
Towards Better Evaluation of GNN Expressiveness with BREC Dataset

Yanbo Wang Muhan Zhang
Institute for Artificial Intelligence, Peking University
yanxwb202@gmail.com, muhan@pku.edu.cn

Abstract

1 Research on the theoretical expressiveness of Graph Neural Networks (GNNs)
2 has developed rapidly, and many methods have been proposed to enhance the
3 expressiveness. However, most methods do not have a uniform expressiveness
4 measure except for a few that strictly follow the k -dimensional Weisfeiler-Lehman
5 (k -WL) test hierarchy. Their theoretical analyses are often limited to distinguishing
6 certain families of non-isomorphic graphs, leading to difficulties in quantitatively
7 comparing their expressiveness. In contrast to theoretical analysis, another way to
8 measure expressiveness is by evaluating model performance on certain datasets
9 containing 1-WL-indistinguishable graphs. Previous datasets specifically designed
10 for this purpose, however, face problems with difficulty (any model surpassing 1-
11 WL has nearly 100% accuracy), granularity (models tend to be either 100% correct
12 or near random guess), and scale (only a few essentially different graphs in each
13 dataset). To address these limitations, we propose a new expressiveness dataset,
14 **BREC**, which includes 400 pairs of non-isomorphic graphs carefully selected from
15 four primary categories (Basic, Regular, Extension, and CFI). These graphs have
16 higher difficulty (up to 4-WL-indistinguishable), finer granularity (able to compare
17 models between 1-WL and 3-WL), and a larger scale (400 pairs). Further, we
18 synthetically test 23 models with higher-than-1-WL expressiveness on our BREC
19 dataset. Our experiment gives the first thorough comparison of the expressiveness
20 of those state-of-the-art beyond-1-WL GNN models. We expect this dataset to
21 serve as a benchmark for testing the expressiveness of future GNNs. Our dataset
22 and evaluation code are released at: <https://github.com/GraphPKU/BREC>.

23 1 Introduction

24 GNNs have been extensively utilized in bioinformatics, recommender systems, social networks, and
25 others, yielding remarkable outcomes [1–6]. Despite impressive empirical achievements, related
26 investigations have revealed that GNNs exhibit limited abilities to distinguish non-isomorphic graphs,
27 such as regular graphs. In a practical scenario, the inability to recognize structure may cause issues,
28 such as confused representation of a benzene ring (a six-cycle that cannot be recognized). Xu et al.
29 [7], Morris et al. [8] established a connection between the expressiveness of message-passing neural
30 networks (MPNNs) and the WL test for graph isomorphism testing, demonstrating that MPNN’s
31 upper bound is 1-WL. Numerous subsequent studies have proposed GNN variants with enhanced
32 expressiveness [9–13].

33 Given the multitude of models employing different approaches, such as feature injection, adherence
34 to the WL hierarchy, equivariance maintenance, and subgraph extraction, a unified framework that
35 can theoretically compare the expressive power among various variants is highly desirable. In
36 this regard, Maron et al. [14] propose the concept of k -order invariant/equivariant graph networks,
37 which unify linear layers while preserving permutation invariance/equivariance. Additionally, Frasca

38 et al. [15] unify recent subgraph GNNs and establish that their expressiveness upper bound is 3-
 39 WL. Zhang et al. [16] construct a comprehensive expressiveness hierarchy for subgraph GNNs,
 40 providing counterexamples for each pairwise distinction. Nonetheless, the magnitude of the gaps
 41 remains unknown. Furthermore, there exist methods that are difficult to categorize within the k -WL
 42 hierarchy. For instance, Papp and Wattenhofer [17] propose four extensions of GNNs, each of which
 43 cannot strictly compare with the other. Similarly, Feng et al. [18] propose a GNN that is partially
 44 stronger than 3-WL yet fails to distinguish many graphs that are distinguishable by 3-WL. In a
 45 different approach, Huang et al. [19] propose evaluating expressiveness by enumerating specific
 46 significant substructures, such as 6-cycles. Zhang et al. [20] introduces graph biconnectivity to test
 47 expressiveness.

48 Without a unified theoretical characterization of expressiveness, employing expressiveness datasets
 49 for testing proves valuable. Notably, three expressiveness datasets, EXP, CSL, and SR25, have been
 50 introduced by Abboud et al. [21], Murphy et al. [22], Balcilar et al. [9] and have found widespread
 51 usage in recent studies. However, these datasets exhibit notable limitations. Firstly, they lack sufficient
 52 difficulty. The EXP and CSL datasets solely consist of examples where 1-WL fails, and most recent
 53 GNN variants have achieved perfect accuracy on these datasets. Secondly, the granularity of these
 54 datasets is too coarse, which means that graphs in these datasets are generated using a single method,
 55 resulting in a uniform level of discrimination difficulty. Consequently, the performance of GNN
 56 variants often falls either at random guessing (completely indistinguishable) or 100% (completely
 57 distinguishable), thereby hindering the provision of a nuanced measure of expressiveness. Lastly,
 58 these datasets suffer from small sizes, typically comprising only a few substantially different graphs,
 59 raising concerns of incomplete measurement.

60 To overcome the limitations of current expressiveness datasets, we propose a new dataset, BREC,
 61 including 400 pairs of non-isomorphic graphs in 4 major categories: Basic graphs, Regular graphs,
 62 Extension graphs, and CFI graphs. Compared to previous ones, BREC has a greater difficulty (up to
 63 4-WL-indistinguishable), finer granularity (able to compare models between 1-WL and 3-WL), and
 64 larger scale (800 non-isomorphic graphs organized as 400 pairs), addressing the shortcomings.

65 Due to the increased size and diversity of the dataset, the traditional classification task may not be
 66 suitable for training-based evaluation methods which rely on generalization ability. Thus, we propose
 67 a novel evaluation procedure based on directly comparing the discrepancies between model outputs to
 68 test pure practical expressiveness. Acknowledging the impact of numerical precision owing to tiny
 69 differences between graph pairs, we propose reliable paired comparisons building upon a statistical
 70 method [23, 24], which offers a precise error bound. Experiments verify that the evaluation procedure
 71 aligns well with known theoretical results.

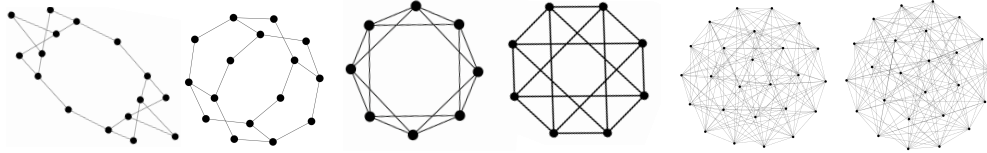
72 Finally, we comprehensively compared 23 representative beyond-1-WL models on BREC. Our
 73 experiments first give a **reliable empirical comparison** of state-of-the-art GNNs’ expressiveness.
 74 The currently most thorough investigation is a good start for gaining deeper insights into various
 75 schemes to enhance GNNs’ expressiveness. On BREC, GNN accuracies range from 41.5% to
 76 70.2%, with I²-GNN [19] performing the best. The 70.2% highest accuracy also implies that the
 77 dataset is **far from saturation**. We expect BREC can serve as a benchmark for testing future GNNs’
 78 expressiveness. We also welcome contributions and suggestions to improve BREC. Our dataset and
 79 evaluation code are included in <https://github.com/GraphPKU/BREC>.

