Towards a Taxonomy of Graph Learning Datasets

Renming Liu[3], **Semih Cantürk**[1,2],
Frederik Wenkel[1,2], **Dylan Sandfelder**[1,4], **Devin Kreuzer**[1,4], **Anna Little**[5], **Sarah McGuire**[3],
Leslie O'Bray[7], **Michael Perlmutter**[6], **Bastian Rieck**[7],
Matthew Hirn[3], **Guy Wolf**[1,2], and **Ladislav Rampášek**[1,2]

[1]Mila - Quebec AI Institute, [2]Université de Montréal, [3]Michigan State University,
[4]McGill University, [5]University of Utah, [6]University of California, Los Angeles, [7]ETH Zürich
mhirn@msu.edu, {wolfguy, ladislav.rampasek}@mila.quebec

Abstract

Graph neural networks (GNNs) have attracted much attention due to their ability to leverage the intrinsic geometries of the underlying data. Although many different types of GNN models have been developed, with many benchmarking procedures to demonstrate the superiority of one GNN model over the others, there is a lack of systematic understanding of the underlying benchmarking datasets, and what aspects of the model are being tested. Here, we provide a principled approach to taxonomize graph benchmarking datasets by carefully designing a collection of graph perturbations to probe the essential data characteristics that GNN models leverage to perform predictions. Our data-driven taxonomization of graph datasets provides a new understanding of critical dataset characteristics that will enable better model evaluation and the development of more specialized GNN models.

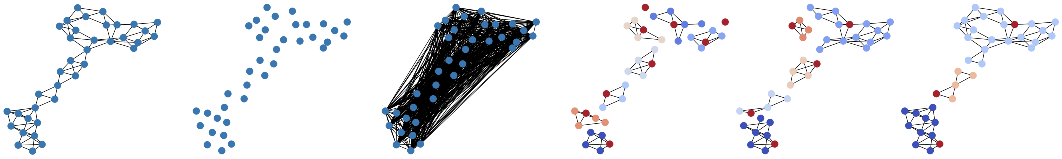
1 Introduction

Machine learning for graph-structured data has seen rapid development in recent years [10]. Originally inspired by convolutional neural networks, which are very successful in regular Euclidean domains thanks to their ability to leverage data-intrinsic geometries, classical graph neural network (GNN) models [6, 12, 21] translate those principles to graph domains. Further advancements in the field lead to a wide selection of complex and powerful GNN architectures. Some models are provably more expressive than others [24, 15], can leverage multi-resolution views of graphs [13], or can account for implicit symmetries of the graph data [3]; comprehensive surveys of graph neural networks can be found in [4, 23, 27]. However these GNN methods are historically evaluated on a set of small datasets [14] that became insufficient to serve as distinguishing benchmarks [8]. Therefore, recent work has focused on compiling a set of large(r) benchmarking datasets across diverse graph domains [8, 11]. Despite these efforts and the introduction of new datasets, it is still not well-understood, for example, whether node features or (sub)graph structural patterns are more influential, or how important long-range interactions are. Hence it is not clear what aspects of GNNs' representation capabilities are tested by a given benchmark. Here, we explore graph data characteristics in relation to the prediction tasks and establish the first taxonomic view of GNN benchmarking datasets. These insights improve our understanding of empirical evaluations of GNNs and will lead to appropriate empirical validation of future models.

2 Methods

Information in graph data is typically encoded by two factors: (i) *node features* that represent the properties at individual nodes of the graph, and (ii) *graph structure* that represents relations between nodes. Leveraging symmetries and other geometric priors in graph data is crucial for generalizable learning [3]. While invariance (i.e., symmetry) or equivariance to some transformations

Preprint. Under review.



(a) original (b) no-edges (c) fully-conn. (d) frag., $k = 1$ (e) frag., $k = 2$ (f) frag., $k = 3$

Figure 1: **Graph structure perturbations.** An illustrative example from ENZYMES dataset.

is inherent, invariance to others may be only empirical or partial. Here we use this lens of empirical transformation sensitivity to gauge *how* task-related information is encoded in graph datasets. We perturb graph datasets with a set of transformations designed to eliminate or emphasize particular types of information embedded in the graphs. As a proxy to how invariant or sensitive a given prediction task is to these graph perturbations, we observe the empirical change in GNN performance on such perturbed versions of the datasets compared to the originals. For a given dataset and its prediction task, the empirical *sensitivity profile* to the set of perturbations represents a comprehensive view of what information is important and needs to be captured by a GNN model. Based on these sensitivity profiles, we cluster the analyzed datasets and propose their taxonomy.

