Rethinking the Effectiveness of Graph Classification Datasets in Benchmarks for Assessing GNNs

Zhengdao Li[1,2], **Yong Cao**[3], **Kefan Shuai**[2], **Yiming Miao**[2*] and **Kai Hwang**[2]
[1]Guangzhou University, Guangzhou, China
[2]The Chinese University of Hong Kong, Shenzhen, China
[3]Huazhong University of Science and Technology, Wuhan, China

Abstract

Graph classification benchmarks, vital for assessing and developing graph neural networks (GNNs), have recently been scrutinized, as simple methods like MLPs have demonstrated comparable performance. This leads to an important question: Do these benchmarks effectively distinguish the advancements of GNNs over other methodologies? If so, how do we quantitatively measure this effectiveness? In response, we first propose an empirical protocol based on a fair benchmarking framework to investigate the performance discrepancy between simple methods and GNNs. We further propose a novel metric to quantify the dataset effectiveness by considering both dataset complexity and model performance. To the best of our knowledge, our work is the first to thoroughly study and provide an explicit definition for dataset effectiveness in the graph learning area. Through testing across 16 real-world datasets, we found our metric to align with existing studies and intuitive assumptions. Finally, we explore the causes behind the low effectiveness of certain datasets by investigating the correlation between intrinsic graph properties and class labels, and we developed a novel technique supporting the correlation-controllable synthetic dataset generation. Our findings shed light on the current understanding of benchmark datasets, and our new platform could fuel the future evolution of graph classification benchmarks.

1 Introduction

Graph Neural Networks (GNNs) have exhibited superior performance in various domains, including recommendation system [Wu *et al.*, 2022], molecule property prediction [Wieder *et al.*, 2020], and natural language processing [Wu *et al.*, 2021], etc. To evaluate GNN models in these tasks, specific datasets are often selected as benchmark datasets. Given this mission, a high-quality benchmark dataset should be capable to differentiate the advancements of diverse models. For example, current available benchmarks, such as OGB [Hu *et al.*, 2020], TUDataset [Morris *et al.*, 2020], etc., serve for various link-wise, node-wise, and graph-wise tasks evaluation, as well as graph classification, aiming to automatically discover the optimal method for given tasks. These datasets and benchmark frameworks have immensely facilitated GNN research.

However, recent studies [Errica *et al.*, 2020; Zhao and Wang, 2019; Hu *et al.*, 2020; Dwivedi *et al.*, 2023; Morris *et al.*, 2020] have shown that GNNs may not consistently surpass other baseline methods in specific graph classification tasks. Some simple baseline methods can achieve performance similar to GNNs, and sometimes even better. For example, in the benchmark of [Errica *et al.*, 2020], the MoleculeFingerprint baseline outperforms significantly the widely used GNN models such as GIN [Xu *et al.*, 2018], GraphSage [Hamilton *et al.*, 2017] on three out of four molecular datasets. Nevertheless, despite current research primarily having made significant achievements in analyzing the theoretical expressive power of GNNs [Xu *et al.*, 2019; Feng *et al.*, 2022; Wang and Zhang, 2022] and training schemes [Duan *et al.*, 2022], the reasons for GNNs' failures from these evidences have not been thoroughly analyzed. Few researchers are paying attention to the issues inherent in the datasets themselves.

Therefore, we have adopted a different perspective: dataset compatibility. Our investigation focuses on whether the datasets themselves are suitable for evaluating the advancements of Graph Neural Networks (GNNs) compared to other methods. This aspect is critical for a fair assessment of whether a GNN method has truly shown improvement. Studies in neural language processing, like [Xiao *et al.*, 2022], define effectiveness as the performance variance across different methods. However, this definition is not directly applicable to graph classification problems. For example, in binary classification problems versus 10-class classification problems, the absolute values of variance are not directly comparable. Hence, in our paper, we reevaluate the effectiveness of existing datasets and attempt to address the following two questions:

RQ1: Can commonly used graph classification datasets serve the benchmarking purpose which is to effectively distinguish advancements of GNNs compared with other methods? To address this question, we propose an empirical protocol (Sec.2.1) to investigate the performance disparity

*Corresponding author

between baseline methods and GNN-based methods in terms of the structures and attributes separately by restricting the information input types, i.e., structural information or attributed information. Specifically, we re-organize 16 real-world datasets (Sec.2.2) from common benchmarks across diverse scales and application domains, and conduct extensive experiments with the proposed protocol to investigate the performance gaps fairly on a well-developed benchmark framework[1] extended from [Errica *et al.*, 2020], with our new improvements: 1) supporting datasets from other benchmarks such as OGB, TUDatasets, and synthetic datasets. 2) supporting the construction and combination of various artificial node features, not limited to the framework proposed by [Cui *et al.*, 2022], helps to investigate the impact of different information inputs on the performance of GNNs.

RQ2: How to measure the effectiveness of existing graph classification datasets? To answer this question, we design a novel metric (Sec.3.1) to quantify the effectiveness of diverse classification datasets by normalizing the performance gap into a scale-free quantity, with the consideration of the prediction difficulty of datasets and diversity in the number of class labels. The fairness and efficacy of the metric are justified on 16 datasets. For further exploration of the causes of the low effectiveness of datasets, we investigate the relationships between basic graph properties and class labels, and develop a novel approach (Sec.4.2) for generating controllable synthetic datasets, which enables precise control over the degree of correlations between graph properties and class labels. This allows us to study the effectiveness in a controlled environment, providing deeper insights across varying conditions. Additionally, inspired by [Xiao *et al.*, 2022], we further develop a straightforward yet effective regression method to predict the effectiveness (Sec.4.3) of a given dataset.