80 2 Limitations of Existing Datasets

81 **Preliminary.** We utilize the notation $\{\}$ to represent sets and $\{\{\}\}$ to represent multisets. The
 82 cardinality of a (multi)set \mathbb{S} is denoted as $|\mathbb{S}|$. The index set is denoted as $[n] = 1, \dots, n$. A graph
 83 is denoted as $\mathcal{G} = (\mathbb{V}(\mathcal{G}), \mathbb{E}(\mathcal{G}))$, where $\mathbb{V}(\mathcal{G})$ represents the set of *nodes* or *vertices* and $\mathbb{E}(\mathcal{G})$
 84 represents the set of *edges*. Without loss of generality, we assume $|\mathbb{V}(\mathcal{G})| = n$ and $\mathbb{V}(\mathcal{G}) = [n]$.

85 The permutation or reindexing of \mathcal{G} is denoted as $\mathcal{G}^\pi = (\mathbb{V}(\mathcal{G}^\pi), \mathbb{E}(\mathcal{G}^\pi))$ with the permutation
 86 function $\pi : [n] \rightarrow [n]$, s.t. $(u, v) \in \mathbb{E}(\mathcal{G}) \iff (\pi(u), \pi(v)) \in \mathbb{E}(\mathcal{G}^\pi)$. Node and edge features are
 87 excluded from the definitions for simplicity. Additional discussions about features can be found in
 88 Appendix B.

89 **Graph Isomorphism (GI) Problem.** Two graphs \mathcal{G} and \mathcal{H} are considered isomorphic (denoted as
 90 $\mathcal{G} \simeq \mathcal{H}$) if $\exists \phi$ (a bijection mapping) $:\mathbb{V}(\mathcal{G}) \rightarrow \mathbb{V}(\mathcal{H})$ s.t. $(u, v) \in \mathbb{E}(\mathcal{G})$ iff. $(\phi(u), \phi(v)) \in \mathbb{E}(\mathcal{H})$.



(a) EXP dataset core pair sample (b) CSL graphs ($m = 10, r = 2$) (c) SR25 dataset sample
Figure 1: Sample graphs in previous datasets

Table 1: Dataset statistics

| Dataset | # Graphs | # Core graphs ^a | # Nodes | Hardness | Metrics |
|-------------|----------|----------------------------|---------------|---------------------------------------|-----------------------------|
| EXP | 1200 | 6 | 33-73 | 1-WL-indistinguishable | 2-way classification |
| CSL | 150 | 10 | 41 | 1-WL-indistinguishable | 10-way classification |
| SR25 | 15 | 15 | 25 | 3-WL-indistinguishable | 15-way classification |
| BREC | 800 | 800 | 10-198 | 1-WL to 4-WL-indistinguishable | Reliable Paired Comparisons |

^a Core graphs represent graphs that actually serve to measure expressiveness.

91 GI is essential in expressiveness. Only if GNN successfully distinguishes two non-isomorphic graphs
 92 can they be assigned different labels. Some researchers [25, 26] indicate the equivalence between GI
 93 and function approximation, underscoring the importance of GI. However, we currently do not have
 94 polynomial-time algorithms for solving the GI problem. A naive solution involves iterating all $n!$
 95 permutations to test whether such a bijection exists.

96 **Weisfeiler-Lehman algorithm (WL).** WL is a well-known isomorphism test relying on color refine-
 97 ment [27]. In each iteration, WL assigns a state (or color) to each node by aggregating information
 98 from its neighboring nodes' states. This process continues until convergence, resulting in a multiset
 99 of node states representing the final graph representation. While WL effectively identifies most
 100 non-isomorphic graphs, it may fail in certain simple graphs, leading to the development of extended
 101 versions. One such extension is k -WL, which treats each k -tuple of nodes as a unit for aggregating
 102 information. Another slightly different method [28] is also referred to as k -WL. To avoid confusion,
 103 we follow Morris et al. [8] to call the former k -WL and the latter k -FWL. Further information can be
 104 found in Appendix C.

105 Given the significance of GI and WL, several expressiveness datasets have been introduced, with the
 106 following three being the most frequently utilized. We selected a pair of graphs from each dataset,
 107 which are illustrated in Figure 1. Detailed statistics for these datasets are presented in Table 1.

108 **EXP Dataset.** This dataset comprises 600 pairs of non-isomorphic graphs where the 1-WL test fails.
 109 Graphs are generated pair-wised, and each graph comprises two disconnected components. The first
 110 component, the "core component," is designed to be non-isomorphic with the other graph's "core
 111 component," each satisfying distinct SAT conditions in the two graphs. The second component,
 112 referred to as the "planar component," is identical in both graphs and introduces noise into the dataset.
 113 However, it is important to note that there are only **three substantially different** core pairs, which
 114 can truly evaluate the expressiveness of the models.

115 Each graph in EXP is labeled 0/1 based on whether its core component satisfies the SAT condition for
 116 a binary classification problem. Although EXP addresses the issue of semantic labeling by introducing
 117 SAT problem and enhances the dataset's size and complexity by including planar components, the
 118 simplicity of core c generation and the insufficient number of different core pairs result in most recent
 119 GNNs achieving nearly 100% accuracy on EXP, making it difficult for detailed comparisons.

120 **CSL Dataset.** This dataset consists of 150 Circulant Skip Links (CSL) graphs, where the 1-WL test
 121 fails. A CSL graph is defined as follows: Let r and m be co-prime natural numbers with $r < m - 1$.
 122 $\mathcal{G}(m, r) = (\mathbb{V}, \mathbb{E})$ is an undirected 4-regular graph with $\mathbb{V} = [m]$, where the edges form a cycle and
 123 include skip links. Specifically, for the cycle, $(j, j + 1) \in \mathbb{E}$ for $j \in [m - 1]$, and $(m, 1) \in \mathbb{E}$. For the
 124 skip links, the sequence is recursively defined as $s_1 = 1, s_{i+1} = (s_i + r) \bmod m + 1$, and $(s_i, s_{i+1}) \in$
 125 \mathbb{E} for any $i \in \mathbb{N}$. In CSL, we consider CSL graphs with $m = 41$ and $r = 2, 3, 4, 5, 6, 9, 11, 12, 13, 16$,
 126 resulting in 10 distinct CSL graphs. For each distinct CSL graph, we generate 14 corresponding
 127 graphs by randomly reindexing the nodes. As a result, the dataset contains a total of 150 graphs.

128 In CSL, each of the 10 distinct CSL graphs is treated as a separate class, and the task is to train a
 129 10-way classification model. While the dataset allows for the generation of 4-regular graphs with
 130 any number of nodes, the final dataset contains only **ten essentially different** regular graphs with the

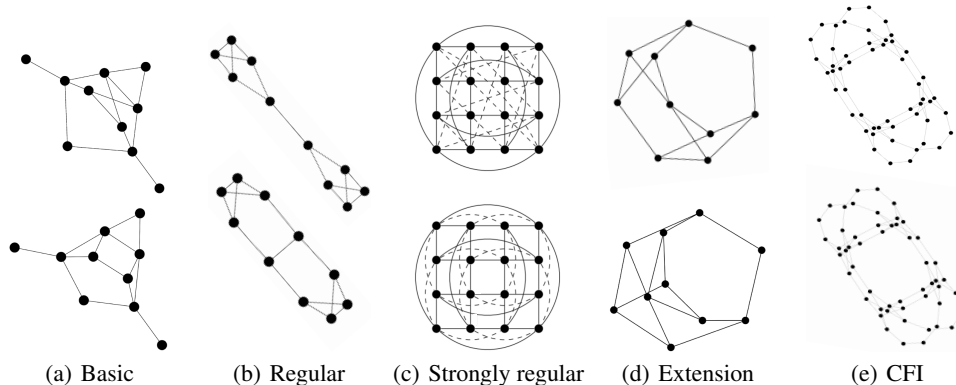


Figure 2: BREC dataset samples

131 **same number of nodes and degree.** Due to the nature of regular graphs and their fixed structure,
 132 many recent expressive GNN models perform well on this dataset, achieving close to 100% accuracy.