First, we utilize two types of node perturbations: node features are either discarded or they are replaced by one-hot encodings of node degrees; we refer to these perturbations as *no-node-features* and *node-degree*, respectively. Secondly, we design a set perturbations acting on graph structure (Fig. 1). Perturbations that remove all edges (*no-edges)* or make the graph *fully-connected* eliminate structural information and essentially turn the graph into a set, causing the nodes to be either processed fully independently, or collectively. Next, to inspect the importance of local vs. global graph structure, we design the *fragmented* perturbation, which partitions the graph into connected components consisting of nodes whose distance to the seed node is less than k. A smaller k implies smaller components, and hence discards the global structure and long-range interactions.

With these perturbations, we comprehensively profile a selection of widely used graph-level classification datasets (Fig. 2) that cover the benchmarking dataset collection of Dwivedi et al. [8], and a wide range of node-level classification datasets (Fig. 3) accessed via the PyG package [9].

3 Results

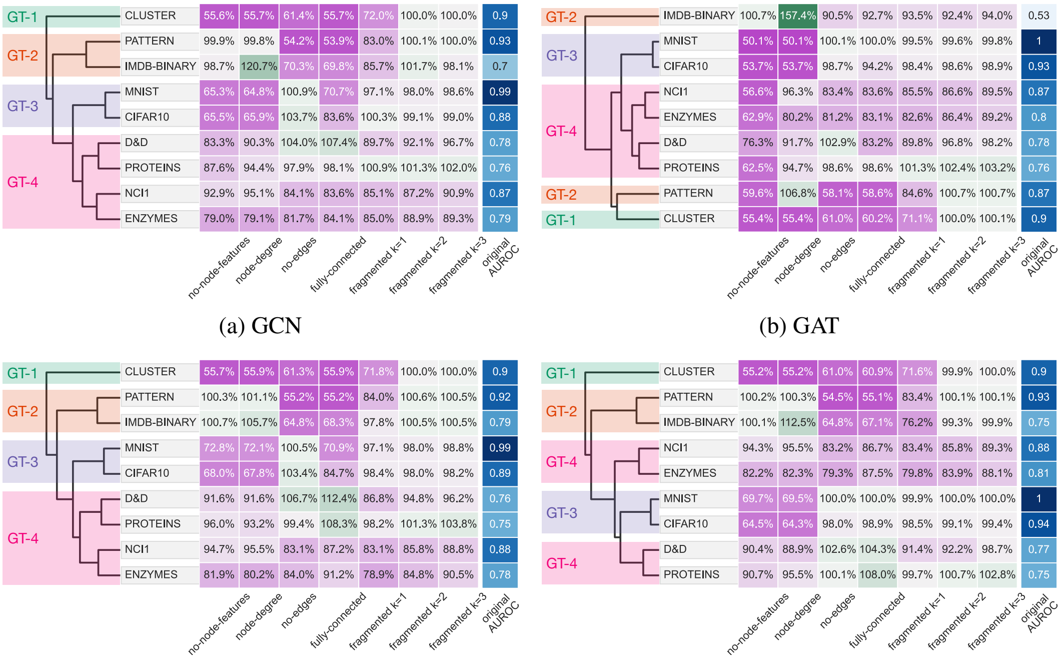
To obtain the perturbation sensitivity profiles, we consider four popular GNN models: GCN [12], GAT [21], GIN [24], and ChebNet [6]. We keep the model hyperparameters identical for each dataset and perturbation combination: 2-layer MLP for node embedding (only for node-level tasks), 5 graph convolutional layers with residual connections and batch normalization, followed by global mean pooling (for graph-level prediction tasks), and finally a 2-layer MLP classifier. The classification performance is evaluated in terms of AUROC by averaging over either 10-fold stratified cross-validation (for datasets without a standard training/validation/test split); or averaged over 10 repetitions with different random initializations using the standard splits.

