75.60 ± 0.85	51.81 ± 0.17	52.16 ± 0.09 (M)
GPR-GNN	73.84 ± 0.69	71.59 ± 0.38	66.90 ± 0.50	84.59 ± 0.29	74.62 ± 0.48	(M)	36.58 ± 1.12	39.38 ± 0.07

- Models that only use node features: MLP [20].
- Models that only use the graph topology: label propagation (standard and two-hop) [43, 58], LINK [57].
- Simple methods: SGC [53], C&S [24], two-hop variants.
- General GNNs: GCN [30], GAT [52], jumping knowledge networks (GCN+JK, GAT+JK) [54], and APPNP [31].
- Non-homophilous GNNs: H_2GCN [60], MixHop [2], and GPR-GNN [11].

Following other works in non-homophilous graph learning evaluation, we take a high proportion of training nodes [44, 55, 60]; we run each method on the same five random 50/25/25 train/val/test splits for each dataset, besides ogbn-proteins, for which we use the original Open Graph Benchmark splits [23]. All methods requiring gradient-based optimization are run for 500 epochs, with test performance reported for the learned parameters of highest validation performance. We use ROC-AUC as the metric for the class-imbalanced Twitch-DE (60.5% of nodes in majority class), YelpChi (85.5% of nodes in majority class), and ogbn-proteins datasets (52.7% to 98.0% in majority, depending on task). For other datasets, we use classification accuracy as the metric. Further experimental details can be found in Appendix B.

4.2 Experimental Results

Table 2 lists the results of each method across the datasets that we propose. Our new measure and new datasets reveal several important properties of non-homophilous node classification. Firstly, both methods that only use node features and methods that only use graph topology appear to perform better than random, thus demonstrating the quality of our datasets. Secondly, the stability of performance across runs is better for our datasets than those of Pei et al. [44] (see [60] results). Moreover, as suggested by prior theory and experiments [2, 11, 60], the non-homophilous GNNs usually do well — though not necessarily on every dataset.

The core assumption of homophily in SGC and C&S that enables them to be simple and efficient does not hold on these non-homophilous datasets, and thus the performance of these methods is typically relatively low. Nevertheless, 2-hop C&S still achieves high performance on some datasets. Indeed, our results indicate that simple 2-hop modifications of learning methods often improve performance in low-homophily, though we note that 1-hop label propagation performs much better than expected on ogbn-proteins, possibly due to some implementation nuance. LINK, a frequently ignored baseline that in some sense acts on two-hop neighborhoods of each node [4], performs well on many datasets — despite not using node feature information.

Finally, one consequence of using larger datasets for benchmarks is that the tradeoff between scalability and learning performance of non-homophilous methods has become starker, with some methods facing memory issues. This tradeoff is especially important to consider in light of the fact that many scalable graph learning methods rely on implicit or explicit homophily assumptions [8, 14, 24, 53], and thus face issues when used in non-homophilous settings.

5 CONCLUSION

In this paper, we introduce a measure of the presence of homophily that alleviates issues with existing measures, propose new, high-quality non-homophilous graph learning datasets, and benchmark simple baselines and representative graph representation learning methods across our datasets. We hope that these contributions will provide researchers of non-homophilous graph learning methods with better tools to test their models and evaluate the utility of new techniques. While we benchmark on transductive node classification, the datasets we propose could be adapted to benchmark link prediction, clustering tasks, and inductive node classification in the case of Twitch-explicit and Facebook100. Future work could study these other tasks in low-homophily settings, reformulate current understandings of homophily that are most natural in node classification, and introduce new benchmarks for a wider range of applications.

ACKNOWLEDGMENTS

We thank Austin Benson and Horace He for insightful discussions. We also thank the rest of Cornell University Artificial Intelligence for their support and discussion. This research was supported by Facebook AI.