133 **SR25 Dataset.** It consists of 15 strongly regular graphs (SR) where the 3-WL test fails. Each graph is
 134 an SR with 25 nodes and a degree of 12. In these graphs, connected nodes have 5 common neighbors,
 135 while non-connected nodes have 6. In practice, SR25 is transformed into a 15-way classification
 136 problem for mapping each graph into a different class where the training and test graphs overlap.

137 Indeed, 3-WL serves as an upper bound for most recent expressive GNNs. Thus most methods
 138 only obtain 6.67% (1/15) accuracy. While some models partially surpassing 3-WL easily achieve
 139 completely distinguishable (100%) performance [18], since each graph is an SR with the same
 140 parameters. This binary outcome can hardly provide a fine-grained expressiveness measure.

141 **Summary.** These three datasets have limitations regarding difficulty, granularity, and scale. In terms
 142 of difficulty, these datasets are all bounded by 3-WL, failing to evaluate models (partly) beyond
 143 3-WL [18, 19]. In terms of granularity, the graphs are generated in one way, and the parameters of
 144 the graphs are repetitive, which easily leads to a 0/1 step function of model performance and cannot
 145 measure subtle differences between models. In terms of scale, the number of substantially different
 146 graphs in the datasets is small, and the test results may be incomplete to reflect expressiveness
 147 measurement.

148 3 BREC: A New Dataset for Expressiveness

149 We propose a new expressiveness dataset, BREC, to address the limitations regarding difficulty,
 150 granularity, and scale. It consists of four major categories of graphs: Basic, Regular, Extension,
 151 and CFI. Basic graphs include relatively simple 1-WL-indistinguishable graphs. Regular graphs
 152 include four types of subcategorized regular graphs. Extension graphs include special graphs that
 153 arise when comparing four kinds of GNN extensions [17]. CFI graphs include graphs generated by
 154 CFI methods¹ [28] with high difficulty. Some samples are shown in Fig 2.

155 3.1 Dataset Composition

156 BREC includes 800 non-isomorphic graphs arranged in a pairwise manner to construct 400 pairs, with
 157 detailed composition as follows: (For a more detailed generation process, please refer to AppendixK)

158 **Basic Graphs.** Basic graphs consist of 60 pairs of 10-node graphs. These graphs are collected
 159 from an exhaustive search and intentionally designed to be non-regular. Although they are 1-
 160 WL-indistinguishable, most can be distinguished by expressive GNN variants. Basic graphs can
 161 also be regarded as an augmentation of the EXP dataset, as they both employ non-regular 1-WL-
 162 indistinguishable graphs. Nevertheless, Basic graphs offer a greater abundance of instances and more
 163 intricate graph patterns. The relatively small size also facilitates visualization and analysis.

164 **Regular Graphs.** Regular graphs consist of 140 pairs of regular graphs, including 50 pairs of simple
 165 regular graphs, 50 pairs of strongly regular graphs, 20 pairs of 4-vertex condition graphs, and 20 pairs

¹CFI is short for Cai-Furer-Immerman algorithm, which can generate counterexample graphs for any k-WL.

166 of distance regular graphs. Each pair of graphs shares identical parameters. A regular graph refers
167 to a graph where all nodes possess the same degree. Regular graphs are 1-WL-indistinguishable,
168 and some studies delve into the analysis of GNN expressiveness from this perspective [29, 13]. We
169 denote regular graphs without any special properties as simple regular graphs. When exploring
170 more intricate regular graphs, the concept of strongly regular graphs (where 3-WL fails) is often
171 introduced. Strongly regular graphs further require that the number of neighboring nodes shared by
172 any two nodes depends solely on their connectivity. Notable examples of strongly regular graphs
173 include the 4×4 -Rook’s graph and the Shrikhande graph (Fig 2(c)). Additionally, the 4×4 -Rook’s
174 graph satisfies the 4-vertex condition property, which signifies that the number of connected edges
175 between the common neighbors of any two nodes is solely determined by their connectivity [30]. It
176 is worth mentioning that the diameter of a connected strongly regular graph is always 2 [31]. A more
177 challenging type of graph known as the distance regular graphs [32] is proposed aiming for extending
178 the diameter. Please refer to Appendix A for a more comprehensive exploration of their relationship.

179 Regular graphs can also as an enriching addition to the CSL and SR25 datasets. By expanding upon
180 the existing subdivisions of regular graphs, this section widens the range of difficulty and raises the
181 upper bound of complexity. Moreover, unlike the previous datasets, regular graphs are not limited to
182 sharing identical parameters for all graphs within each category, greatly enhancing diversity.

183 **Extension Graphs.** Extension graphs include 100 pairs of graphs inspired by Papp and Wattenhofer
184 [17]. They proposed 4 types of theoretical GNN extensions: k -WL hierarchy-based, substructure-
185 counting-based, k -hop-subgraph-based, and marking-based methods. The authors reveal that most of
186 them are not strictly comparable. Leveraging the insights from theoretical analysis and some empiri-
187 cally derived findings, we generated 100 pairs of 1-WL-indistinguishable and 3-WL-distinguishable
188 graphs to improve the granularity. Noting that it was not considered in any of the previous datasets.

189 **CFI Graphs.** CFI graphs consist of 100 pairs of graphs inspired by Cai et al. [28]. They developed a
190 method to generate graphs distinguishable by k -WL but not by $(k - 1)$ -WL for any k . We utilized
191 this method to create 100 pairs of graphs spanning up to 4-WL-indistinguishable, even surpassing the
192 current research’s upper bounds. Specifically, 60 pairs are solely distinguishable by 3-WL, 20 are
193 solely distinguishable by 4-WL, and 20 are even 4-WL-indistinguishable. Similar to the previously
194 mentioned parts, CFI graphs were not considered in the previous datasets. As the most challenging
195 part, it pushes the upper limit of difficulty even higher. Furthermore, the graph sizes in this section
196 are larger than other parts (up to 198 nodes). This aspect intensifies the challenge of the dataset,
197 demanding a model’s ability to process graphs with heterogeneous sizes effectively.

198 3.2 Advantages

199 **Difficulty.** By utilizing the CFI method, we specifically provide graphs being 4-WL-indistinguishable.
200 Additionally, we include 4-vertex condition graphs and distance regular graphs, which are variants of
201 strongly regular graphs (3-WL-indistinguishable) but pose greater challenges in terms of complexity.

202 **Granularity.** The different classes of graphs in BREC exhibit varying difficulty levels, each con-
203 tributing to the dataset in distinct ways. Basic graphs contain fundamental 1-WL-indistinguishable
204 graphs, similar to the EXP dataset, as a starting point for comparison. Regular graphs extend the CSL
205 and SR25 datasets. The major components of regular graphs are simple regular graphs and strongly
206 regular graphs, where 1-WL and 3-WL fail, respectively. Including 4-vertex condition graphs and
207 distance regular graphs further elevates the complexity. Extension graphs bridge the gap between
208 1-WL and 3-WL, offering a finer-grained comparison for evaluating models beyond 1-WL. CFI
209 graphs span the spectrum of difficulty from 1-WL to 4-WL-indistinguishable. By comprehensive
210 graph composition, BREC explores the boundaries of graph pattern distinguishability.