3.1 Graph-level prediction tasks

We identify a categorization into four dataset clusters based on hierarchical clustering of their perturbation sensitivity profiles. We refer to these **g**raph-level **t**ask clusters as GT-1 to GT-4, respectively. This categorization is stable across the four GNN models with only minor deviations. Here we include CLUSTER and PATTERN datasets, that are in fact node classification datasets, but their inductive learning setup makes them more akin to graph-level tasks than to transductive node-level tasks.

Grouped in GT-3, the image-derived datasets CIFAR10 and MNIST [8] share very similar sensitivity profiles, showing that the color of underlying superpixels (node features) is the dominant information, while the graph structure is irrelevant. Interestingly, however, early global symmetry (*fully-connected*) appears to lead to detrimental (over-)smoothing for GCN and GIN, while GAT and ChebNet are robust to this perturbation.

CLUSTER and PATTERN [8], generated from a stochastic block model, as well as the actor/actress ego-network dataset IMDB-BINARY [25], are greatly affected by elimination of *local* graph structure.



		no-node-features	node-degree	no-edges	fully-connected	fragmented k=1	fragmented k=2	fragmented k=3	original AUROC
GT-1	CLUSTER	55.6%	55.7%	61.4%	55.7%	72.0%	100.0%	100.0%	0.9
GT-2	PATTERN	99.9%	99.8%	54.2%	53.9%	83.0%	100.1%	100.0%	0.93
	IMDB-BINARY	98.7%	120.7%	70.3%	69.8%	85.7%	101.7%	98.1%	0.7
GT-3	MNIST	65.3%	64.8%	100.9%	70.7%	97.1%	98.0%	98.6%	0.99
	CIFAR10	65.5%	65.9%	103.7%	83.6%	100.3%	99.1%	99.0%	0.88
GT-4	D&D	83.3%	90.3%	104.0%	107.4%	89.7%	92.1%	96.7%	0.78
	PROTEINS	87.6%	94.4%	97.9%	98.1%	100.9%	101.3%	102.0%	0.76
	NCI1	92.9%	95.1%	84.1%	83.6%	85.1%	87.2%	90.9%	0.87
	ENZYMES	79.0%	79.1%	81.7%	84.1%	85.0%	88.9%	89.3%	0.79

(a) GCN

		no-node-features	node-degree	no-edges	fully-connected	fragmented k=1	fragmented k=2	fragmented k=3	original AUROC
GT-2	IMDB-BINARY	100.7%	157.4%	90.5%	92.7%	93.5%	92.4%	94.0%	0.53
GT-3	MNIST	50.1%	50.1%	100.1%	100.0%	99.5%	99.6%	99.8%	1
	CIFAR10	53.7%	53.7%	98.7%	94.2%	98.4%	98.6%	98.9%	0.93
GT-4	NCI1	56.6%	96.3%	83.4%	83.6%	85.5%	86.6%	89.5%	0.87
	ENZYMES	62.9%	80.2%	81.2%	83.1%	82.6%	86.4%	89.2%	0.8
	D&D	76.3%	91.7%	102.9%	83.2%	89.8%	96.8%	98.2%	0.78
	PROTEINS	62.5%	94.7%	98.6%	98.6%	101.3%	102.4%	103.2%	0.76
GT-2	PATTERN	59.6%	106.8%	58.1%	58.6%	84.6%	100.7%	100.7%	0.87
GT-1	CLUSTER	55.4%	55.4%	61.0%	60.2%	71.1%	100.0%	100.1%	0.9

(b) GAT

		no-node-features	node-degree	no-edges	fully-connected	fragmented k=1	fragmented k=2	fragmented k=3	original AUROC
GT-1	CLUSTER	55.7%	55.9%	61.3%	55.9%	71.8%	100.0%	100.0%	0.9
GT-2	PATTERN	100.3%	101.1%	55.2%	55.2%	84.0%	100.6%	100.5%	0.92
	IMDB-BINARY	100.7%	105.7%	64.8%	68.3%	97.8%	100.5%	100.5%	0.79
GT-3	MNIST	72.8%	72.1%	100.5%	70.9%	97.1%	98.0%	98.8%	0.99
	CIFAR10	68.0%	67.8%	103.4%	84.7%	98.4%	98.0%	98.2%	0.89
GT-4	D&D	91.6%	91.6%	106.7%	112.4%	86.8%	94.8%	96.2%	0.76
	PROTEINS	96.0%	93.2%	99.4%	108.3%	98.2%	101.3%	103.8%	0.75
	NCI1	94.7%	95.5%	83.1%	87.2%	83.1%	85.8%	88.8%	0.88
	ENZYMES	81.9%	80.2%	84.0%	91.2%	78.9%	84.8%	90.5%	0.78

