Open Graph Benchmark:
Datasets for Machine Learning on Graphs

Weihua Hu1, Matthias Fey2, Marinka Zitnik3, Yuxiao Dong4, Hongyu Ren5, Bowen Liu3, Michele Catasta1, Jure Leskovec1
1Department of Computer Science, 5Chemistry, Stanford University
2Department of Computer Science, TU Dortmund University
3Department of Biomedical Informatics, Harvard University
4Microsoft Research, Redmond
ogb@cs.stanford.edu

Steering Committee
Regina Barzilay, Peter Battaglia, Yoshua Bengio, Michael Bronstein, Stephan Günnemann, Will Hamilton, Tommi Jaakkola, Stefanie Jegelka, Maximilian Nickel, Chris Re, Le Song, Jian Tang, Max Welling, Rich Zemel

Abstract

We present the Open Graph Benchmark (OGB), a diverse set of challenging and realistic benchmark datasets to facilitate scalable, robust, and reproducible graph machine learning (ML) research. OGB datasets are large-scale, encompass multiple important graph ML tasks, and cover a diverse range of domains, ranging from social and information networks to biological networks, molecular graphs, source code ASTs, and knowledge graphs. For each dataset, we provide a unified evaluation protocol using meaningful application-specific data splits and evaluation metrics. In addition to building the datasets, we also perform extensive benchmark experiments for each dataset. Our experiments suggest that OGB datasets present significant challenges of scalability to large-scale graphs and out-of-distribution generalization under realistic data splits, indicating fruitful opportunities for future research. Finally, OGB provides an automated end-to-end graph ML pipeline that simplifies and standardizes the process of graph data loading, experimental setup, and model evaluation. OGB will be regularly updated and welcomes inputs from the community. OGB datasets as well as data loaders, evaluation scripts, baseline code, and leaderboards are publicly available at https://ogb.stanford.edu.

1 Introduction

Graphs are widely used for abstracting complex systems of interacting objects, such as social networks [30], knowledge graphs [63], molecular graphs [92], and biological networks [9], as well as for modeling 3D objects [75], manifolds [15], and source code [4]. Machine learning, especially deep learning, on graphs is an emerging field [15, 38]. Recently, significant methodological advances have been made in graph ML [35, 49, 84, 94, 100], which have produced promising results in applications from diverse domains [77, 99, 107].

How can we further advance research in graph ML? Historically, high-quality and large-scale datasets have played significant roles in advancing research, as exemplified by IMAGENET [23] and MS COCO [58] in computer vision, GLUE BENCHMARK [86] and SQUAD [69] in natural language processing, and LIBRISPEECH [64] and CHiME [10] in speech processing. However, in graph
ML research, commonly-used datasets and evaluation procedures present issues that may negatively impact future progress.

**Issues with current benchmarks.** Most of the frequently-used graph datasets are extremely small compared to graphs found in real applications (with more than 1 million nodes or 100 thousand graphs) [12, 43, 85, 87, 92, 99]. For example, the widely-used node classification datasets, CORA, CITESEER, and PUBMED [98], only have 2,700 to 20,000 nodes, the popular graph classification datasets from the TU collection [47, 95] only contain 200 to 5,000 graphs, and the commonly-used knowledge graph completion datasets, FB15k and WN18 [14], only have 15,000 to 40,000 entities. As models are extensively developed on these small datasets, the majority of them turn out to be not scalable to larger graphs [14, 49, 81, 83]. The small datasets also make it hard to rigorously evaluate data-hungry models, such as Graph Neural Networks (GNNs) [28, 34, 56, 94]. In fact, the performance of GNNs on these datasets is often unstable and nearly statistically identical to each other, due to the small number of samples the models are trained and evaluated on [29, 40].

Furthermore, there is no unified and commonly-followed experimental protocol. Different studies adopt their own dataset splits, evaluation metrics, and cross-validation protocols, making it challenging to compare performance reported across various studies [29, 31, 74]. In addition, many studies follow the convention of using random splits to generate training/test sets [14, 49, 94], which is not realistic or useful for real-world applications and generally leads to overly optimistic performance results [59]. An extensive discussion on the shortcomings of the current benchmarks is further provided in Appendix A.