211 **Scale.** While previous datasets relied on only tens of different graphs to generate the dataset, BREC
212 utilizes a collection of 800 different graphs. This significant increase in the number of graphs greatly
213 enhances the diversity. The larger graph set in BREC also contributes to a more varied distribution of
214 graph statistics. In contrast, previous datasets such as CSL and SR25 only have the same number of
215 nodes and degrees across all graphs. For detailed statistics of BREC, please refer to Appendix D.

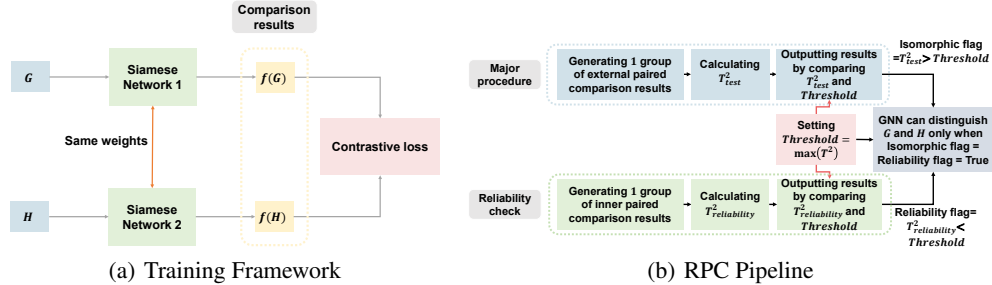


Figure 3: Evaluation Method

216 4 RPC: A New Evaluation Method

217 This section introduces a novel training framework and evaluation method for BREC. Unlike previous
 218 datasets, BREC departs from the conventional classification setting, where each graph is assigned
 219 a label, a classification model is trained, and the accuracy on test graphs serves as the measure
 220 of expressiveness. The labeling schemes used in previous datasets like semantic labels based on
 221 SAT conditions in EXP, or distinct labels for essentially different graphs in CSL and SR25, do not
 222 apply to BREC. There are two primary reasons. First, BREC aims to enrich the diversity of graphs,
 223 which precludes using a semantic label tied to SAT conditions, as it would significantly limit the
 224 range of possible graphs. Second, assigning a distinct label to each graph in BREC would result in
 225 an 800-class classification problem, where performance could be influenced by factors other than
 226 expressiveness. Our core idea is to measure models' "separating power" directly. Thus BREC is
 227 organized in pairs, where each pair is individually tested to determine whether a GNN can distinguish
 228 them. By adopting a pairwise evaluation method, BREC provides a more focused measure of models'
 229 expressiveness, aligning to assess distinguishing ability.

230 Nevertheless, how can we say a pair of graphs is successfully distinguished? Previous researchers
 231 tend to set a small threshold (like 1E-4) manually. If the embedding distance between them is
 232 larger than the threshold, the GNN is considered can distinguish them. However, this method
 233 lacks **reliability** due to numerical precision, especially when graphs vary in size. In order to yield
 234 dependable outcomes, we propose an evaluation method measuring both **external difference** and
 235 **internal fluctuations**. Furthermore, we introduce a training framework for pairwise data, employing
 236 the siamese network design [33] and contrastive loss [34, 35]. The pipeline is depicted in Fig 3(a).

237 4.1 Training Framework

238 We adhere to the siamese network design [33] to train a model to distinguish each pair of graphs. The
 239 central component consists of two identical models that maintain identical parameters. When a pair
 240 of graphs is inputted, it produces a corresponding pair of embeddings. Subsequently, the difference
 241 between them is assessed using cosine similarity. The loss function is formulated as follows:

$$L(f, \mathcal{G}, \mathcal{H}) = \text{Max}(0, \frac{f(\mathcal{G}) \cdot f(\mathcal{H})}{\|f(\mathcal{G})\| \|f(\mathcal{H})\|} - \gamma), \quad (1)$$

242 where the GNN model $f : \{\mathcal{G}\} \rightarrow \mathbb{R}^d$, \mathcal{G} and \mathcal{H} are two non-isomorphic graphs, and γ is a margin
 243 hyperparameter (set to 0 in our experiments). The loss function aims to promote the cosine similarity
 244 value lower than γ , thereby encouraging a greater separation between the two graph embeddings.

245 The training process **yields several benefits** for the models. Firstly, it enables the GNN to achieve
 246 its theoretical expressiveness. The theoretical analysis of GNN expressiveness focuses primarily
 247 on the network's structure without imposing any constraints on its parameters, which means we
 248 are exploring the expressiveness of **a group of functions**. If a model with particular parameters
 249 can distinguish a pair of graphs, the model's design and structure possess sufficient expressiveness.
 250 However, it is impractical to iterate all possible parameter combinations to test the real upper bound.
 251 Hence, training can **realize searching** in the function space, enabling models to achieve better
 252 practical expressiveness. Furthermore, training aids components to **possess specific properties**, such
 253 as injectivity and universal approximation, which are vital for attaining theoretical expressiveness.
 254 These properties require specific parameter configurations, and randomly initialized parameters may
 255 not satisfy these requirements. Moreover, through training, model-distinguishable pairs are **more**
 256 **easily discriminated** from model-indistinguishable pairs, which helps reduce the false negative rate

257 caused by numerical precision. The difference between their embeddings is further magnified in
 258 the pairwise contrastive training process if the model distinguishes them. However, the difference
 259 remains unaffected mainly and is only influenced by numerical errors for model-indistinguishable
 260 pairs. The training framework is illustrated in Fig 3(a).

261 4.2 Evaluation Method

262 Recall that our approach involves comparing the outputs on a pair of non-isomorphic graphs. If
 263 there exists a notable disparity between them, we consider the GNN to be able to distinguish them.
 264 However, determining an appropriate threshold poses a challenge. A large threshold may yield
 265 false negatives where the model is expressive enough, but the observed difference falls short of
 266 the threshold. Conversely, a small threshold may result in false positives, where the model fails to
 267 distinguish the graphs. However, the fluctuating or numerical errors cause the difference to exceed
 268 the small threshold.

269 To address the issue of fluctuating errors, we draw inspiration from Paired Comparisons [23]. It
 270 involves comparing two groups of results instead of a single pair. The influence of random errors is
 271 mitigated by repeatedly generating results and comparing the two groups of results. Building upon it,
 272 we introduce a method called **Reliable Paired Comparison (RPC)** to verify whether a GNN genuinely
 273 produces distinct outputs for a pair of graphs. The pipeline is depicted in Fig 3(b).

274 RPC consists of two main components: Major procedure and Reliability check. The Major procedure
 275 is conducted on a pair of non-isomorphic graphs to measure their dissimilarity. In comparison,
 276 the Reliability check is conducted on graph automorphisms to capture internal fluctuations with
 277 numerical precision.