(c) GIN

		no-node-features	node-degree	no-edges	fully-connected	fragmented k=1	fragmented k=2	fragmented k=3	original AUROC
GT-1	CLUSTER	55.2%	55.2%	61.0%	60.9%	71.6%	99.9%	100.0%	0.9
GT-2	PATTERN	100.2%	100.3%	54.5%	55.1%	83.4%	100.1%	100.1%	0.93
	IMDB-BINARY	100.1%	112.5%	64.8%	67.1%	76.2%	99.3%	99.9%	0.75
GT-4	NCI1	94.3%	95.5%	83.2%	86.7%	83.4%	85.8%	89.3%	0.88
	ENZYMES	82.2%	82.3%	79.3%	87.5%	79.8%	83.9%	88.1%	0.81
GT-3	MNIST	69.7%	69.5%	100.0%	100.0%	99.9%	100.0%	100.0%	1
	CIFAR10	64.5%	64.3%	98.0%	98.9%	98.5%	99.1%	99.4%	0.94
GT-4	D&D	90.4%	88.9%	102.6%	104.3%	91.4%	92.2%	98.7%	0.77
	PROTEINS	90.7%	95.5%	100.1%	108.0%	99.7%	100.7%	102.8%	0.75

(d) ChebNet

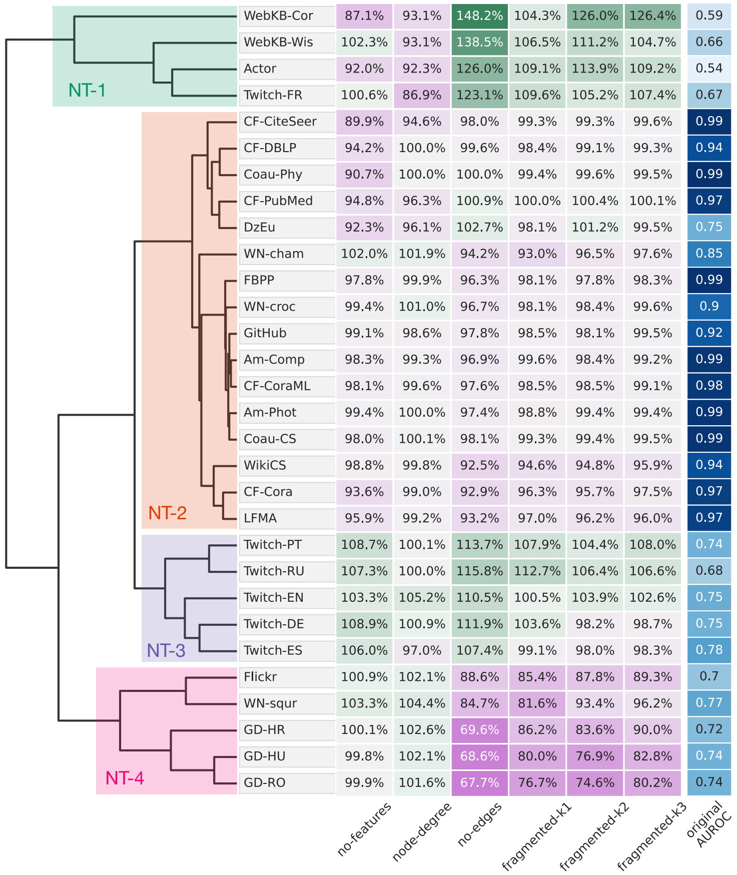
Figure 2: **Taxonomy of inductive graph learning datasets via graph perturbations.** For each GNN model (a–d), dataset, and perturbation combination, we show the model's performance relative to its performance on the unmodified dataset. The categorization into 4 dataset clusters is stable across various GNN models with only minor deviation.