As a result, there is an urgent need for a comprehensive suite of real-world benchmarks that combine a diverse set of datasets of various sizes coming from different domains. Fixed data splits as well as evaluation metrics are important so that progress can be measured in a consistent and reproducible way. Last but not least, benchmarks need to provide different types of tasks, such as node classification, link prediction, and graph classification.

**Present work: OGB.** Here, we present the Open Graph Benchmark (OGB) with the goal of facilitating scalable, robust, and reproducible graph ML research. The premise of OGB is to develop a diverse set of challenging and realistic benchmark datasets that can empower the rigorous advancements in graph ML. As illustrated in Figure 1, the OGB datasets are designed to have the following three characteristics:

1. **Large scale.** The OGB datasets are orders-of-magnitude larger than existing benchmarks and can be categorized into three different scales (small, medium, and large). Even the “small” OGB graphs have more than 100 thousand nodes or more than 1 million edges, but are small enough to fit into the memory of a single GPU, making them suitable for testing computationally intensive algorithms. Additionally, OGB introduces “medium” (more than 1 million nodes or more than 10 million edges) and “large” (on the order of 100 million nodes or 1 billion edges) datasets, which can facilitate the development of scalable models based on mini-batching and distributed training.

2. **Diverse domains.** The OGB datasets aim to include graphs that are representative of a wide variety of domains, ranging from social and information networks to biological networks, molecular graphs, source code ASTs, and knowledge graphs. The broad coverage of domains in OGB empowers the development and demonstration of general-purpose models, and can be used to distinguish them from domain-specific techniques. Furthermore, for each dataset, OGB adopts domain-specific data splits (e.g., based on time, species, molecular structure, GitHub project, etc.) that are more realistic and meaningful than conventional random splits.

3. **Multiple task categories.** Besides data diversity, OGB supports three categories of fundamental graph ML tasks, i.e., node, link, and graph property predictions, each of which requires the models to make predictions at different levels of graphs, i.e., at the level of a node, link, and entire graph, respectively.

The currently-available OGB datasets are summarized in Table 1, and their graph statistics are provided in Table 2. Currently, OGB includes 15 diverse graph datasets, with at least 4 datasets...
Table 1: Summary of currently-available OGB datasets. An OGB dataset, e.g., ogbg-molhiv, is identified by its prefix (ogbg-) and its name (molhiv). The prefix specifies the category of the graph ML task, i.e., node (ogbn-), link (ogbl-), or graph (ogbg-) property prediction. Datasets come from diverse domains: Nature domain includes biological networks and molecular graphs, Society domain includes academic graphs and e-commerce networks, and Information domain includes knowledge graphs. A realistic data split scheme is provided for each dataset, whose detail can be found in Appendices B, C, and D, for each dataset.