2 Empirical Studies of Existing Graph Classification Datasets

This section delves into empirical studies of existing graph classification datasets, offering a clear, concise, and engaging overview. Initially, we establish a straightforward and insightful evaluation measurement, utilizing diverse datasets to gain empirical insights that align with findings from other research. However, this simple measurement has its limitations. To address these shortcomings, we will introduce a novel metric designed specifically to overcome these limitations in Section 3.

2.1 An Empirical Protocol for Evaluating Dataset Discriminability

We propose a protocol that can fairly evaluate the ability of a graph classification dataset for discriminating the advancements of graph-aware methods including GNNs and graph-kernel based approches over baselines. An effective strategy is to evaluate the performance gap between them. If the performance of graph-aware methods and simple baselines exhibit similarity, it indicates that the dataset lacks the necessary discriminatory power, thus questioning its suitability as a benchmark. The protocol encompasses three main components: (1) the baselines and GNNs for classification; (2) the evaluation framework; and (3) the performance gaps as determined by the evaluation framework. In the rest of this section, we will delve into these three key components and introduce some notations for further usage.

Evaluation Framework. The framework is built upon the benchmarking framework proposed by [Errica *et al.*, 2020], the detailed architecture can be found in the supplementary. This framework leverages risk assessment and model selection schemes to provide a fair comparison of GNN models using a k-fold cross-validation procedure for model assessment. Each validation procedure incorporates a model selection process with varying hyperparameters. We further enhance this basic framework in the following ways:

(1) We expand the dataset splitting schemes to support additional strategies, such as the molecular scaffold splitting scheme and user-defined splitting schemes. These offer meaningful, domain-aware splits as opposed to random splits.

(2) Our framework allows for the loading of datasets from various sources, including PyTorch Geometric, Open Graph Benchmark (OGB), as well as user-defined synthetic datasets.

(3) Drawing inspiration from the studies [Cui *et al.*, 2022], we have equipped our framework to accept various compositions of graph-level or node-level statistical features as model input, moving beyond the support for only single node labels or edge labels.

Assessment of performance gap. GNNs are superior to other neural network structures on graph data because of their ability to capture structure information. However, performance gaps in previous works fail to distinguish the effects of structure and attribute. To solve this problem, we decouple the performance gap into the *structural gap* and *attributed gap*. *Structural performance gap* is denoted by $\delta_{\mathbf{S}}$. It measures the difference in classification accuracy between a structure-dominated baseline and the best performance achieved by structure-aware methods without any attributed information, including graph-kernel based approaches and GNNs with artificial node attributes as input features. (e.g., node degree and random noise). *Attributed performance gap* is denoted by $\delta_{\mathbf{A}}$. It quantifies the accuracy difference between an attribute-dominated baselines that vary across applications and GNNs that utilize real node or edge attributes as input features. Note that GNNs with real node or edge attributes inevitably involve a part of structural information. A formal performance gap is given by the Definition 1.

Definition 1. *Given a dataset D, a baseline method $\mathcal{M}_{\boldsymbol{type}}^{Baseline}$, and a graph-based method $\mathcal{M}_{\boldsymbol{type}}^{Graph}$, the performance gap $\delta_{\boldsymbol{type}}(D, R, \mathcal{M}^{Graph}, \mathcal{M}^{Baseline})$ (simply denoted by $\delta_{\boldsymbol{type}}$) between baseline and graph-based method is defined as:*

$$\delta_{\boldsymbol{type}} \triangleq R(D, \mathcal{M}_{\boldsymbol{type}}^{Graph}) - R(D, \mathcal{M}_{\boldsymbol{type}}^{Baseline}),\ \boldsymbol{type} \in \{\boldsymbol{S}, \boldsymbol{A}\},$$

where $R(D, \mathcal{M})$ is the numerical value of a given evaluation metric such as mean classification accuracy or mean AUC-ROC (Area Under the Receiver Operating Characteristics), obtained by model $\mathcal{M}$ on dataset D.

[1] https://github.com/ICLab4DL/GNNBenchEffectiveness

Choices of baselines. Following the insights from [Cui *et al.*, 2022; Errica *et al.*, 2020], we categorize baselines into two types based on input information: *structure-dominated baselines* and *attribute-dominated baselines*. This classification helps in delivering a detailed analysis as node attributes and structures contribute differently to model performance across datasets. For structure-dominated baselines, we use shallow MLPs with average graph degrees as input, denoted as $\mathcal{M}_{\mathbf{S}}^{\text{Baseline}}$. For attribute-dominated baselines in attribute-graph datasets, we encode molecules as per [Hu *et al.*, 2020] and use the MoleculeFingerprint model for classification, while non-attribute graph datasets employ a combination of pooling layers and shallow MLPs. These are represented as $\mathcal{M}_{\mathbf{A}}^{\text{Baseline}}$.

Choices of graph-aware methods. In our experiments, we carefully choose the $\mathcal{M}_{\mathbf{type}}^{\text{Graph}}$ and $\mathcal{M}_{\mathbf{type}}^{\text{Baseline}}$. We use a diverse range of graph-based methodologies, including GNN models and graph-kernel based methods for $\mathcal{M}_{\mathbf{type}}^{\text{Graph}}$. Specifically, we employed the Graph Isomorphism Network (GIN) for its spatial approach, which focuses on the physical layout of the graph, accentuating local structures and node-level relationships. Concurrently, the Graph Convolutional Network (GCN) was chosen for its spectral method approximation, where node features are transformed into the spectral domain using the graph Fourier transform, allowing for a global analysis of the overall graph structure and relationships. Additionally, we utilized graph-kernel based methods including Weisfeiler-Lehman graph kernel (WL-GK) [Shervashidze *et al.*, 2011], the Subgraph-Matching kernel (SM-GK) [Kriege and Mutzel, 2012], and the Shortest-Path kernel (SP-GK) [Borgwardt and Kriegel, 2005]. This selection of both spatial (GIN) and spectral (GCN) methods, complemented by kernel-based techniques, provides a comprehensive and balanced evaluation of local and global graph features, crucial for the depth and breadth of our study in graph-based machine learning.