278 **Major procedure.** For two non-isomorphic graphs \mathcal{G} and \mathcal{H} , we create q copies of each by randomly
 279 reindexing (operate permutation on node indexes, thus generating an isomorphic graph but with
 280 different node orders) them. It results in two groups of graphs, where each copy is represented as:

$$\mathcal{G}_i, \mathcal{H}_i, i \in [q]. \quad (2)$$

281 Supposing the GNN $f : \{\mathcal{G}\} \rightarrow \mathbb{R}^d$, we first calculate q differences utilizing Paired Comparisons.

$$\mathbf{d}_i = f(\mathcal{G}_i) - f(\mathcal{H}_i), i \in [q]. \quad (3)$$

282 **Assumption 4.1.** \mathbf{d}_i are independent $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ random vectors.

283 The above assumption is based on a more basic assumption that $f(\mathcal{G}_i)$, $f(\mathcal{H}_i)$ follow Gaussian
 284 distributions, which presumes that random reindexing only introduces Gaussian noise to the result.

285 The mean difference between two graph embeddings $\boldsymbol{\mu} = \mathbf{0}$ implies the GNN cannot distinguish them.
 286 Therefore, we can obtain the distinguishing result by conducting an α -level Hotelling's T-square test,
 287 comparing the hypotheses $H_0 : \boldsymbol{\mu} = \mathbf{0}$ against $H_1 : \boldsymbol{\mu} \neq \mathbf{0}$. We calculate the T^2 -statistic for $\boldsymbol{\mu}$ as:

$$T^2 = q(\bar{\mathbf{d}} - \boldsymbol{\mu})^T \mathbf{S}^{-1}(\bar{\mathbf{d}} - \boldsymbol{\mu}), \quad (4)$$

288 where

$$\bar{\mathbf{d}} = \frac{1}{q} \sum_{i=1}^q \mathbf{d}_i, \mathbf{S} = \frac{1}{q-1} \sum_{i=1}^q (\mathbf{d}_i - \bar{\mathbf{d}})(\mathbf{d}_i - \bar{\mathbf{d}})^T. \quad (5)$$

289 Hotelling's T-square test proves that T^2 is distributed as an $\frac{(q-1)d}{q-d} F_{d, q-d}$ random variable, whatever
 290 the true $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ [36]. The theorem establishes a connection between the unknown parameter $\boldsymbol{\mu}$ and a
 291 definite probability distribution $F_{d, q-d}$, allowing us to confirm the confidence interval of $\boldsymbol{\mu}$ by testing
 292 the distribution fit. In order to test the hypothesis $H_0 : \boldsymbol{\mu} = \mathbf{0}$, we substitute $\boldsymbol{\mu} = \mathbf{0}$ into Equation (4)
 293 to obtain $T_{\text{test}}^2 = q\bar{\mathbf{d}}^T \mathbf{S}^{-1}\bar{\mathbf{d}}$. Then, for a specific α , an α -level test of $H_0 : \boldsymbol{\mu} = \mathbf{0}$ versus $H_1 : \boldsymbol{\mu} \neq \mathbf{0}$
 294 for a population following $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution accepts H_0 (the GNN cannot distinguish the pair) if:
 295

$$T_{\text{test}}^2 = q\bar{\mathbf{d}}^T \mathbf{S}^{-1}\bar{\mathbf{d}} < \frac{(q-1)d}{(q-d)} F_{d, q-d}(\alpha), \quad (6)$$

296 where $F_{d,q-d}(\alpha)$ is the upper (100α) th percentile of the F -distribution $F_{d,q-d}$ [37] with d and $q-d$
 297 degrees of freedom. Similarly, we reject H_0 (the GNN can distinguish the pair) if

$$T_{\text{test}}^2 = q\bar{\mathbf{d}}^T \mathbf{S}^{-1}\bar{\mathbf{d}} > \frac{(q-1)d}{(q-d)} F_{d,q-d}(\alpha). \quad (7)$$

298 **Reliability check.** Although the above test is theoretically valid for evaluating the expressiveness
 299 of GNNs, in practice, it is susceptible to computational precision limitations. These limitations can
 300 manifest in various scenarios, such as comparing numbers close to zero or inverting a matrix close to
 301 zero, making it difficult to rely on the test constantly. We incorporate the Reliability check to monitor
 302 abnormal results to address this concern. This step effectively bridges the external difference between
 303 two graphs and the internal fluctuations within a single graph.

304 WLOG, we replace \mathcal{H} by reindexing of \mathcal{G} , i.e., \mathcal{G}^π . Thus, we can obtain the internal fluctuations
 305 within \mathcal{G} by comparing it with \mathcal{G}^π , and the external difference between \mathcal{G} and \mathcal{H} by comparing \mathcal{G} and
 306 \mathcal{H} . We utilize the same step as Major procedure on \mathcal{G} and \mathcal{G}^π , calculating the T^2 -statistics as follows:

$$T_{\text{reliability}}^2 = q\bar{\mathbf{d}}^T \mathbf{S}^{-1}\bar{\mathbf{d}}, \quad (8)$$

$$\text{where } \bar{\mathbf{d}} = \frac{1}{q} \sum_{i=1}^q \mathbf{d}_i, \mathbf{d}_i = f(\mathcal{G}_i) - f(\mathcal{G}_i^\pi), i \in [q], \mathbf{S} = \frac{1}{q-1} \sum_{i=1}^q (\mathbf{d}_i - \bar{\mathbf{d}})(\mathbf{d}_i - \bar{\mathbf{d}})^T. \quad (9)$$

307 Recalling that \mathcal{G} and \mathcal{G}^π are isomorphic, the GNN should not distinguish between them, implying
 308 that $\boldsymbol{\mu} = \mathbf{0}$. Therefore, the test result is considered reliable only if $T_{\text{reliability}}^2 < \frac{(q-1)d}{(q-d)} F_{d,q-d}(\alpha)$.
 309 Combining the reliability and distinguishability results, we get the complete RPC (Fig 3) as follows:

310 For each pair of graphs \mathcal{G} and \mathcal{H} , we first calculate the threshold value, denoted as Threshold =
 311 $\frac{(q-1)d}{(q-d)} F_{d,q-d}(\alpha)$. Next, we conduct the Major procedure on \mathcal{G} and \mathcal{H} for distinguishability and
 312 perform the Reliability check on \mathcal{G} and \mathcal{G}^π for Reliability. Only when the T^2 -statistic from the Major
 313 procedure, denoted as T_{test}^2 , and the T^2 -statistic from the Reliability check, denoted as $T_{\text{reliability}}^2$,
 314 satisfying $T_{\text{reliability}}^2 < \text{Threshold} < T_{\text{test}}^2$, do we conclude that the GNN can distinguishing \mathcal{G} and \mathcal{H} .

315 We further propose **Reliable Adaptive Pairwise Comparison (RAPC)**, aiming to adaptively adjust the
 316 threshold and provide an upper bound for false positive rates. In practice, we use **RPC** due to its less
 317 computational time and satisfactory performance. For more about RAPC, please refer to Appendix E.

318 5 Experiment

319 In this section, we evaluate the expressiveness of 23 representative models using our BREC dataset.

320 **Model selection.** We evaluate six categories of methods: non-GNN methods, subgraph-based GNNs,
 321 k -WL-hierarchy-based GNNs, substructure-based GNNs, transformer-based GNNs, and random
 322 GNNs. Our primary focus will be on the first three categories. We implement four types of non-
 323 GNN baselines based on Papp and Wattenhofer [17], Ying et al. [38], including WL test (3-WL and
 324 SPD-WL), counting substructures (S_3 and S_4), neighborhood up to a certain radius (N_1 and N_2),
 325 and marking (M_1). We implemented them by adding additional features during the WL test update
 326 or using heterogeneous message passing. It is important to note that they are more theoretically
 327 significant than practical since they may require exhaustive enumeration or exact isomorphism
 328 encoding of various substructures. We additionally included 16 state-of-the-art GNNs, including
 329 NGNN [13], DE+NGNN [29], DS/DSS-GNN [10], SUN [15], SSWL_P [16], GNN-AK [39], KP-
 330 GNN [18], I²-GNN [19], PPGN [40], δ -k-LGNN [41], KC-SetGNN [42], GSN [43], DropGNN [44],
 331 OSAN [45], and Graphormer [38].