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Domain</th>
<th>Node</th>
<th>Edge</th>
<th>Directed</th>
<th>Hetero</th>
<th>#Tasks</th>
<th>Split Scheme</th>
<th>Split Ratio</th>
<th>Task Type</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>ogbn-products</td>
<td>Society</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>Sales rank</td>
<td>8/2/90</td>
<td>Multi-cls.</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>proteins</td>
<td>Nature</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>–</td>
<td>112</td>
<td>Species</td>
<td>65/16/19</td>
<td>Binary class</td>
<td>ROC-AUC</td>
</tr>
<tr>
<td></td>
<td>arxiv</td>
<td>Society</td>
<td>✓</td>
<td>–</td>
<td>✓</td>
<td>–</td>
<td>1</td>
<td>Time</td>
<td>54/18/28</td>
<td>Multi-cls.</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>papers100M</td>
<td>Society</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>1</td>
<td>Time</td>
<td>78/8/14</td>
<td>Multi-cls.</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>mag</td>
<td>Information</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>1</td>
<td>Time</td>
<td>85/9/6</td>
<td>Multi-cls.</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Link</td>
<td>ogbl-ppa</td>
<td>Nature</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>Throughput</td>
<td>70/20/10</td>
<td>Link prediction</td>
<td>Hits@100</td>
</tr>
<tr>
<td></td>
<td>collab</td>
<td>Society</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>Time</td>
<td>92/4/4</td>
<td>Link prediction</td>
<td>Hits@50</td>
</tr>
<tr>
<td></td>
<td>ddi</td>
<td>Nature</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>Protein target</td>
<td>80/10/10</td>
<td>Link prediction</td>
<td>Hits@20</td>
</tr>
<tr>
<td></td>
<td>citation</td>
<td>Society</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>Time</td>
<td>99/1/1</td>
<td>Link prediction</td>
<td>MRR</td>
</tr>
<tr>
<td></td>
<td>wikikg</td>
<td>Information</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>1</td>
<td>Time</td>
<td>94/3/3</td>
<td>KG completion</td>
<td>MRR</td>
</tr>
<tr>
<td></td>
<td>biokg</td>
<td>Information</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>1</td>
<td>Random</td>
<td>94/3/3</td>
<td>KG completion</td>
<td>MRR</td>
</tr>
<tr>
<td>Graph</td>
<td>ogbg-molhiv</td>
<td>Nature</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>1</td>
<td>Scaffold</td>
<td>80/10/10</td>
<td>Binary class</td>
<td>ROC-AUC</td>
</tr>
<tr>
<td></td>
<td>molpcba</td>
<td>Nature</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>128</td>
<td>Scaffold</td>
<td>80/10/10</td>
<td>Binary class</td>
<td>AP</td>
</tr>
<tr>
<td></td>
<td>ppa</td>
<td>Nature</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1</td>
<td>Species</td>
<td>49/29/22</td>
<td>Multi-class class</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td>code</td>
<td>Information</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>1</td>
<td>Project</td>
<td>90/5/5</td>
<td>Sub-token prediction</td>
<td>F1 score</td>
</tr>
</tbody>
</table>

Table 2: Statistics of currently-available OGB datasets. The first 3 statistics are calculated over raw training/validation/test graphs. The last 4 graph statistics are calculated over the ‘standardized’ training graphs, where the graphs are first converted into undirected and unlabeled homogeneous graphs with duplicated edges removed. The SNAP library [53] is then used to compute the graph statistics, where the graph diameter is approximated by performing BFS from 1,000 randomly-sampled nodes. The MaxSCC ratio represents the fraction of nodes in the largest strongly connected component of the graph.

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Scale</th>
<th>#Graphs</th>
<th>Average #Nodes</th>
<th>Average #Edges</th>
<th>Average Node Deg.</th>
<th>Average Clust. Coeff.</th>
<th>MaxSCC Ratio</th>
<th>Graph Diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>ogbn-products</td>
<td>medium</td>
<td>1</td>
<td>2,440,029</td>
<td>61,859,140</td>
<td>50.5</td>
<td>0.411</td>
<td>0.674</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>proteins</td>
<td>medium</td>
<td>1</td>
<td>132,534</td>
<td>39,561,252</td>
<td>597.0</td>
<td>0.280</td>
<td>1.000</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>arxiv</td>
<td>small</td>
<td>1</td>
<td>169,343</td>
<td>1,166,243</td>
<td>13.7</td>
<td>0.226</td>
<td>1.000</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>papers100M</td>
<td>large</td>
<td>1</td>
<td>111,059,956</td>
<td>1,615,685,872</td>
<td>29.1</td>
<td>0.085</td>
<td>1.000</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>mag</td>
<td>medium</td>
<td>1</td>
<td>1,939,743</td>
<td>25,582,108</td>
<td>21.7</td>
<td>0.098</td>
<td>1.000</td>
<td>6</td>
</tr>
<tr>
<td>Link</td>
<td>ogbl-ppa</td>
<td>medium</td>
<td>1</td>
<td>576,289</td>
<td>30,326,273</td>
<td>73.7</td>
<td>0.223</td>
<td>0.674</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>collab</td>
<td>small</td>
<td>1</td>
<td>235,868</td>
<td>1,285,465</td>
<td>8.2</td>
<td>0.729</td>
<td>0.487</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>ddi</td>
<td>small</td>
<td>1</td>
<td>4,263</td>
<td>1,334,889</td>
<td>500.5</td>
<td>0.514</td>
<td>1.000</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>citation</td>
<td>medium</td>
<td>1</td>
<td>2,927,963</td>
<td>30,561,187</td>
<td>20.7</td>
<td>0.178</td>
<td>0.674</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>wikikg</td>
<td>medium</td>
<td>1</td>
<td>2,500,604</td>
<td>17,137,181</td>
<td>12.2</td>
<td>0.168</td>
<td>1.000</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>biokg</td>
<td>small</td>
<td>1</td>
<td>93,773</td>
<td>5,088,434</td>
<td>47.5</td>
<td>0.409</td>
<td>0.674</td>
<td>8</td>
</tr>
<tr>
<td>Graph</td>
<td>ogbg-molhiv</td>
<td>small</td>
<td>1</td>
<td>41,127</td>
<td>25.5</td>
<td>27.5</td>
<td>2.2</td>
<td>0.002</td>
<td>0.674</td>
</tr>
<tr>
<td></td>
<td>molpcba</td>
<td>medium</td>
<td>1</td>
<td>437,929</td>
<td>26.0</td>
<td>28.1</td>
<td>2.2</td>
<td>0.002</td>
<td>0.674</td>
</tr>
<tr>
<td></td>
<td>ppa</td>
<td>medium</td>
<td>1</td>
<td>158,100</td>
<td>243.4</td>
<td>2,266.1</td>
<td>18.3</td>
<td>0.513</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>code</td>
<td>medium</td>
<td>1</td>
<td>452,741</td>
<td>125.2</td>
<td>124.2</td>
<td>2.0</td>
<td>0.0</td>
<td>1.000</td>
</tr>
</tbody>
</table>