The performance gaps $\delta_{\mathbf{S}}$ and $\delta_{\mathbf{A}}$ can indicate the discriminating ability of the dataset and provide insights into it. In particular, a narrow gap with high accuracy from both methods suggests the dataset may be too simple to offer discrimination. Conversely, low accuracy from both methods implies the dataset's information is underutilized, necessitating a more advanced approach. A large gap indicates the dataset's strong discriminative power for these two models.

2.2 Collection of Diverse Datasets

Bio&Chem. In the fields of biology and chemistry, the ability to predict molecular properties, such as toxicity or biological activity of proteins plays a pivotal role in drug discovery and development. Datasets such as MUTAG, D&D [Yanardag and Vishwanathan, 2015], PROTEINS, and NCI1 furnish a wealth of information for constructing and training machine learning models in these disciplines. Similarly, in chemical research, datasets like HIV and ENZYMES are indispensable for decoding the interactions between chemical compounds and their potential impacts on living organisms. The large-scale PPA dataset facilitates an understanding of intricate protein interactions and functions, significantly contributing to advancements in personalized medicine and therapeutic approaches.

Domain	Dataset	Graphs	Classes	Average nodes	Features
Bio&Chem	BACE♡	1513	2	34.09	9|-
	Tox21♡	7831	2	18.57	9|3
	HIV♡	41,127	2	25.5	9|3
	PPA♡	158,100	37	243.4	-|7
	MUTAG★	188	2	17.9	7|-
	NCI1★	4,110	2	29.8	37|-
	PROTEINS★	1,113	2	39.1	3|-
	AIDS★	2000	2	15.69	38|-
	DD★	41,127	2	25.5	9|3
	ENZYMES★	600	6	32.6	3|-
Social science	IMDB-B★	1,000	2	19.77	-
	IMDB-M★	1,500	3	13	-
	REDDIT-B★	2,000	2	429.61	-
	COLLAB★	5,000	3	74.49	-
CV	MNIST■	55,000	10	70.6	3|-
	CIFAR10■	45,000	10	117.6	5|-

Table 1: Summary of datasets with different scales, feature types and classification numbers in our experiments.

Social science. In the domain of social science, datasets like IMDB-Binary (IMDB-B), IMDB-Multi (IMDB-M), REDDIT-Binary (REDDIT-B), COLLAB are used to study and understand various aspects of social interactions and behaviors.

Computer vision (CV). The MNIST and CIFAR10 have been fundamental in shaping the field of computer vision, offering a wide range of images for tasks like object recognition and classification. These two datasets can verify the positional learning ability of GNNs, as the samples are transformed from images into graphs with the super-pixels and coordinates as the node features that inherently carry the positional information of each node.

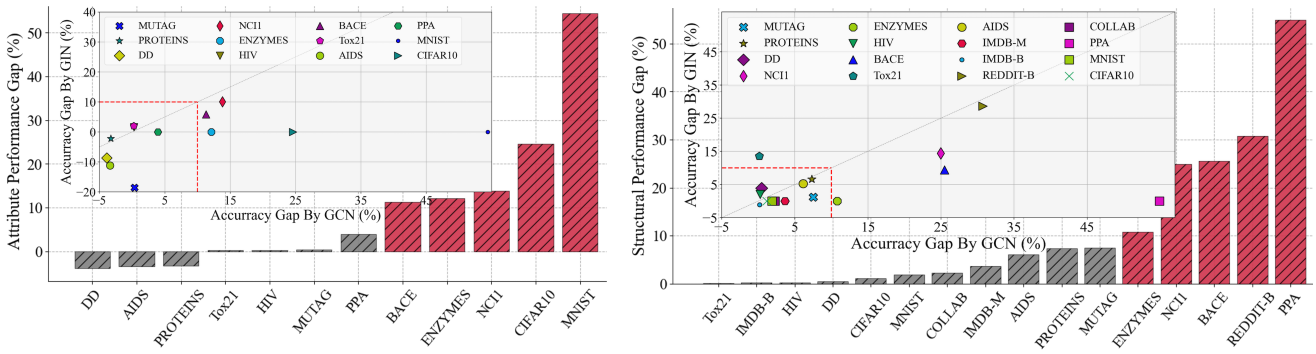
2.3 Observations of Performance Gaps on 16 Real-world Datasets

Experimental setup. Utilizing our proposed framework, we assessed 16 real-world datasets following the standard protocol. In Table 2, we show the main experimental results obtained by the protocol over 16 real-world datasets, in which 14 datasets except for PPA and Tox21 were tested by 10-fold cross-validation. The baselines, GNNs and graph kernel methods are introduced in Sec2.1. (Note that, NA denotes the dataset with no attributes, - denotes the dataset is too large to run.) The values that are both bolded and underlined represent the highest accuracy across all attributed and structural models, the solely bolded values indicate the highest accuracy within one type of models. As observed from this table, GNNs excel as the state-of-the-art (SOTA) on the majority of datasets. However, it's important to highlight that the performance gap between the baseline methods and GNNs is minimal for approximately half of the datasets, which is visually represented in Figure 1.