REFERENCES

[1] Emmanuel Abbe. 2017. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research* 18, 1 (2017), 6446–6531.
[2] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In *International Conference on Machine Learning*. 21–29.
[3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv:1907.10902 [cs.LG]
[4] Kristen M Altenburger and Johan Ugander. 2018. Monophily in social networks introduces similarity among friends-of-friends. *Nature human behaviour* 2, 4 (2018), 284–290.
[5] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. 2021. Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective. In *International Conference on Learning Representations*. https://openreview.net/forum?id=-qh0M9XWxnv
[6] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.
[7] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* (2018).
[8] Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. 2020. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2464–2473.
[9] Duen Horng Chau, Shashank Pandit, and Christos Faloutsos. 2006. Detecting fraudulent personalities in networks of online auctioneers. In *European conference on principles of data mining and knowledge discovery*. Springer, 103–114.
[10] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*. PMLR, 1725–1735.
[11] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*. https://openreview.net/forum?id=n6jl7fLxrP
[12] Alex Chin, Yatong Chen, Kristen M. Altenburger, and Johan Ugander. 2019. Decoupled smoothing on graphs. In *The World Wide Web Conference*. 263–272.
[13] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* (2015).
[14] Chenhui Deng, Zhiqiang Zhao, Yongyu Wang, Zhiru Zhang, and Zhuo Feng. 2020. GraphZoom: A Multi-level Spectral Approach for Accurate and Scalable Graph Embedding. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1lGO0EKDH
[15] Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. 2018. Contextual stochastic block models. In *Advances in Neural Information Processing Systems*. 8581–8593.
[16] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 315–324.
[17] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982* (2020).
[18] Paul Erdős and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.
[19] Wolfgang Gatterbauer. 2014. Semi-supervised learning with heterophily. *arXiv preprint arXiv:1412.3100* (2014).
[20] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.
[21] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.
[22] William L. Hamilton. 2020. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14, 3 (2020), 1–159.
[23] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687* (2020).
[24] Qian Huang, Horace He, Abhay Singh, Ser-Nam Lim, and Austin Benson. 2021. Combining Label Propagation and Simple Models out-performs Graph Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=8E1-f3VhX1o
[25] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.
[26] Junteng Jia and Austion R Benson. 2020. Residual Correlation in Graph Neural Network Regression. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 588–598.
[27] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. 2018. Homophily influences ranking of minorities in social networks. *Scientific reports* 8, 1 (2018), 1–12.
[28] Dongkwan Kim and Alice Oh. 2021. How to Find Your Friendly Neighborhood: Graph Attention Design with Self-Supervision. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Wi5KUNlqWty
[29] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[30] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG]
[31] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1gL-2A9Ym
[32] Johannes Klicpera, Stefan Weiß enberger, and Stephan Günnemann. 2019. Diffusion Improves Graph Learning. In *Advances in Neural Information Processing Systems*, Vol. 32.
[33] David Laniado, Yana Volkovich, Karolin Kappler, and Andreas Kaltenbrunner. 2016. Gender homophily in online dyadic and triadic relationships. *EPJ Data Science* 5 (2016), 1–23.
[34] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 177–187.
[35] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.
[36] Meng Liu, Zhengyang Wang, and Shuiwang Ji. 2020. Non-Local Graph Neural Networks. arXiv:2005.14612 [cs.LG]
[37] Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.
[38] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.
[39] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013. What yelp fake review filter might be doing?. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 7.
[40] Mark EJ Newman. 2003. Mixing patterns in networks. *Physical review E* 67, 2 (2003), 026126.
[41] Hoang Nt and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550* (2019).
[42] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*. 201–210.
[43] Leto Peel. 2017. Graph-based semi-supervised learning for relational networks. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 435–443.
[44] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2019. Geom-GCN: Geometric Graph Convolutional Networks. In *International Conference on Learning Representations*.
[45] Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*. 985–994.
[46] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2019. Multi-scale attributed node embedding. *arXiv preprint arXiv:1909.13021* (2019).
[47] Benedek Rozemberczki and Rik Sarkar. 2020. Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 1325–1334.
[48] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).
[49] Neil JA Sloane. 2007. The on-line encyclopedia of integer sequences. In *Towards mechanized mathematical assistants*. Springer, 130–130.
[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[51] Amanda L Traud, Peter J Mucha, and Mason A Porter. 2012. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications* 391, 16 (2012), 4165–4180.
[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJXMpikCZ
[53] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.
[54] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*. PMLR, 5453–5462.
[55] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. 2021. Two Sides of the Same Coin: Heterophily and Oversmoothing in Graph Convolutional Neural Networks. *arXiv preprint arXiv:2102.06462* (2021).
[56] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*. PMLR, 40–48.
[57] Elena Zheleva and Lise Getoor. 2009. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th international conference on World wide web*. 531–540.
[58] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. (2004).
[59] Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. 2020. Graph Neural Networks with Heterophily. *arXiv preprint arXiv:2009.13566* (2020).
[60] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. *Advances in Neural Information Processing Systems* 33 (2020).