332 Table 2 presents the primary results. N_2 achieves the highest accuracy among non-GNNs, and I²-GNN
 333 achieves the highest among GNNs. We detail each method’s accuracy on different graphs, showing
 334 that it matches theoretical results well. Detailed experiment settings are included in Appendix J.

335 **Non-GNN baselines.** 3-WL successfully distinguishes all Basic graphs, Extension graphs, simple
 336 regular graphs and 60 CFI graphs as expected. S_3 , S_4 , N_1 , and N_2 demonstrate excellent performance
 337 on small-radius graphs such as Basic, Regular, and Extension graphs. However, due to their limited
 338 receptive fields, they struggle to distinguish large-radius graphs like CFI graphs. Noting that the

Table 2: Pair distinguishing accuracies on BREC

| Model | Basic Graphs (60) | | Regular Graphs (140) | | Extension Graphs (100) | | CFI Graphs (100) | | Total (400) | |
|---------------------|-------------------|----------|----------------------|----------|------------------------|----------|------------------|----------|-------------|----------|
| | Number | Accuracy | Number | Accuracy | Number | Accuracy | Number | Accuracy | Number | Accuracy |
| 3-WL | 60 | 100% | 50 | 35.7% | 100 | 100% | 60 | 60.0% | 270 | 67.5% |
| SPD-WL | 16 | 26.7% | 14 | 11.7% | 41 | 41% | 12 | 12% | 83 | 20.8% |
| S_3 | 52 | 86.7% | 48 | 34.3% | 5 | 5% | 0 | 0% | 105 | 26.2% |
| S_4 | 60 | 100% | 99 | 70.7% | 84 | 84% | 0 | 0% | 243 | 60.8% |
| N_1 | 60 | 100% | 99 | 85% | 93 | 93% | 0 | 0% | 252 | 63% |
| N_2 | 60 | 100% | 138 | 98.6% | 100 | 100% | 0 | 0% | 298 | 74.5% |
| M_1 | 60 | 100% | 50 | 35.7% | 100 | 100% | 41 | 41% | 251 | 62.8% |
| NGNN | 59 | 98.3% | 48 | 34.3% | 59 | 59% | 0 | 0% | 166 | 41.5% |
| DE+NGNN | 60 | 100% | 50 | 35.7% | 100 | 100% | 21 | 21% | 231 | 57.8% |
| DS-GNN | 58 | 96.7% | 48 | 34.3% | 100 | 100% | 16 | 16% | 222 | 55.5% |
| DSS-GNN | 58 | 96.7% | 48 | 34.3% | 100 | 100% | 15 | 15% | 221 | 55.2% |
| SUN | 60 | 100% | 50 | 35.7% | 100 | 100% | 13 | 13% | 223 | 55.8% |
| SSWL_P | 60 | 100% | 50 | 35.7% | 100 | 100% | 38 | 38% | 248 | 62% |
| GNN-AK | 60 | 100% | 50 | 35.7% | 97 | 97% | 15 | 15% | 222 | 55.5% |
| KP-GNN | 60 | 100% | 106 | 75.7% | 98 | 98% | 11 | 11% | 275 | 68.8% |
| I ² -GNN | 60 | 100% | 100 | 71.4% | 100 | 100% | 21 | 21% | 281 | 70.2% |
| PPGN | 60 | 100% | 50 | 35.7% | 100 | 100% | 23 | 23% | 233 | 58.2% |
| δ -k-LGNN | 60 | 100% | 50 | 35.7% | 100 | 100% | 6 | 6% | 216 | 54% |
| KC-SetGNN | 60 | 100% | 50 | 35.7% | 100 | 100% | 1 | 1% | 211 | 52.8% |
| GSN | 60 | 100% | 99 | 70.7% | 95 | 95% | 0 | 0% | 254 | 63.5% |
| DropGNN | 52 | 86.7% | 41 | 29.3% | 82 | 82% | 2 | 2% | 177 | 44.2% |
| OSAN | 56 | 93.3% | 8 | 5.7% | 79 | 79% | 5 | 5% | 148 | 37% |
| Graphormer | 16 | 26.7% | 12 | 10% | 41 | 41% | 10 | 10% | 79 | 19.8% |

expressiveness of S_3 and S_4 is bounded by N_1 and N_2 , respectively, as analyzed by Papp and Wattenhofer [17]. Conversely, M_1 is implemented by heterogeneous message passing, which makes it unaffected by large graph diameters, thus maintaining its performance across different graphs. SPD-WL is another 1-WL extension operated on a complete graph with shortest path distances as edge features. It may degrade to 1-WL on low-radius graphs, causing its relatively poor performance.

Subgraph-based GNNs. Regarding subgraph-based models, they can generally distinguish almost all Basic graphs, simple regular graphs and Extension graphs. However, an exception lies with NGNN, which performs poorly in Extension graphs due to its simplicial node selection policy and lack of node labeling. Two other exceptions are KP-GNN and I²-GNN, both exhibiting exceptional performance in Regular graphs. KP-GNN can differentiate a substantial number of strongly regular graphs and 4-vertex condition graphs, surpassing the 3-WL partially. And I²-GNN surpasses the limitations of 3-WL partially through its enhanced cycle-counting power. An influential aspect that impacts the performance is the subgraph radius. Approaches incorporating appropriate encoding functions are expected to yield superior performance as the subgraph radius increases. However, in practice, enlarging the radius may result in the smoothness of information, wherein the receptive field expands, encompassing some irrelevant or noisy information. Hence, we treat the subgraph radius as a hyperparameter, fine-tuning it for each model, and present the best results in Table 2. Please refer to Appendix F for further details regarding the radius selection.

When comparing various subgraph GNNs, KP-GNN can discriminate part of the strongly regular graphs by peripheral subgraphs. Additionally, distance encoding in DE+NGNN and I²-GNN enables better discrimination among different hops within a given subgraph radius, enhancing the discriminative ability, particularly in larger subgraph radii. As for DS-GNN, DSS-GNN, GNN-AK, SUN and SSWL_P, they employ similar aggregation schemes with slight variations in their operations. These models exhibit comparable performance, with SSWL_P outperforming others, which aligns with expectations since SSWL_P is more expressive but with the least components.

k -WL hierarchy-based GNNs. For the k -WL-hierarchy-based models, we adopt two implemented approaches: high-order simulation and local-WL simulation. PPGN serves as the representative work for the former, while δ -k-LGNN and KCSet-GNN embody the latter. PPGN aligns its performance with 3-WL across all graphs except for CFI graphs. For CFI graphs with large radii, more WL iterations (layers of GNNs) are required. However, employing many layers may lead to over-smoothing, resulting in a gap between theoretical expectations and actual performance. Nonetheless, PPGN still surpasses most GNNs in CFI graphs due to global k -WL’s global receptive field. For δ -k-LGNN, we set $k = 2$, while for KCSet-GNN, we set $k = 3, c = 2$ to simulate local 3-WL, adhering to the original configuration. By comparing the output results with relatively small diameters, we observed that local WL matches the performance of general k -WL. However, local WL exhibits lower performance for CFI graphs with larger radii due to insufficient receptive fields.