In Figure 1(a), we depict the highest attributed accuracy gap $\delta_{\mathbf{A}}$, comparing the GIN and GCN models. For molecular and protein datasets (HIV, PPA, BACE, and Tox21), we employed

Dataset	$\mathcal{M}_{\mathbf{A}}^{\text{Baseline}}$	$\mathcal{M}_{\mathbf{A}}^{\text{GIN}}$	$\mathcal{M}_{\mathbf{A}}^{\text{GCN}}$	$\mathcal{M}_{\mathbf{S}}^{\text{Baseline}}$	$\mathcal{M}_{\mathbf{S}}^{\text{GraphKernel}}$	$\mathcal{M}_{\mathbf{S}}^{\text{GIN}}$	$\mathcal{M}_{\mathbf{S}}^{\text{GCN}}$
MUTAG	83.7 ± 8.35	**84.07** ± 6.26	70.7 ± 6.89	79.18 ± 9.83	86.23 ± 8.50	**86.71** ± 4.67	82.86 ± 10.43
PROTEINS	**74.24** ± 3.09	70.97 ± 3.79	73.28 ± 3.22	60.95 ± 0.79	**72.50** ± 2.58	68.24 ± 4.39	64.29 ± 2.6
HIV	96.58 ± 0.1	**96.86** ± 0.13	96.69 ± 0.06	96.49 ± 0.01	51.00 ± 0.00	**96.74** ± 0.09	96.49 ± 0.09
PPA	20.12 ± 0.0	**24.05** ± 0.0	16.08 ± 0.0	9.28 ± 0.0	-	64.19 ± 0.0	**66.72** ± 0.0
D&D	**76.12** ± 2.78	72.22 ± 3.18	70.49 ± 2.13	62.29 ± 2.55	62.39 ± 1.89	62.73 ± 2.23	**64.03** ± 2.64
ENZYMES	29.67 ± 5.74	**41.78** ± 3.92	31.72 ± 4.54	17.56 ± 1.93	25.00 ± 3.33	**28.33** ± 4.24	22.0 ± 3.48
NCI1	66.76 ± 1.98	**80.54** ± 1.16	76.85 ± 2.78	50.58 ± 1.02	62.50 ± 1.79	**75.55** ± 1.31	65.03 ± 2.43
BACE	68.64 ± 4.68	**79.95** ± 2.9	73.5 ± 3.48	54.33 ± 0.17	61.84 ± 0.00	**79.82** ± 3.18	61.32 ± 5.07
AIDS	**99.07** ± 0.85	95.68 ± 1.53	90.05 ± 2.27	89.2 ± 1.16	**99.55** ± 0.52	95.33 ± 1.16	86.65 ± 2.24
moltox21	91.05 ± 0.0	**91.31** ± 0.0	90.88 ± 0.0	90.43 ± 0.0	-	**90.53** ± 0.0	90.45 ± 0.0
IMDB-B	NA	NA	NA	70.63 ± 3.57	67.10 ± 4.76	**70.8** ± 2.81	69.5 ± 2.94
IMDB-M	NA	NA	NA	42.31 ± 4.54	**47.00** ± 5.84	45.93 ± 4.19	44.98 ± 4.78
REDDIT-B	NA	NA	NA	58.33 ± 1.18	73.50 ± 2.05	**89.05** ± 2.14	86.98 ± 2.52
COLLAB	NA	NA	NA	67.68 ± 0.94	63.92 ± 1.63	**69.92** ± 1.09	69.79 ± 1.11
MNIST	24.1 ± 0.33	**78.48** ± 0.72	54.03 ± 2.15	9.86 ± 0.01	-	11.74 ± 1.49	**21.93** ± 0.35
CIFAR10	25.27 ± 0.6	**49.87** ± 0.4	45.75 ± 0.6	10.0 ± 0.0	-	11.13 ± 0.99	**13.7** ± 0.62

Table 2: Mean test accuracy and variations of different methods in 16 graph classification datasets.



(a) Attributed accuracy gap

(b) Structural accuracy gap

Figure 1: The performance gaps on 16 graph classification datasets are categorized into two types: *Ineffective* (gray) and *Effective* (red) benchmarks. These are sorted in ascending order based on the size of the performance gap. An empirical threshold of 10% is used for categorization, as observed in the inner box of each figure. This box represents the distribution of the accuracy gap for GCN and GIN.

a baseline model formed by AtomEncoder [Hu *et al.*, 2020] and MolecularFingerprint [Errica *et al.*, 2020], and solely the MolecularFingerprint model for the other datasets. Subplots provide further comparisons of $\delta_{\mathbf{A}}$ of GIN and GCN across different datasets. Likewise, Figure 1(b) reveals the greatest structural accuracy gap ($\delta_{\mathbf{S}}$) among GIN, GCN, SP-GK, WL-GK, and SM-GK approaches.

From our experimental results, the following observations and insights are derived:

Observation 1. Most datasets excel in either attributed or structural performance gaps. Computer vision datasets MNIST and CIFAR10 showcase significant attributed performance gaps, attributable to their dependency on positional and color information of target nodes. Chemical datasets like PPA and Tox21 display noteworthy structural performance gaps due to the inadequacy of average degree information for baseline model predictions, consistent with prior findings [Dwivedi *et al.*, 2023; Cui *et al.*, 2022; Errica *et al.*, 2020; Hu *et al.*, 2020].

Observation 2. Datasets displaying huge gaps for both $\delta_{\mathbf{S}}$ and $\delta_{\mathbf{A}}$, like ENZYMES, BACE, and NCI1, reinforce the importance of structures and specific subgraph functions in molecules and compounds. GNNs demonstrate superior performance across most of the datasets by effectively capturing both attributed and structural information simultaneously.

Observation 3. Interestingly, among the social science datasets, only REDDIT-B displayed a noteworthy performance gap, indicating a weak correlation between degree information and task labels. This intriguing observation will be further explored and investigated in subsequent sections.