A COMPATIBILITY MATRICES AND STATISTICS

Following previous work [60], for a graph G with C node classes we define the $C \times C$ compatibility matrix $\mathbf{H}$ by

$$\mathbf{H}_{kl} = \frac{|(u,v) \in E : k_u = k,\ k_v = l|}{|(u,v) \in E : k_u = k|} \tag{4}$$

This captures finer details of label-topology relationships in graphs than single scalar metrics (like edge homophily and our $\hat{h}$) capture. For classes k and l, the entry $\mathbf{H}_{kl}$ measures the proportion of edges from nodes of class k that are connected to nodes of class l. Compatibility matrices for our proposed datasets are shown in Figure 2. As evidenced by the different patterns, the proposed datasets show interesting types of label-topology relationships besides homophily. For instance, the citation datasets arXiv-year and snap-patents have primarily lower-triangular structure, since most citations reference past work. The Penn94 matrix is mostly uniform, so there is little to no gender preference in aggregate, though past work has shown other useful signals in the social structure of gender [4]. In Twitch-DE, while streamers that use explicit content (class 1) often connect to other streamers of class 1, streamers that do not use explicit content (class 0) also often connect to streamers of class 1, thus giving an overall non-homophilous structure. In Pokec, there is some heterophily, in that one gender has some preference for friends of another gender.

A.1 Homophilous Data Statistics

In contrast to the different compatibility matrix structures of our proposed non-homophilous datasets, much other graph data have primarily homophilous relationships, as can be seen in Figure 4 and Table 3. The Cora, CiteSeer, PubMed, ogbn-arxiv, and ogbn-products datasets are widely used as benchmarks for node classification [23, 56], and are highly homophilous, as can be seen by the diagonally dominant structure of the compatibility matrices and by the high edge homophily and $\hat{h}$.

Table 3: Statistics for homophilic graph datasets. # C is the number of node classes.

Dataset	# Nodes	# Edges	# C	Edge hom.	$\hat{h}$ (ours)
Cora	2,708	5,278	7	.81	.766
Citeseer	3,327	4,552	6	.74	.627
Pubmed	19,717	44,324	3	.80	.664
ogbn-arXiv	169,343	1,166,243	40	.66	.416
ogbn-products	2,449,029	61,859,140	47	.81	.459
oeis	226,282	761,687	5	.50	.532

We collected the oeis dataset displayed in the bottom right of Figure 4. The nodes are entries in the Online Encyclopedia of Integer Sequences [49], and directed edges link an entry to any other entry that it cites. In analogy to arXiv-year and snap-patents, the node labels are the time of posting of the sequence. However, in this case the graph relationships are homophilous, even as we vary the number of distinct classes (time periods). This is in part due to differences between posting in this online encyclopedia and publication of academic papers or patents. For instance, there is less overhead to posting an entry in the OEIS, so users often post separate related entries and variants of these entries in rapid succession. Also, an entry in the encyclopedia often inspires other people to work on similar entries, which can be created in much less time than an academic follow-up work to a given paper. These related entries tend to cite each other, which contributes to homophilic relationships over time. Thus, the data here does not follow the special temporal citation structure of academic publications and patents.

A.2 Previous Non-Homophilous Data

For the six datasets in Pei et al. [44] often used in evaluation of graph representation learning methods in non-homophilous regimes [60], basic statistics are listed in Table 4 and compatibility matrices are displayed in Figure 5. We propose many more datasets (including the 100 graphs from Facebook100) that have up to orders of magnitude more nodes and edges and come from a wider range of contexts. There are several cases of class-imbalance in these datasets, which may make the edge homophily misleading. As discussed in Section 3.1, our measure may be able to alleviate issues with edge homophily in measuring homophily of these datasets, and offers a way to distinguish between the Chameleon, Actor, and Squirrel datasets that all have similar edge homophily.

A.3 Class-Imbalance and Metrics

In this section, we present experiments that demonstrate an instance in which our metric is not affected by imbalanced classes, while edge homophily is. We generate graphs in which node labels are independent of edges by randomly choosing node labels and generating graph edges by the Erdős-Rényi random graph model [18]. In particular, we fix the number of classes to two, the number of nodes to 100, and the probability of edge formation as .25 between every pair of nodes. Then we generate 100 samples of these random graphs, and compute the mean and standard deviation of