for each task category. All the datasets are constructed by ourselves, except for ogbn-products, ogbg-molpcba, and ogbg-molhiv, whose graphs and target labels are adopted from Chiang et al. [17] and Wu et al. [92]. For these datasets, we resolve critical issues of the existing data splits by presenting more meaningful and standardized splits. OGB is a community-driven, open-source initiative. Over time, we plan to release new datasets and tasks, based on the input from the community.

In addition to building the graph datasets, we also perform extensive benchmark experiments for each dataset. Through the experiments and ablation studies, we highlight research challenges and opportunities provided by each dataset, especially on (1) scaling models to large graphs, and (2) improving out-of-distribution generalization performance under the realistic data split scenarios.
Figure 2: Overview of the OGB pipeline: (a) OGB provides realistic graph benchmark datasets that cover different prediction tasks (node, link, graph), are from diverse application domains, and are at different scales. (b) OGB fully automates dataset processing and splitting. That is, the OGB data loaders automatically download and process graphs, provide graph objects (compatible with PyTorch [65] and its associated graph libraries, PyTorch Geometric [33] and Deep Graph Library [88]), and further split the datasets in a standardized manner. (c) After an ML model is developed, (d) OGB evaluates the model in a dataset-dependent manner, and outputs the model performance appropriate for the task at hand. Finally, (e) OGB provides public leaderboards to keep track of recent advances. Steps (b) and (d) are supported by our OGB Python package, whose usage is explained in Appendix E.

Finally, as illustrated in Figure 2, OGB presents an automated end-to-end graph ML pipeline that simplifies and standardizes the process of graph data loading, experimental setup, and model evaluation, in the same spirit as OpenML [32, 82]. Specifically, given the OGB datasets (a), the end-users can focus on developing their graph ML models (c) by using the OGB data loaders (b) and evaluators (d), both of which are provided by our OGB Python package (https://github.com/snap-stanford/ogb). OGB also hosts a public leaderboard (e) for publicizing state-of-the-art, reproducible graph ML research (https://ogb.stanford.edu/docs/leader_overview).

2 OGB Datasets and Benchmark Analyses: Overview

The goal of OGB is to catalyze graph ML research by providing realistic, diverse, and large-scale graph datasets with unified evaluation protocols. Table 1 summarizes the OGB datasets along with their graph types, prediction tasks, as well as evaluation protocols (data splits and evaluation metrics).