2.4 Limitations of Using Performance Gap as Effectiveness Measurement

In general, half of the 16 graph classification datasets may not be effective to discriminate baselines and GNNs, by using the absolute value of performance gaps. It is important to note that solely relying on the absolute performance gap as an indicator to assess dataset effectiveness could potentially lead to some unfairness and overlook certain inherent limitations.

For instance, two binary classification datasets D_1 and D_2

have the same performance gap such as 10%, while for D_1, the $R(D_1, \mathcal{M}^{\text{Baseline}}) = 80\%$, $R(D_1, \mathcal{M}^{\text{GNN}}) = 90\%$, and for D_2, the $R(D_2, \mathcal{M}^{\text{Baseline}}) = 50\%$, $R(D_2, \mathcal{M}^{\text{GNN}}) = 60\%$. It is obvious that the D_2 has more complex characteristics leading to the failures of both methods. In that case, we prefer that D_2 has more potential performance improvements by using advanced methods, such that it has larger effectiveness. Another limitation is the lack of consideration of the number of class labels. Suppose that D_1 has 2 labels, and the D_2 has 10 labels, the complexity of datasets is different even when the $R(D_1, \mathcal{M}^{\text{Baseline}}) = R(D_2, \mathcal{M}^{\text{Baseline}}) = 80\%$, and $R(D_1, \mathcal{M}^{\text{GNN}}) = R(D_2, \mathcal{M}^{\text{GNN}}) = 90\%$.

Therefore, in the following section, we will deliberate on the determination of a dataset's suitability for benchmarking purposes and introduce a novel, unified metric designed to measure this degree. This metric takes into account not only the inherent complexity of the dataset and number of class labels, but also the absolute performance gaps observed among different approaches.

3 Quantifying the Effectiveness of Benchmark Datasets

In this section, we introduce our proposed metric designed to address the limitations of the performance gap (Definition 1). This metric quantifies the degree to discriminate the ability of different methods. We then demonstrate the properties of this new metric and validate it on 16 benchmarks, providing answers to research questions RQ1 and RQ2.

3.1 Dataset Effectiveness

The new metric is defined as follows and is simply denoted by $\mathcal{E}(D)$. We refer to it as Effectiveness over a dataset D.

Definition 2. *Given a graph classification dataset D which has $|Y|$ classes, and the performance gap $\delta_{type}(D)$ between two methods $\mathcal{M}^1$ and $\mathcal{M}^2$, the $\mathcal{E}$ to quantify the discriminating degree of $\mathcal{M}^1$ and $\mathcal{M}^2$ is defined as follows:*

$$\mathcal{E}(D) = \sum_{type \in \{S,A\}} \frac{|\delta_{type}(D)|}{R^*(|Y|-1)} \cdot \frac{1-R^*}{1-|Y|^{-1}}, \qquad (1)$$

where $R^* = \min(R1, R2)$, which is the minimal value of two accuracy values from $\mathcal{M}^1$ and $\mathcal{M}^2$, denoted by $R1$ and $R2$ respectively.

The Definition 2 aggregates two types of effectiveness, each type of effectiveness is the product of two components. The first component $\frac{|\delta_{\text{type}}(D)|}{R^*(|Y|-1)}$ is the absolute changing proportion of the performance gap which is normalized by the total number of class labels $|Y|-1$. This component varies from 0 to 1, if the worst performance is not less than random guessing.

The second component $\frac{1-R^*}{1-|Y|^{-1}}$, termed the complexity factor and denoted as λ, ranges between 0 and 1, indicating a dataset's relative complexity. The denominator, $|Y|^{-1}$, reflects random guessing accuracy, with $|Y|$ being the total task labels. The numerator represents the gap between the worst method and perfect classification. If the worst method's accuracy is near $|Y|^{-1}$, λ nears 1, indicating high complexity. If it's near 100%, λ is close to 0, suggesting a trivial dataset.

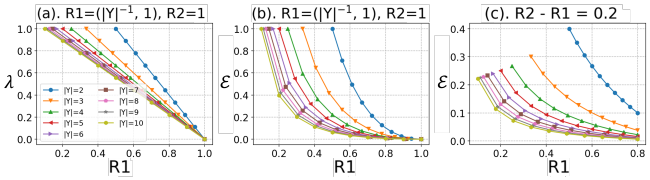


Figure 2: Properties illustration of λ and $\mathcal{E}$.

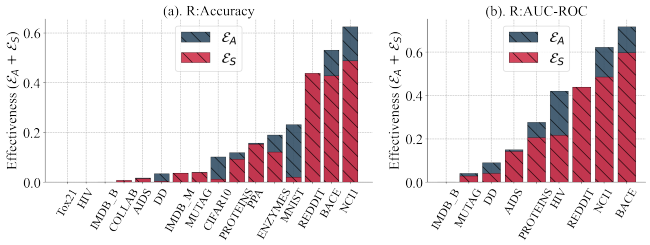


Figure 3: Effectiveness using Accuracy metric and AUC-ROC metric in terms of structural type and attributed type.

Note that, for binary datasets, R can be AUC-RUC or accuracy. This is because the AUC-ROC value for random guessing is 0.5, aligning with $1 - |Y|^{-1}$ when $|Y| = 2$.

3.2 Properties of Complexity Factor and Effectiveness

In this section, we delve into the properties of the complexity factor λ and how it manages dataset intricacy considering task label counts. Figure 2 elucidates properties of λ and effectiveness $\mathcal{E}$ via variations in Eq. 1.

Property 1: As the worst method accuracy rises, λ linearly decreases (Figure 2(a)). Each curve represents a dataset with task labels from 2 to 10. Essentially, a higher worst method accuracy means a simpler dataset.

Property 2: A smaller performance gap leads to a reduced $\mathcal{E}$ (Figure 2(b)). As the gap decreases, the dataset's distinguishability diminishes.