375 **Substructure-based GNNs** For substructure-based GNNs, we select GSN, which incorporate sub-
376 structure isomorphism counting as features. The best result obtained for GSN-e is reported when
377 setting $k = 4$. For further exploration of policy and size, please refer to Appendix H.

378 **Random GNNs** Random GNNs are unsuitable for GI problems since even identical graphs can yield
379 different outcomes due to inherent randomness. However, the RPC can quantify fluctuations in the
380 randomization process, thereby enabling the testing of random GNNs. We test DropGNN and OSAN.
381 For more information regarding the crucial factor of random samples, please refer to Appendix I.

382 **Transformer-based GNNs** For transformer-based GNNs, we select graphormer, which is anticipated
383 to possess a level of expressiveness comparable to SPD-WL. The experimental results verify that.

384 6 Conclusion and Future Work

385 This paper proposes a new dataset, BREC, for GNN expressiveness comparison. BREC addresses
386 the limitations of previous datasets, including difficulty, granularity, and scale, by incorporating
387 400 pairs of diverse graphs in four categories. A new evaluation method is proposed for principled
388 expressiveness evaluation. Finally, a thorough comparison of 23 baselines on BREC is conducted.

389 Apart from the expressiveness comparison based on GI, there are various other metrics for GNN
390 expressiveness evaluation, such as substructure counting, diameter counting, and biconnectivity
391 checking. However, it’s worth noting that these tests are often conducted on datasets not specifically
392 designed for expressiveness [19, 39, 46], which can lead to biased results caused by spurious
393 correlations. In other words, certain methods may struggle to identify a particular substructure,
394 but they can capture another property that correlates with substructures, resulting in false high
395 performance. This problem can be alleviated in BREC because of the difficulty. We reveal the
396 data generation process of BREC in Appendix K, hoping that researchers can utilize them in more
397 tasks. We also hope the test of practical expressiveness will aid researchers in exploring its effects on
398 performance in real datasets and other domains.

399 References

- 400 [1] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel,
401 Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning
402 molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.
- 403 [2] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-
404 based approach to human disease. *Nature reviews genetics*, 12(1):56–68, 2011.
- 405 [3] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural
406 networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.
- 407 [4] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi
408 Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender sys-
409 tems. In *Proceedings of the 27th ACM international conference on information and knowledge*
410 *management*, pages 417–426, 2018.
- 411 [5] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion.
412 *arXiv preprint arXiv:1706.02263*, 2017.
- 413 [6] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng
414 Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and
415 applications. *AI open*, 1:57–81, 2020.
- 416 [7] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural
417 networks? In *7th International Conference on Learning Representations, ICLR 2019, New*
418 *Orleans, LA, USA, May 6-9, 2019, Conference Track Proceedings*. OpenReview.net, 2019.
419 URL <https://openreview.net/forum?id=ryGs6iA5Km>.
- 420 [8] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen,
421 Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural
422 networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages
423 4602–4609, 2019.

- 424 [9] Muhammet Balcilar, Pierre H eroux, Benoit Gauzere, Pascal Vasseur, S ebastien Adam, and
425 Paul Honeine. Breaking the limits of message passing graph neural networks. In International
426 Conference on Machine Learning, pages 599–608. PMLR, 2021.
- 427 [10] Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai,
428 Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant subgraph aggrega-
429 tion networks. In The Tenth International Conference on Learning Representations, ICLR
430 2022, Virtual Event, April 25-29, 2022. OpenReview.net, 2022. URL [https://openreview.
431 net/forum?id=dFbKQaRk15w](https://openreview.net/forum?id=dFbKQaRk15w).
- 432 [11] Leonardo Cotta, Christopher Morris, and Bruno Ribeiro. Reconstruction for powerful graph
433 representations. Advances in Neural Information Processing Systems, 34:1713–1726, 2021.
- 434 [12] Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph
435 neural networks. In Proceedings of the AAAI conference on artificial intelligence, volume 35,
436 pages 10737–10745, 2021.
- 437 [13] Muhan Zhang and Pan Li. Nested graph neural networks. Advances in Neural Information
438 Processing Systems, 34:15734–15747, 2021.
- 439 [14] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant
440 graph networks. In 7th International Conference on Learning Representations, ICLR 2019,
441 New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL [https://openreview.
442 net/forum?id=Syx72jC9tm](https://openreview.net/forum?id=Syx72jC9tm).
- 443 [15] Fabrizio Frasca, Beatrice Bevilacqua, Michael Bronstein, and Haggai Maron. Understanding
444 and extending subgraph gnns by rethinking their symmetries. Advances in Neural Information
445 Processing Systems, 35:31376–31390, 2022.
- 446 [16] Bohang Zhang, Guhao Feng, Yiheng Du, Di He, and Liwei Wang. A complete expressive-
447 ness hierarchy for subgraph GNNs via subgraph weisfeiler-lehman tests. In Andreas Krause,
448 Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett,
449 editors, Proceedings of the 40th International Conference on Machine Learning, volume 202
450 of Proceedings of Machine Learning Research, pages 41019–41077. PMLR, 23–29 Jul 2023.
451 URL <https://proceedings.mlr.press/v202/zhang23k.html>.
- 452 [17] P al Andr as Papp and Roger Wattenhofer. A theoretical comparison of graph neural network
453 extensions. In International Conference on Machine Learning, pages 17323–17345. PMLR,
454 2022.
- 455 [18] Jiarui Feng, Yixin Chen, Fuhai Li, Anindya Sarkar, and Muhan Zhang. How powerful are k-hop
456 message passing graph neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave,
457 and Kyunghyun Cho, editors, Advances in Neural Information Processing Systems, 2022. URL
458 <https://openreview.net/forum?id=nN3aVRQsxGd>.
- 459 [19] Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. Boosting the cycle counting
460 power of graph neural networks with $i\hat{S}^2\hat{S}$ -gnns. In The Eleventh International Conference
461 on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net,
462 2023. URL <https://openreview.net/pdf?id=kDSmx0spsXQ>.
- 463 [20] Bohang Zhang, Shengjie Luo, Liwei Wang, and Di He. Rethinking the expressive power
464 of gnns via graph biconnectivity. In The Eleventh International Conference on Learning
465 Representations, 2023.
- 466 [21] Ralph Abboud, İsmail İlkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. The surprising
467 power of graph neural networks with random node initialization. In Zhi-Hua Zhou, editor,
468 Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI
469 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, pages 2112–2118. ijcai.org, 2021.
470 doi: 10.24963/ijcai.2021/291. URL <https://doi.org/10.24963/ijcai.2021/291>.
- 471 [22] Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational
472 pooling for graph representations. In International Conference on Machine Learning, pages
473 4663–4673. PMLR, 2019.