In the subsequent sections (Sections 3, 4, and 5), we detail currently-available datasets for each task category. Along with this, we provide an extensive benchmark analysis for each dataset, using representative node embedding models, GNNs, as well as recently-introduced mini-batch-based GNNs. We discuss our initial findings, and highlight research challenges and opportunities in: (1) scaling models to large graphs, and (2) improving out-of-distribution generalization under the realistic data splits. We repeat each experiment 10 times using different random seeds and report the mean and unbiased standard deviation of all training and test results corresponding to the best validation results. All code to reproduce our baseline experiments is publicly available at https://github.com/snap-stanford/ogb/tree/master/examples and is meant as a starting point to accelerate further research on our proposed datasets. We refer the interested reader to our code base for the details of model architectures and hyper-parameter settings.

Finally, we highlight the diversity of our graph datasets by comparing their basic graph statistics in Table 2. Importantly, we observe the diversity in graph structure, beyond the diversity in the dataset scales. For example, comparing the average node degrees, we see that biology-related graphs (e.g., ogbn-proteins, ogbl-ddi, ogbl-ppa, ogbg-ppa) are much denser than the social and information networks. The other statistics (average clustering coefficient and graph diameter) also vary significantly across different datasets. These differences in graph structure result in the inherent difference in how information propagates in the graphs, which can significantly affect the behavior of many graph ML models such as GNNs and random-walk-based node embeddings [93]. For the graph property prediction datasets, it is worth highlighting the diversity of graph sizes (the number of nodes and edges per graph), ranging from small molecular graphs (ogbg-molhiv and ogbg-molpcba), to medium-sized source code ASTs (ogbg-code), up to large and dense protein-protein association subgraphs (ogbg-ppa). Overall, the diversity in graph characteristics originates from the diverse application domains and is crucial to evaluate the versatility of graph ML models.
3 OGB Node Property Prediction

We currently provide 5 datasets, adopted from diverse application domains, for predicting the properties of individual nodes. Specifically, ogbn-products is an Amazon products co-purchasing network [12] originally developed by Chiang et al. [17]. The ogbn-archiv, ogbn-mag, and ogbn-papers100M datasets are extracted from the Microsoft Academic Graph (MAG) [87], with different scales, tasks, and include both homogeneous and heterogeneous graphs. Specifically, ogbn-archiv is a paper citation network of ARXIV papers, ogbn-mag is a heterogeneous academic graph containing different node types (papers, authors, institutions, and topics) and their relations, and ogbn-papers100M is an extremely large paper citation network from the entire MAG with more than 100 million nodes and 1 billion edges. The ogbn-proteins dataset is a protein-protein association network [80]. Below we present the ogbn-products dataset and its baseline experiments. Due to space constraints, we present all the datasets comprehensively in Appendix B.

The ogbn-products dataset. This dataset is an undirected and unweighted graph, representing an Amazon product co-purchasing network [12]. Nodes represent products sold in Amazon, and edges between two products indicate that the products are purchased together. The graphs, target labels, and node features are generated following Chiang et al. [17], where node features are dimensionality-reduced bag-of-words of the product descriptions. Our contribution, when adopting the dataset in OGB, is to resolve its critical data split issue by presenting a more realistic and challenging split (see below).

Prediction task. The task is to predict the category of a product in a multi-class classification setup, where the 47 top-level categories are used for target labels.

Dataset splitting. We consider a more challenging and realistic dataset splitting that differs from the one used in Chiang et al. [17]. Instead of randomly assigning 90% of the nodes for training and 10% of the nodes for testing (without a validation set), we use the sales ranking (popularity) to split nodes into training/validation/test sets. Specifically, we sort the products according to their sales ranking and use the top 8% for training, next top 2% for validation, and the rest for testing. This is a more challenging splitting procedure that closely matches the real-world application where manual labeling is prioritized to important nodes in the network and ML models are subsequently used to make predictions on less important ones.

Baselines. We perform an extensive empirical study, including the representative node embedding model, GNNs, and as well as recently-introduced mini-batch-based GNN models, as baselines.

- MLP: A multilayer perceptron (MLP) predictor that uses the given raw node features directly as input. Graph structure information is not utilized.
- NODE2VEC: An MLP predictor that uses as input the concatenation of the raw node features and NODE2VEC embeddings [35, 66].
- Full-batch GNNs: GCN [49] and GRAPH SAGE (mean pool) [37].
- Mini-batch training of GNNs based on NEIGHBORSAMPLING [37], CLUSTERGCN [17] and GRAPH SAINT [103].