Property 3: With a constant performance gap, $\mathcal{E}$ varies based on the worst performance values of $R1$ and $R2$ (Figure 2(c)). Higher accuracies yield a lower $\mathcal{E}$ than lower accuracies. For instance, a 20% difference in accuracy between two methods results in a higher $\mathcal{E}$ if the accuracies are lower.

Property 4: For datasets D_1 and D_2 with the same performance gaps and accuracy, if $|Y_1| < |Y_2|$, then $\mathcal{E}(D_1) > \mathcal{E}(D_2)$ (Figure 2(a-c)). A dataset with more classes has a larger $\mathcal{E}$.

3.3 Effectiveness of Real-world Datasets

We examined the effectiveness $\mathcal{E}$ of 16 real-world datasets using our protocol. Figure 3(a) shows the attributed effectiveness $\mathcal{E}\mathbf{A}$ (in grey) and structural effectiveness $\mathcal{E}\mathbf{S}$ (in red) for all datasets. In Figure 3(b), we assess $\mathcal{E}$ for binary datasets using the AUC-ROC metric. While $\mathcal{E}$ values are consistent across metrics for most datasets, HIV's $\mathcal{E}$ jumps from near 0 to 0.4 with AUC-ROC, emphasizing its suitability for evaluation. Generally, $\mathcal{E}$ remains stable across different metrics. The ranking by effectiveness aligns with

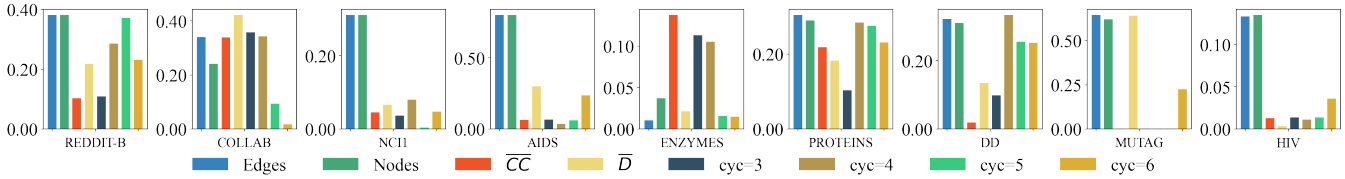


Figure 4: Correlations between graph property sequences and class labels on 9 real-world datasets.

the performance gap, confirming a high Spearman correlation between $\mathcal{E}$ sequences and their performance gap sequences.

In conclusion, by leveraging the definition of $\mathcal{E}$ across various metrics R and models $\mathcal{M}$, we can gain valuable insights. These insights aid in the assessment of a dataset's fairness and suitability for benchmarking purposes. Furthermore, this definition guides the selection of appropriate metrics and models for a given dataset, such as opting for accuracy or AUC-ROC as a metric.

4 Investigation of Causes of Low Effectiveness of Datasets

4.1 Correlation Between Graph Properties and Class Labels

Inspired by [Cui *et al.*, 2022; Errica *et al.*, 2020; Hu *et al.*, 2020], we hypothesize that for some simple graph properties, they are highly correlated with class labels. This high correlation is what allows simple methods to achieve good accuracy. Therefore, we first examine the correlation between certain simple graph properties and class labels.

Graph property sequence. We generate graph property sequences in terms of some basic graph properties such as number of nodes, average degree, count of cycles, etc. Suppose we have a non-attribute dataset $\mathbb{D}$ with N samples, i.e., $\mathbb{D} = \{g_i\}_{i=1}^{N}$, and the corresponding labels $\mathbb{Y} = \{y_i\}_{i=1}^{n}$, where $y_i \in \{0, 1\}$ for a binary classification dataset. Following this sample sequence, we can generate various corresponding graph property sequences. For instance, the average degree sequence, i.e., $\overline{D} = \{\overline{d}_i\}_{n=1}^{n}$, where $\overline{d}_i$ is the average degree of the graph sample g_i. Similarly, we construct the average clustering coefficient (CC) sequence, i.e., $\overline{CC} = \{\overline{cc}_i\}_{i=1}^{n}$, where $\overline{cc}_i$ is the average clustering coefficient of g_i. Besides these two basic properties, we obtain sequences of other different graph properties, i.e., edge count sequence (denoted by Edges), node count sequence (denoted by Nodes), cycle count sequence (denoted by cyc=k), where k represents the cycle length, $k \in \{3, 4, 5, 6\}$.

Correlation analysis between graph property sequences and label series. Figure 4 shows the correlations between 8 graph properties and labels $\mathbb{Y}$. The correlation of Edges and Nodes with $\mathbb{Y}$ exceeds 0.2 in most datasets, often above 0.4. In molecular datasets like MUTAG, cycle count is highly correlated with labels, indicating the impact of cyclic structures. Studies [Chen *et al.*, 2020; Rieck *et al.*, 2019; Bouritsas *et al.*, 2022] suggest WL kernels and GNNs struggle to capture substructures, underlining the importance of analyzing graph properties for method performance.

4.2 Controllable Synthetic Datasets

Real datasets are finite and insufficiently diverse for an exhaustive exploration of the effects of varying correlations between different graph properties and labels on effectiveness. Existing synthetic datasets [Murphy *et al.*, 2019; Tsitsulin *et al.*, 2022; Chen *et al.*, 2020], present limitations as they rigidly utilize specific properties as labels, unable to adjust the correlation between properties and labels.

We introduce a method to generate controllable datasets, enabling precise modulation of the correlations between any graph properties and class labels. First, we propose a technique to generate random variables with a given correlation coefficient.