In the latter two, which compose GT-2, the node features bear no importance: using the node degrees instead improves the performance in IMDB-BINARY by 20%, possibly by enhancing structural information that the prediction task depends on. The profile of CLUSTER differs from the other two in that it shows reliance on both types of information; in fact, in this dataset, the node features encode class information of a few key nodes, while maintaining graph structure is essential for correct cluster prediction of the unlabelled nodes. Note that original AUROC is retained for *fragmented* $k = \{2, 3\}$, as due to the dense nature of the graph 2-hop neighborhoods for each node recover the original graph.

Finally, the (bio)molecular datasets NCI1 [22], D&D [7], PROTEINS, and ENZYMES [2] are grouped into the cluster GT-4. D&D and PROTEINS show limited or no reliance on graph structure, as opposed to NCI1 and ENZYMES. Interestingly, even though PROTEINS and ENZYMES contain similar protein structure graphs, their different prediction tasks lead to notably different perturbation sensitivities; this illustrates the task dependency of the optimal graph representations.

3.2 Node-level prediction tasks

For node-level prediction tasks we restrict our experiments to a GCN model as our experiments for the graph-level tasks suggest the GNN model of choice does not have a profound impact on the resulting taxonomy. We identify four groups of node-level datasets. First, NT-1 and NT-3 contain all the datasets from Twitch [18], WebKB [5], and Actor [17]. Both groups, particularly NT-1, benefit from removing all edges, indicating the richness of the constructed node features for the corresponding tasks without needing additional structural information. WebKB aims to classify web pages into, e.g., student or staff. One may imagine the web page descriptions (node features) are informative for distinguishing student and staff pages since, for example, the word "Ph.D." is more likely to appear on a student page. Meanwhile, it is less likely that student pages link between each other (edges), as suggested by the heterophile nature of the WebKB networks [16]. Notice the NT-3 datasets also benefit from removing node features instead of removing edges, as opposed to NT-1. This difference indicates the Twitch datasets in NT-3 contain node features and structural information that are individually useful for the prediction tasks, but collectively work against each other.



	no-features	node-degree	no-edges	fragmented-k1	fragmented-k2	fragmented-k3	original AUROC
WebKB-Cor	87.1%	93.1%	148.2%	104.3%	126.0%	126.4%	0.59
WebKB-Wis	102.3%	93.1%	138.5%	106.5%	111.2%	104.7%	0.66
Actor	92.0%	92.3%	126.0%	109.1%	113.9%	109.2%	0.54
Twitch-FR	100.6%	86.9%	123.1%	109.6%	105.2%	107.4%	0.67
CF-CiteSeer	89.9%	94.6%	98.0%	99.3%	99.3%	99.6%	0.99
CF-DBLP	94.2%	100.0%	99.6%	98.4%	99.1%	99.3%	0.94
Coau-Phy	90.7%	100.0%	100.0%	99.4%	99.6%	99.5%	0.99
CF-PubMed	94.8%	96.3%	100.9%	100.0%	100.4%	100.1%	0.97
DzEu	92.3%	96.1%	102.7%	98.1%	101.2%	99.5%	0.75
WN-cham	102.0%	101.9%	94.2%	93.0%	96.5%	97.6%	0.85
FBPP	97.8%	99.9%	96.3%	98.1%	97.8%	98.3%	0.99
WN-croc	99.4%	101.0%	96.7%	98.1%	98.4%	99.6%	0.9
GitHub	99.1%	98.6%	97.8%	98.5%	98.1%	99.5%	0.92
Am-Comp	98.3%	99.3%	96.9%	99.6%	98.4%	99.2%	0.99
CF-CoraML	98.1%	99.6%	97.6%	98.5%	98.5%	99.1%	0.98
Am-Phot	99.4%	100.0%	97.4%	98.8%	99.4%	99.4%	0.99
Coau-CS	98.0%	100.1%	98.1%	99.3%	99.4%	99.5%	0.99
WikiCS	98.8%	99.8%	92.5%	94.6%	94.8%	95.9%	0.94
CF-Cora	93.6%	99.0%	92.9%	96.3%	95.7%	97.5%	0.97
LFMA	95.9%	99.2%	93.2%	97.0%	96.2%	96.0%	0.97
Twitch-PT	108.7%	100.1%	113.7%	107.9%	104.4%	108.0%	0.74
Twitch-RU	107.3%	100.0%	115.8%	112.7%	106.4%	106.6%	0.68
Twitch-EN	103.3%	105.2%	110.5%	100.5%	103.9%	102.6%	0.75
Twitch-DE	108.9%	100.9%	111.9%	103.6%	98.2%	98.7%	0.75
Twitch-ES	106.0%	97.0%	107.4%	99.1%	98.0%	98.3%	0.78
Flickr	100.9%	102.1%	88.6%	85.4%	87.8%	89.3%	0.7
WN-squr	103.3%	104.4%	84.7%	81.6%	93.4%	96.2%	0.77
GD-HR	100.1%	102.6%	69.6%	86.2%	83.6%	90.0%	0.72
GD-HU	99.8%	102.1%	68.6%	80.0%	76.9%	82.8%	0.74
GD-RO	99.9%	101.6%	67.7%	76.7%	74.6%	80.2%	0.74