Note that the full-batch GCN and GRAPH SAGE are GPU memory-intensive for large graphs as all the node embeddings need to loaded onto GPU all at once. The mini-batch training techniques, NEIGHBORSAMPLING, CLUSTERGCN, and GRAPH SAINT, do not suffer from this issue and are more GPU memory-efficient. All models are trained with a fixed hidden dimensionality of 256, a fixed number of three layers, and a tuned dropout ratio $\in \{0.0, 0.5\}$.

Results and discussion. Our benchmarking results in Table 3 show that the highest test performances are attained by GNNs, while the MLP baseline that solely relies on a product’s description is not sufficient for accurately predicting the category of a product. Even with the GNNs, we observe the huge generalization gap, which can be explained by differing node distributions across the splits, as visualized in Figure 3. This is in stark contrast with the conventional random split used by Chiang et al. [17]. Even with the same split ratio (8/2/88), we find GRAPH SAGE already achieves 88.20±0.08% test accuracy with only $\approx 1$ percentage points of generalization gap. These results indicate that the realistic split is much more challenging than the random split and offer an important opportunity to improve out-of-distribution generalization.

\[\text{Percentage points of generalization gap.} \]
We currently provide 6 datasets, adopted from diverse application domains, for predicting the properties of links (pairs of nodes). Specifically, ogbl-ppa is a protein-protein association network [80], ogbl-collab is an author collaboration network [87], ogbl-ddi is a drug-drug interaction network [90], ogbl-citation is a paper citation network [87], ogbl-biokg is a heterogeneous knowledge graph compiled from a large number of biomedical repositories, and ogbl-wikikg is a Wikidata knowledge graph [85]. Below we present the ogbl-wikikg dataset and its baseline experiments. Due to space constraints, we present all the datasets comprehensively in Appendix C.

The ogbl-wikikg dataset. This dataset is a Knowledge Graph (KG) extracted from the Wikidata knowledge base [85]. It contains a set of triplet edges (head, relation, tail), capturing the different types of relations between entities in the world, e.g., Canada citizen Hinton. We retrieve all the relational statements in Wikidata and filter out rare entities. Our KG contains 2,500,604 entities and 535 relation types.

Prediction task. The task is to predict new triplet edges given the training edges. The evaluation metric follows the standard filtered metric widely used in KGs [14, 78, 81, 96]. Specifically, we corrupt each test triplet edge by replacing its head or tail with randomly-sampled 1,000 negative entities (500 for head and 500 for tail), while ensuring the resulting triplets do not appear in the KG. The goal is to rank the true head (or tail) entities higher than the negative entities, which is measured by the Mean Reciprocal Rank (MRR).

Dataset splitting. We split the triplets according to time, simulating a realistic KG completion scenario that aims to fill in missing triplets that are not present at a certain timestamp. Specifically, we downloaded Wikidata at three different time stamps (May, August, and November of 2015), and construct three KGs, where we only retain entities and relation types that appear in the earliest May

Table 3 also shows that the recent mini-batch-based GNNs² give promising results, even slightly outperforming the full-batch version of GRAPH SAGE that does not fit into ordinary GPU memory. The improved performance can be attributed to the regularization effects of mini-batch noise and edge dropout [71]. Nevertheless, the mini-batch GNNs have been much less explored compared to the full-batch GNNs due to the prevalent use of the extremely small benchmark datasets such as CORA and CitationSeer. As a result, many important questions remain open, e.g., what mini-batch training methods can induce the best regularization effect, and how to allow mini-batch training for advanced GNNs that rely on large receptive-field sizes [50, 54, 93], since the current mini-batch methods are rather limited by the number of nodes from which they aggregate information. Overall, ogbl-products is an ideal benchmark dataset for the field to move beyond the extremely small graph datasets and to catalyze the development of scalable mini-batch-based graph models with improved out-of-distribution prediction accuracy.

In Appendix B.4, we present ogbl-papers100M, which is even larger and is meant to push the scalability to gigantic web-scale graphs in the real world.