Generate correlated random variables with given coefficients. Suppose each graph property $\mathcal{P}$, and the class label $\mathcal{Y}$ are random variables, the goal is to sample a graph property sequence $\mathbb{P}$ (e.g., $\overline{CC}$) and the class label sequence $\mathbb{Y}$ from the distributions of $\mathcal{P}$ and $\mathcal{Y}$ respectively, which satisfy a given Pearson correlation coefficient r between the property and label, i.e., $r = \text{Pearson}(\mathbb{P}, \mathbb{Y})$.

Theorem 1. *Given a set of property variables $\{\mathcal{P}_i\}_{i=1}^{K}$, each $\mathcal{P}_i$ follows a Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k)$ or Uniform distribution $\mathcal{U}(a_k, b_k)$, and given corresponding Pearson correlation coefficients $\{r_i\}_{i=1}^{K}$ with label variable $\mathcal{Y}$, with the constraint $\sum_{i=1}^{K} r_i^2 \leq 1$, then we have:*

$$\mathcal{Y} = \sigma_{\mathcal{Y}} \left(\sum_{i=1}^{K} n_i r_i + n_0 \sqrt{1 - \sum_{i=1}^{K} r_i^2} \right), \tag{2}$$

where $\sigma_{\mathcal{Y}}$ is any desired standard deviation, and each n_i is mutually independent and follows the same distribution as the corresponding $\mathcal{P}_i$ with the same mean value μ_i but with standard deviation equals to 1. (The proof is based on Cholesky decomposition of a given covariance matrix.)

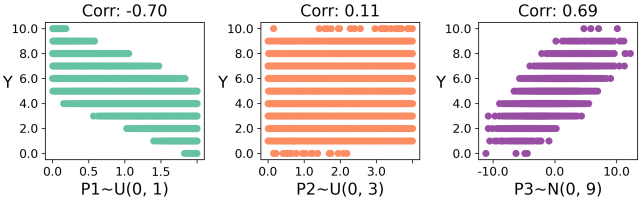


Figure 5: Generated $\mathbb{Y}$ with 11 classes by $\mathcal{P}_2, \mathcal{P}_2, \mathcal{P}_3$ following two uniform and one Gaussian distributions with the correlations $r_1 = -0.7, r_2 = 0.1, r_3 = 0.7$ respectively.

Algorithm 1: Controllable dataset construction

Input: $\{r_k\}_{k=1}^{K}$, number of labels C, $\{\mathcal{P}_k\}_{k=1}^{K} \sim \mathcal{N}(\mu_k, \sigma_k)$ or $\mathcal{U}(a_k, b_k)$;
Output: Dataset $\mathbb{D}$ with size N;
for $k = 1$ **to** K **do**
  Sample $n_k \sim \mathcal{N}(0, \sigma_k)$ or $\mathcal{U}(-\sqrt{3}, \sqrt{3})$;
  $\mathbb{P}_k \leftarrow \mu_k + \sigma_k n_k$ or $\frac{a_k+b_k}{2} + \sqrt{\frac{b_k-a_k}{12}} n_k$;
end
Calculate $\mathcal{Y}$ by the Eq. 2 ;
$\mathbb{Y} \leftarrow \text{ROUND}(\text{NORM}(\mathcal{Y}) * C)$;
$\mathbb{D} \leftarrow \{(g_i, y_i)\}_{i=1}^{N}$,
where each graph g_i has properties $\{\mathbb{P}_k[i]\}_{k=1}^{K}$, and corresponding label $y_i = \mathbb{Y}[i]$;

Inverse graph generation by correlation. By Theorem 1, we can easily generate a desired dataset (includes some graph properties with specific correlations with class labels) following the Algorithm 1. In Alg. 1, it is easy to prove that $\mathbb{P}_k \sim \mathcal{N}(\mu_k, \sigma_k)$, or $\mathbb{P}_k \sim \mathcal{U}(a_k, b_k)$. The **NORM** function is to normalize the $\mathcal{Y}$ into 0 to 1 by min-max normalization, and the **ROUND** function is to convert $\mathcal{Y}$ from decimal to an integer between 0 and $C-1$, to be used as a class label.

The Figure 5 show the precise correlated relationships of generated $\mathbb{Y}$ and each properties $\mathcal{P}_1$,$\mathcal{P}_2$,and $\mathcal{P}_3$ with different correlation coefficients $r_1 = -0.7, r_2 = 0.1, r_3 = 0.7$ respectively. We demonstrate the different distributions of each property. The properties follow three uniform distributions as shown in the left three boxes, and follow three normal distributions as shown in the right three boxes.

Construction of two synthetic datasets. Utilizing Theorem 1, we construct two types of binary classification datasets, specifically **Syn-Degree** and **Syn-CC**. These are generated using Erdos–Renyi (ER) graphs, with a focus on controlling the average graph degree property $\overline{D}$ and average clustering coefficient property $\overline{CC}$, respectively. It's important to note that Theorem 1 is versatile and can be adapted to various graph generation processes beyond ER graphs, by defining specific numerical graph properties. We have created 9 datasets for each type, with each dataset comprising 4096 graphs. In Syn-Degree, both $r_i^{\overline{D}}$ and $r_i^{\overline{CC}}$ range from 0.1 to 0.9. Conversely, in Syn-CC, all $r_i^{\overline{D}}$ are set to 0, while $r_i^{\overline{CC}}$ varies from 0.1 to 0.9. Further details on the construction of these synthetic datasets are available in the supplementary materials, owing to page constraints.

Under our framework, the two dataset types showed notable differences in Figure 6. As correlation rises, the accuracy gap and GIN's accuracy both increase linearly, with the baseline mirroring random guessing. For the Syn-Degree dataset, GIN's accuracy and the baseline both rise linearly, keeping a minimal gap. This suggests two things: a model's prediction accuracy strongly correlates with the coefficient if it captures a graph attribute linked to the label, and GIN effectively captures clustering coefficient and degree information.