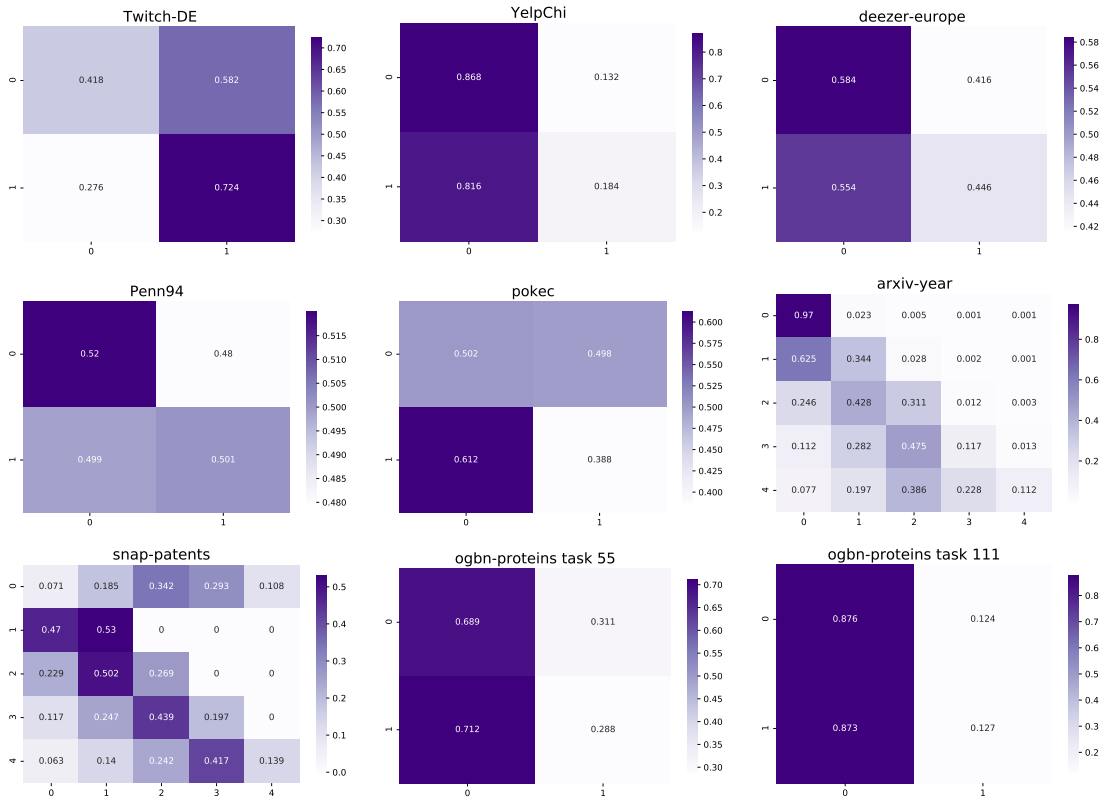

Figure 2: Compatibility matrices of our proposed datasets. These datasets from a variety of different contexts exhibit a wide range of non-homophilous structures. We choose 2 of the 112 ogbn-proteins tasks to display.

Dataset	# Nodes	# Edges	# Node Feat.	# C	Context	Edge hom.	$\hat{h}$ (ours)
Chameleon	2,277	36,101	2,325	5	Wiki pages	.23	.062
Cornell	183	295	1,703	5	Web pages	.30	.047
Actor	7,600	29,926	931	5	Actors in movies	.22	.011
Squirrel	5,201	216,933	2,089	5	Wiki pages	.22	.025
Texas	183	309	1,703	5	Web pages	.11	.001
Wisconsin	251	499	1,703	5	Web pages	.21	.094

Table 4: Statistics for datasets from Pei et al. [44]. #C is the number of node classes.

both edge homophily h and our measure $\hat{h}$. As seen in Figure 3, our measure $\hat{h}$ is constantly near zero as we increase the size of the majority class, while the edge homophily h increases as the size of majority class increases.

B EXPERIMENTAL DETAILS

For gradient-based optimization, we use the AdamW optimizer [29, 37] with weight decay .001 and learning rate .01 by default, unless we tune the optimizer for a particular method (as noted below in B.1). In all cases, we use full batch gradient descent across the entire graph dataset. Hyperparameter tuning is conducted using grid search for most methods. Tuning for C&S is done as in the original paper [24], which uses Optuna [3] for Bayesian hyperparameter optimization. GCN and GCN-JK on ogbn-proteins use the hyper-parameters of methods on the Open Graph Benchmark leaderboards [23]. All graphs are treated as undirected besides arXiv-year and snap-patents, in which the directed nature of the edges capture useful temporal information; however, we find that label propagation and C&S (which builds on label propagation) perform better with undirected graphs in these cases, so we keep the graphs as undirected for these methods.