4 OGB Link Property Prediction

We currently provide 6 datasets, adopted from diverse application domains, for predicting the properties of links (pairs of nodes). Specifically, ogbl-ppa is a protein-protein association network [80], ogbl-collab is an author collaboration network [87], ogbl-ddi is a drug-drug interaction network [90], ogbl-citation is a paper citation network [87], ogbl-biokg is a heterogeneous knowledge graph compiled from a large number of biomedical repositories, and ogbl-wikikg is a Wikidata knowledge graph [85]. Below we present the ogbl-wikikg dataset and its baseline experiments. Due to space constraints, we present all the datasets comprehensively in Appendix C.

The ogbl-wikikg dataset. This dataset is a Knowledge Graph (KG) extracted from the Wikidata knowledge base [85]. It contains a set of triplet edges (head, relation, tail), capturing the different types of relations between entities in the world, e.g., Canada citizen Hinton. We retrieve all the relational statements in Wikidata and filter out rare entities. Our KG contains 2,500,604 entities and 535 relation types.

Prediction task. The task is to predict new triplet edges given the training edges. The evaluation metric follows the standard filtered metric widely used in KGs [14, 78, 81, 96]. Specifically, we corrupt each test triplet edge by replacing its head or tail with randomly-sampled 1,000 negative entities (500 for head and 500 for tail), while ensuring the resulting triplets do not appear in the KG. The goal is to rank the true head (or tail) entities higher than the negative entities, which is measured by the Mean Reciprocal Rank (MRR).

Dataset splitting. We split the triplets according to time, simulating a realistic KG completion scenario that aims to fill in missing triplets that are not present at a certain timestamp. Specifically, we downloaded Wikidata at three different time stamps (May, August, and November of 2015), and construct three KGs, where we only retain entities and relation types that appear in the earliest May

2The GRAPH SAGE architecture is used for neighbor aggregation.

²Available at https://archive.org/search.php?query=creator%3A%22Wikidata+editors%22

Figure 3: T-SNE visualization of training/validation/test nodes in ogbl-products.
A promising approach to improve scalability is to distribute training across multiple commodity GPUs. This would require a high-end GPU, which is extremely costly and not scalable to even larger KGs. Nevertheless, we demonstrate that with the limited dimensionality, it obtains the best performance with the increased dimensionality. Specifically, on the high-end GPU with 48GB memory, we are able to train all the models with a smaller number of embedding parameters need to be put onto the GPU memory at once.

First, we see from the upper-half of Table 4 that when the limited embedding dimensionality is used, the baselines are implemented on a single commodity GPU with 11GB memory, while the bottom-half baselines are implemented on a high-end GPU with 48GB memory. Training MRR in Table 4 is an unfiltered metric, as filtering is computationally expensive for the large number of training triplets. A different approach is to share parameters across entities and relations, so that a different number of embedding parameters need to be put onto the GPU memory at once. This means that the training MRR is computed by ranking against randomly-selected negative entities without filtering out triplets that appear in KG. The unfiltered metric has the systematic bias of being smaller than the filtered counterpart (computed by ranking against “true” negative entities, i.e., the resulting triplets do not appear in the KG) [14].

### Baselines
We consider the four representative KG embedding models: `TRANSE` [14], `DISTMULT` [96], `COMPLEX` [81], and `ROTATE` [78]. For KGs with many entities and relations, the embedding dimensionality can be limited by the available GPU memory, as the embeddings need to be loaded into GPU all at once. We therefore choose the dimensionality such that training can be performed on a fixed-budget of GPU memory. Our training procedure follows Sun *et al.* [78], where we perform negative sampling and use margin-based logistic loss for the loss function.

### Results and discussion
Our benchmark results are provided in Table 4, where the upper-half baselines are implemented on a single commodity GPU with 11GB memory, while the bottom-half baselines are implemented on a high-end GPU with 48GB memory. Training MRR in Table 4 is an unfiltered metric, as filtering is computationally expensive for the large number of training triplets. A different approach is to share parameters across entities and relations, so that a different number of embedding parameters need to be put onto the GPU memory at once.