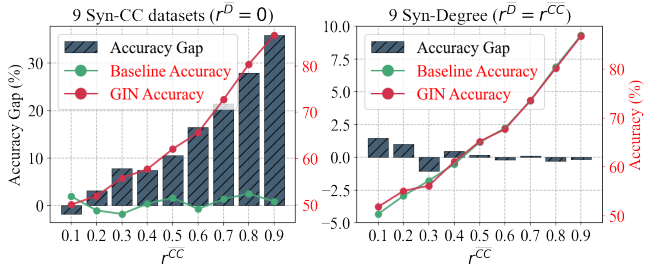


Figure 6: Controllable performance gaps by two types of synthetic datasets.

Regressor	Real-world datasets		Synthetic-CC datasets	
	Pearson	P-Value	Pearson	P-Value
Ridge	0.80 ± 0.09	$\leq 1 \times 10^{-6}$	0.87 ± 0.03	$\leq 1 \times 10^{-6}$
SVR	0.80 ± 0.09	$\leq 1 \times 10^{-6}$	0.89 ± 0.04	$\leq 1 \times 10^{-6}$
RF	0.89 ± 0.03	$\leq 1 \times 10^{-6}$	0.87 ± 0.06	$\leq 1 \times 10^{-6}$

Table 3: Summary of regression results

4.3 Effective Prediction of Effectiveness Through Graph Properties and Statistical Features

Most datasets show strong correlations between graph properties and labels, prompting us to explore predicting dataset effectiveness using these properties, which is computationally cheaper than benchmarking. Drawing from [Xiao *et al.*, 2022], we split each dataset into 10 distinct sets, define 26 features for graph classification, and regress effectiveness using regressors like Random Forest, SVR, and Ridge regression. Using 16 real-world datasets and 9 Syn-CC datasets, we allocate 70% of the splits for training and 30% for testing. Regression performance, verified by the Spearman rank coefficient in Table 3, is based on 10 repeated experiments. Both real-world and Syn-CC datasets show that basic graph properties can effectively predict dataset effectiveness.

5 Conclusions

Our work provides a detailed analysis of graph classification benchmarks essential for the evaluation and enhancement of GNN models. We introduced an empirical protocol to compare the performance of methods like MLPs to GNNs on certain datasets. Our novel Effectiveness metric serves as a pivotal tool for dataset validation in benchmarking. By devising a method to generate synthetic datasets, we can precisely control the correlation between graph properties and task labels, addressing the issue of low effectiveness in some benchmarks. Our efforts play a significant role in the selection of impactful benchmarks, paving the way for the development of robust GNN models and further advancements in graph learning research.

Acknowledgements

This work was supported by National Natural Science Foundation of China (NSFC) under Grant No. 62202410, No. 62311530344, and Shenzhen Science and Technology Program under Grant JCYJ20220530143808019.

References

[Borgwardt and Kriegel, 2005] Karsten M. Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. ICDM '05, page 74–81, USA, 2005. IEEE Computer Society.

[Bouritsas *et al.*, 2022] Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2022.

[Chen *et al.*, 2020] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395, 2020.

[Cui *et al.*, 2022] Hejie Cui, Zijie Lu, Pan Li, and Carl Yang. On positional and structural node features for graph neural networks on non-attributed graphs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3898–3902, 2022.

[Duan *et al.*, 2022] Keyu Duan, Zirui Liu, Peihao Wang, Wenqing Zheng, Kaixiong Zhou, Tianlong Chen, Xia Hu, and Zhangyang Wang. A comprehensive study on large-scale graph training: Benchmarking and rethinking. In *NeurIPS*, 2022.

[Dwivedi *et al.*, 2023] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

[Errica *et al.*, 2020] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. In *ICLR*. OpenReview.net, 2020.

[Feng *et al.*, 2022] Jiarui Feng, Yixin Chen, Fuhai Li, Anindya Sarkar, and Muhan Zhang. How powerful are k-hop message passing graph neural networks. In *NeurIPS*, 2022.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[Hu *et al.*, 2020] 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. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[Kriege and Mutzel, 2012] Nils Kriege and Petra Mutzel. Subgraph matching kernels for attributed graphs. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 291–298, 2012.

[Morris *et al.*, 2020] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

[Murphy *et al.*, 2019] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In *International Conference on Machine Learning*, pages 4663–4673. PMLR, 2019.

[Rieck *et al.*, 2019] Bastian Rieck, Christian Bock, and Karsten Borgwardt. A persistent weisfeiler-lehman procedure for graph classification. In *International Conference on Machine Learning*, pages 5448–5458. PMLR, 2019.

[Shervashidze *et al.*, 2011] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[Tsitsulin *et al.*, 2022] Anton Tsitsulin, Benedek Rozemberczki, John Palowitch, and Bryan Perozzi. Synthetic graph generation to benchmark graph learning. *arXiv preprint arXiv:2204.01376*, 2022.

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

[Wieder *et al.*, 2020] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

[Wu *et al.*, 2021] Lingfei Wu, Yu Chen, Heng Ji, and Bang Liu. Deep learning on graphs for natural language processing. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2651–2653, 2021.

[Wu *et al.*, 2022] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.

[Xiao *et al.*, 2022] Yang Xiao, Jinlan Fu, See-Kiong Ng, and Pengfei Liu. Are all the datasets in benchmark necessary? a pilot study of dataset evaluation for text classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2930–2941, Seattle, United States, July 2022. Association for Computational Linguistics.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Xu *et al.*, 2019] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[Yanardag and Vishwanathan, 2015] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, pages 1365–1374, 2015.

[Zhao and Wang, 2019] Qi Zhao and Yusu Wang. Learning metrics for persistence-based summaries and applications for graph classification. *Advances in Neural Information Processing Systems*, 32, 2019.