Figure 3: **Taxonomy of transductive node-level prediction datasets.** Categorization into 4 dataset groups (NT-1 to NT-4) is based on clustering of sensitivity profiles w.r.t. a GCN-based model.

NT-2 contains a broad spectrum of datasets from citation networks to social networks and web pages, which are relatively insensitive to any graph perturbation. This implies that both the node features and the structural information are useful for the tasks; and unlike NT-3, the two sources of information are consistent with each other. This also accounts for the relatively high AUROC scores of NT-2 datasets (averaging 0.956) compared to NT-3 (averaging 0.740). As an example, Amazon [20] aims to classify the categories of products (nodes) given their co-purchasing relationships (edges) and the product descriptions (node features). One may expect that both sources of information are insightful for this classification task. Notice the citation networks (CF [1]) and the streaming service social networks (DzEu, LFMA [19]) are slightly more dependent on the node features, indicating the node features are somewhat more informative. Finally, NT-4 datasets are dominantly reliant on the structural information, with Flickr [26] being particularly dependent on long-range interactions, as suggested by the same performance drop caused by the edge removal and all *fragmented* perturbations.

4 Conclusion

We provide a principled approach for the taxonomization of graph datasets based on their type of prediction task signal, rather than application domain. We studied inductive and transductive classification tasks without edge features or global graph properties; for future work we plan to extend our analysis to a wider range of graph tasks, datasets, and GNN models. Our taxonomy is determined by the set of perturbations and the expressive power of the GNN models used. Combining a wider range of these views will lead to a more complete taxonomy, potentially leading to a better understanding of the modeling capabilities of GNNs as well as providing valuable insights required to improve upon existing models.

References

[1] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *Proc. of ICLR*, 2018.

[2] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21:i47–i56, 2005.

[3] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv:2104.13478*, 2021.

[4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, Jul 2017.

[5] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1):69–113, 2000.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 3844–3852, 2016.

[7] Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, July 2003.

[8] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking Graph Neural Networks. *arXiv:2003.00982*, 2020.

[9] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[10] William L Hamilton. *Graph Representation Learning*. Morgan & Claypool Publishers, 2020.

[11] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *Adv. in NeurIPS 33*, 2020.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*, 2017.

[13] Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering GCN: Overcoming Oversmoothness in Graph Convolutional Networks. In *Adv. in NeurIPS 33*, pages 14498–14508, 2020.

[14] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Graph Representation Learning and Beyond (GRL+) Workshop*, 2020.

[15] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609, 2019.

[16] Hesham Mostafa, Marcel Nassar, and Somdeb Majumdar. On local aggregation in heterophilic graphs. *arXiv:2106.03213*, 2021.

[17] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-GCN: Geometric graph convolutional networks. In *Proc. of ICLR*, 2020.

[18] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[19] Benedek Rozemberczki and Rik Sarkar. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1325–1334, 2020.

[20] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *Relational Representation Learning Workshop (R2L 2018), NeurIPS*, 2018.

[21] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *The 6th International Conference on Learning Representations (ICLR)*, 2018.

[22] Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.

[23] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[24] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *Proc. of ICLR*, 2019.

[25] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015.

[26] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. GraphSAINT: Graph sampling based inductive learning method. In *Proc. of ICLR*, 2020.

[27] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.