Simple methods are run on a Nvidia 2080 Ti with 11 GB GPU memory. In cases where the Nvidia 2080Ti did not provide enough memory, we re-ran experiments on a Nvidia Titan RTX with 24GB

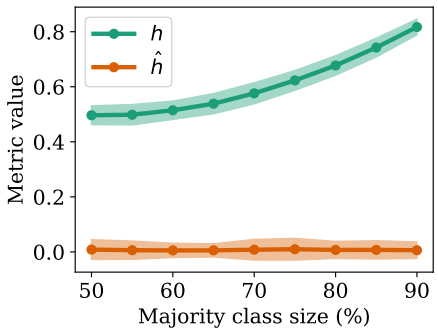

Figure 3: Comparison of edge homophily h and our measure $\hat{h}$ on random class-imbalanced graph data with edges independent of node labels. Three standard deviations are shaded. Our measure is mostly constant as the classes become more imbalanced, while edge homophily increases.

GPU memory, reporting (M) in Table 2 if the GPU memory was still insufficient.

B.1 Hyperparameters

Experimental results are reported on the hyperparameter settings below, where we choose the settings that achieve the highest performance on the validation set. We choose hyperparameter grids that do not necessarily give optimal performance, but hopefully cover enough regimes so that each model is reasonably evaluated on each dataset. Unless otherwise stated, each GNN has dropout of .5 [50] and BatchNorm [25] in each layer. The hyperparameter grids for the different methods are:

- MLP: hidden dimension $\in \{16, 32, 64, 128, 256\}$, number of layers $\in \{2, 3\}$. We use ReLU activations.
- Label propagation: $\alpha \in \{.01, .1, .25, .5, .75, .9, .99\}$. We use 50 propagation iterations.
- LINK: weight decay $\in \{.001, .01, .1\}$.
- SGC: weight decay $\in \{.001, .01, .1\}$.
- C&S: Normalized adjacency matrix $\mathrm{A}_1, \mathrm{A}_2 \in \{D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, D^{-1}A, AD^{-1}\}$ for the residual propagation and label propagation, where A is the adjacency matrix of the graph and D is the diagonal degree matrix; $\alpha_1, \alpha_2 \in (0.0, 1.0)$ for the two propagations. Both Autoscale and FDiff-scale were used for all experiments, and scale $\in (0.1, 10.0)$ was searched in FDiff-scale settings. The base predictor is chosen as the best MLP model for each dataset.
- GCN: lr $\in \{.1, .01, .001\}$, hidden dimension $\in \{4, 8, 16, 32, 64\}$, except for snap-patents and pokec, where we omit hidden dimension = 64, and ogbn-proteins, where we use the experimental setup of Hu et al. [23]. Each activation is a ReLU. 2 layers were used for all experiments except for ogbn-proteins.
- GAT: lr $\in \{.1, .01, .001\}$. For snap-patents and pokec: hidden channels $\in \{4, 8, 12\}$ and gat heads $\in \{2, 4\}$. For all other datasets: hidden channels $\in \{4, 8, 12, 32\}$ and gat heads $\in \{2, 4, 8\}$. We use the ELU activation [13]. 2 layers were used for all experiments.
- GCN-JK: Identical for GCN, also including JK Type $\in$ {cat, max}.
- GAT-JK: Identical for GAT, also including JK Type $\in$ {cat, max}.
- APPNP: MLP hidden dimension $\in \{16, 32, 64, 128, 256\}$, learning rate $\in \{.01, .05, .002\}$, $\alpha \in \{.1, .2, .5, .9\}$. BatchNorm is not used in the initial MLP. We truncate the series at the $K = 10$th power of the adjacency.
- H_2GCN: hidden dimension $\in \{8, 16, 32, 64\}$, number of layers $\in \{1, 2\}$, dropout $\in \{0, .5\}$. The architecture follows Section 3.2 of [60].
- MixHop: hidden dimension $\in \{8, 16, 32\}$, number of layers $\in \{2, 3\}$. Each layer has uses the 0th, 1st, and 2nd powers of the adjacency and has ReLU activations. The last layer is a linear projection layer, instead of the attention output mechanism in [2].
- GPR-GNN: The basic setup and grid is the same as that of APPNP. We use their Personalized PageRank weight initialization.