- 474 [23] Ronald Aylmer Fisher. Statistical methods for research workers. Springer, 1992.
- 475 [24] Richard A. Johnson and Dean W. Wichern. Applied multivariate statistical analysis. Pearson
476 Prentice Hall, Upper Saddle River, N.J., 6th ed edition, 2007. ISBN 978-0-13-187715-3. OCLC:
477 ocm70867129.
- 478 [25] Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph
479 isomorphism testing and function approximation with gnns. Advances in neural information
480 processing systems, 32, 2019.
- 481 [26] Floris Geerts and Juan L. Reutter. Expressiveness and approximation properties of graph neural
482 networks. In The Tenth International Conference on Learning Representations, ICLR 2022,
483 Virtual Event, April 25-29, 2022. OpenReview.net, 2022. URL [https://openreview.net/
484 forum?id=wIzUeM3TAU](https://openreview.net/forum?id=wIzUeM3TAU).
- 485 [27] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra
486 which appears therein. nti, Series, 2(9):12–16, 1968.
- 487 [28] J.-Y. Cai, M. Furer, and N. Immerman. An optimal lower bound on the number of variables for
488 graph identification. In 30th Annual Symposium on Foundations of Computer Science, pages
489 612–617, 1989. doi: 10.1109/SFCS.1989.63543.
- 490 [29] Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design
491 provably more powerful neural networks for graph representation learning. Advances in Neural
492 Information Processing Systems, 33:4465–4478, 2020.
- 493 [30] A. E. Brouwer, F. Ihringer, and W. M. Kantor. Strongly regular graphs satisfying the 4-vertex
494 condition. Combinatorica, 43(2):257–276, apr 2023. doi: 10.1007/s00493-023-00005-y. URL
495 <https://doi.org/10.1007%2Fs00493-023-00005-y>.
- 496 [31] Andries E Brouwer, Willem H Haemers, Andries E Brouwer, and Willem H Haemers. Strongly
497 regular graphs. Spectra of graphs, pages 115–149, 2012.
- 498 [32] Andries E Brouwer, Willem H Haemers, Andries E Brouwer, and Willem H Haemers.
499 Distance-regular graphs. Springer, 2012.
- 500 [33] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for
501 one-shot image recognition. In ICML deep learning workshop, volume 2. Lille, 2015.
- 502 [34] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an
503 invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and
504 Pattern Recognition (CVPR’06), volume 2, pages 1735–1742. IEEE, 2006.
- 505 [35] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and
506 Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the
507 IEEE conference on computer vision and pattern recognition, pages 5265–5274, 2018.
- 508 [36] Harold Hotelling. The generalization of student’s ratio. In Breakthroughs in statistics:
509 Foundations and basic theory, pages 54–65. Springer, 1992.
- 510 [37] Ronald Aylmer Fisher. Contributions to mathematical statistics. 1950.
- 511 [38] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen,
512 and Tie-Yan Liu. Do transformers really perform badly for graph representation? Advances in
513 Neural Information Processing Systems, 34:28877–28888, 2021.
- 514 [39] Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any
515 GNN with local structure awareness. In International Conference on Learning Representations,
516 2022. URL https://openreview.net/forum?id=Mspk_WYKoEH.
- 517 [40] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful
518 graph networks. Advances in neural information processing systems, 32, 2019.

- 519 [41] Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards
520 scalable higher-order graph embeddings. Advances in Neural Information Processing Systems,
521 33:21824–21840, 2020.
- 522 [42] Lingxiao Zhao, Neil Shah, and Leman Akoglu. A practical, progressively-expressive gnn.
523 Advances in Neural Information Processing Systems, 35:34106–34120, 2022.
- 524 [43] Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving
525 graph neural network expressivity via subgraph isomorphism counting. IEEE Transactions on
526 Pattern Analysis and Machine Intelligence, 45(1):657–668, 2022.
- 527 [44] Pál András Papp, Karolis Martinkus, Lukas Faber, and Roger Wattenhofer. Dropgnn: Random
528 dropouts increase the expressiveness of graph neural networks. Advances in Neural Information
529 Processing Systems, 34:21997–22009, 2021.
- 530 [45] Chendi Qian, Gaurav Rattan, Floris Geerts, Mathias Niepert, and Christopher Morris. Ordered
531 subgraph aggregation networks. Advances in Neural Information Processing Systems, 35:
532 21030–21045, 2022.
- 533 [46] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. Can graph neural networks count
534 substructures? Advances in neural information processing systems, 33:10383–10395, 2020.
- 535 [47] László Babai and Ludik Kucera. Canonical labelling of graphs in linear average time. In 20th
536 Annual Symposium on Foundations of Computer Science (sfcs 1979), pages 39–46. IEEE,
537 1979.
- 538 [48] Ryoma Sato. A survey on the expressive power of graph neural networks. arXiv preprint
539 arXiv:2003.04078, 2020.
- 540 [49] Ningyuan Teresa Huang and Soledad Villar. A short tutorial on the weisfeiler-lehman test and
541 its variants. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and
542 Signal Processing (ICASSP), pages 8533–8537. IEEE, 2021.

543 Checklist

- 544 1. For all authors...
- 545 (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s
546 contributions and scope? [Yes] We elucidate the constraints inherent in prior datasets
547 pertaining to expressiveness and present a novel dataset along with comprehensive
548 experiments.
- 549 (b) Did you describe the limitations of your work? [Yes] We refer to alternative metrics
550 for assessing expressiveness and welcome additional experiments on our foundational
551 dataset (shown in Section 6)
- 552 (c) Did you discuss any potential negative societal impacts of your work? [N/A] All data
553 points are synthetic.
- 554 (d) Have you read the ethics review guidelines and ensured that your paper conforms to
555 them? [Yes]
- 556 2. If you are including theoretical results...
- 557 (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please refer to
558 assumption 4.1.
- 559 (b) Did you include complete proofs of all theoretical results? [Yes] Please refer to proof E.
- 560 3. If you ran experiments (e.g. for benchmarks)...
- 561 (a) Did you include the code, data, and instructions needed to reproduce the main experi-
562 mental results (either in the supplemental material or as a URL)? [Yes] Please refer to
563 <https://github.com/GraphPKU/BREC>
- 564 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they
565 were chosen)? [Yes] Please refer to Appendix J

- 566 (c) Did you report error bars (e.g., with respect to the random seed after running experi-
567 ments multiple times)? [Yes] Please refer to Appendix J
- 568 (d) Did you include the total amount of compute and the type of resources used (e.g., type
569 of GPUs, internal cluster, or cloud provider)? [Yes] Please refer to Appendix J
- 570 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
- 571 (a) If your work uses existing assets, did you cite the creators? [Yes]
- 572 (b) Did you mention the license of the assets? [Yes] All assets were openly accessible, and
573 the licenses for each asset were retained in the corresponding repositories. For more
574 details, please refer to <https://github.com/GraphPKU/BREC>.
- 575 (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
576 Please refer to <https://github.com/GraphPKU/BREC>
- 577 (d) Did you discuss whether and how consent was obtained from people whose data you're
578 using/curating? [N/A] All assets were openly accessible.
- 579 (e) Did you discuss whether the data you are using/curating contains personally identifiable
580 information or offensive content? [N/A] All data points are synthetic.
- 581 5. If you used crowdsourcing or conducted research with human subjects...
- 582 (a) Did you include the full text of instructions given to participants and screenshots, if
583 applicable? [N/A] No utilization of crowdsourcing or engagement in research with
584 human subjects took place.
- 585 (b) Did you describe any potential participant risks, with links to Institutional Review Board
586 (IRB) approvals, if applicable? [N/A] No utilization of crowdsourcing or engagement
587 in research with human subjects took place.
- 588 (c) Did you include the estimated hourly wage paid to participants and the total amount
589 spent on participant compensation? [N/A] No utilization of crowdsourcing or engage-
590 ment in research with human subjects took place.