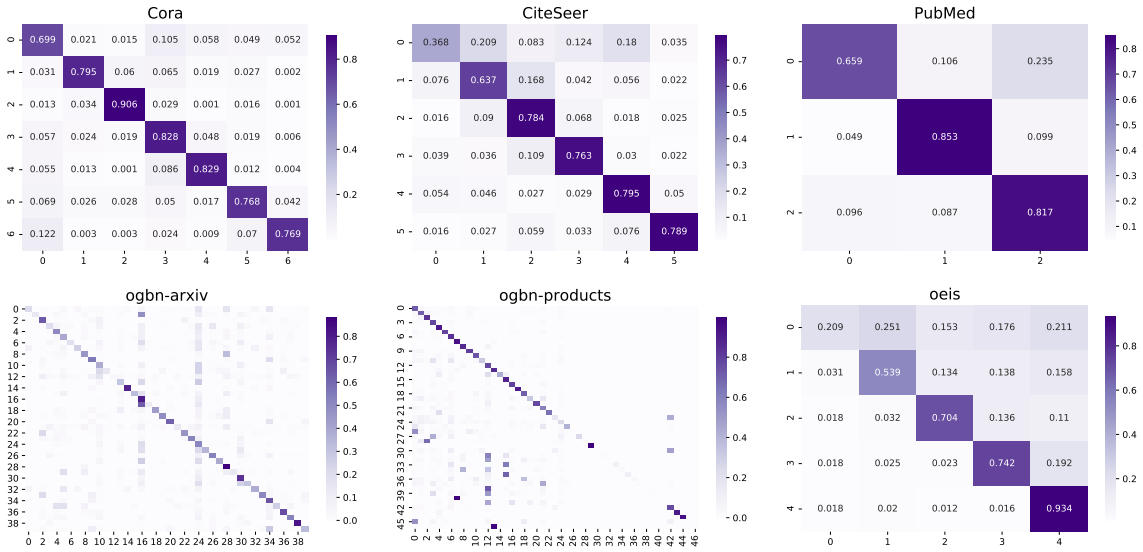

Figure 4: Compatibility matrices of homophilic datasets. The diagonal dominance indicates strong homophily.

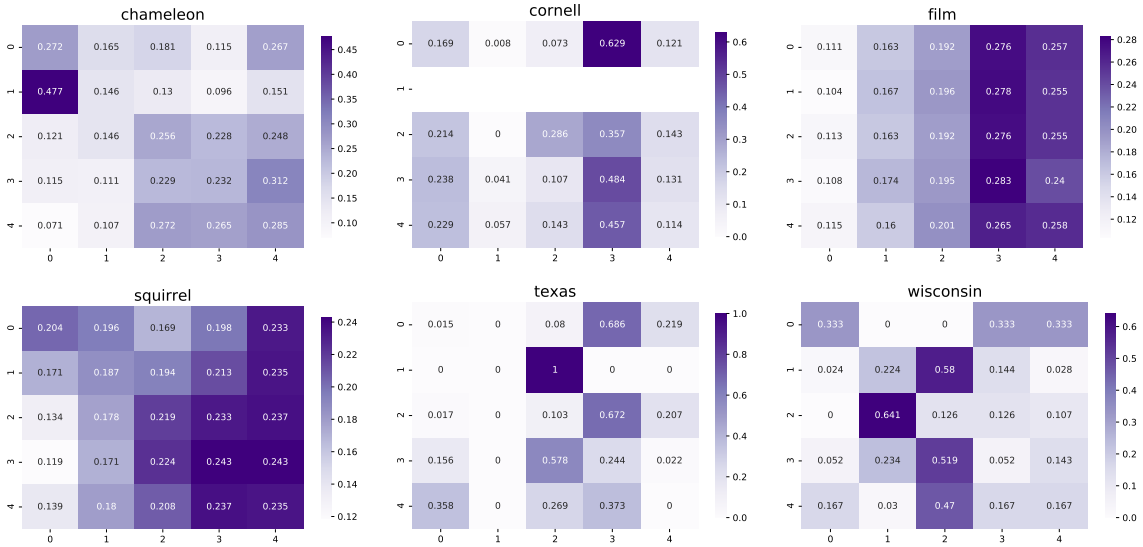

Figure 5: Compatibility matrices of datasets in Pei et al. [44]. The "film" dataset is also referred to as "Actor". Note that there are no edges leading out of the nodes of class 1 in the Cornell dataset, so there is an empty row in its matrix.