**Table 4: Results for ogbl-wikikg.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Training (Unfiltered)</th>
<th>MRR Validation (Filtered)</th>
<th>Test (Filtered)</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TRANSE</code></td>
<td>0.3326±0.0041</td>
<td>0.2314±0.0035</td>
<td>0.2535±0.0036</td>
</tr>
<tr>
<td><code>DISTMULT</code></td>
<td>0.4131±0.0057</td>
<td>0.3142±0.0066</td>
<td>0.3434±0.0079</td>
</tr>
<tr>
<td><code>COMPLEX</code></td>
<td>0.4605±0.0020</td>
<td>0.3612±0.0063</td>
<td>0.3877±0.0051</td>
</tr>
<tr>
<td><code>ROTATE</code></td>
<td>0.3469±0.0055</td>
<td>0.2366±0.0043</td>
<td>0.2681±0.0047</td>
</tr>
<tr>
<td><code>TRANSE (6×dim)</code></td>
<td>0.6491±0.0022</td>
<td>0.4587±0.0031</td>
<td>0.4536±0.0028</td>
</tr>
<tr>
<td><code>DISTMULT (6×dim)</code></td>
<td>0.4339±0.0011</td>
<td>0.3403±0.0009</td>
<td>0.3612±0.0030</td>
</tr>
<tr>
<td><code>COMPLEX (6×dim)</code></td>
<td>0.4712±0.0045</td>
<td>0.3787±0.0036</td>
<td>0.4028±0.0033</td>
</tr>
<tr>
<td><code>ROTATE (6×dim)</code></td>
<td>0.6084±0.0025</td>
<td>0.3613±0.0031</td>
<td>0.3626±0.0041</td>
</tr>
</tbody>
</table>

Note that our test MRR on ogbl-wikikg is computed using only 500 negative entities per triplet, which is much less than the number of negative entities used to compute MRR in the existing KG datasets, such as FB15k and FB15k-237 (around 15,000 negative entities). Nevertheless, `ROTATE` gives either lower or comparable test MRR on ogbl-wikikg compared to FB15k and FB15k-237 [78].
We currently provide 4 datasets, adopted from 3 distinct application domains, for predicting the properties of entire graphs or subgraphs. Specifically, ogbg-molhiv and ogbg-molpcba are molecular graphs originally curated by Wu et al. [92], ogbg-ppa is a set of protein-protein association subgraphs [108], and ogbg-code is a collection of ASTs of source code [43]. Below we present the ogbg-molhiv and ogbg-molpcba datasets and their baseline experiments. Due to space constraints, we present all the datasets comprehensively in Appendix D.

The ogbg-molhiv and ogbg-molpcba datasets. These datasets are two molecular property prediction datasets adopted from the MOLECULENET [92], and are among the largest of the MOLECULENET datasets. Besides the two main molecule datasets, we also provide the 10 other MOLECULENET datasets, which are summarized and benchmarked in Appendix F. These datasets can be used to stress-test molecule-specific methods [46, 97] and transfer learning [40]. All the molecules are pre-processed using RDKIT [51]. Each graph represents a molecule, where nodes are atoms, and edges are chemical bonds. Input node features are 9-dimensional, containing atomic number and chirality, as well as other additional atom features such as formal charge and whether the atom is in the ring. Input edge features are 3-dimensional, containing bond type, bond stereochemistry as well as an additional bond feature indicating whether the bond is conjugated. Note that the above additional features are not needed to uniquely identify molecules, and are not adopted in the previous work [40, 44]. In the experiments, we perform an ablation study on the molecule features and find that including the additional features improves generalization performance.

The task is to predict the target molecular properties as accurately as possible, where the molecular properties are cast as binary labels, e.g., whether a molecule inhibits HIV virus replication or not. For evaluation metric, we closely follow Wu et al. [92]. Specifically, for ogbg-molhiv, we use ROC-AUC for evaluation. For ogbg-molpcba, as the class balance is extremely skewed (only 1.4% of data is positive) and the dataset contains multiple classification tasks, we use the Average Precision (AP) averaged over the tasks as the evaluation metric.

Dataset splitting. We adopt the scaffold splitting procedure that splits the molecules based on their two-dimensional structural frameworks. The scaffold splitting attempts to separate structurally different molecules into different subsets, which provides a more realistic estimate of model performance in prospective experimental settings. The scaffold splitting was originally proposed by Wu et al. [92] and has been adopted by the follow-up works [40, 44, 70, 97]; however, the precise implementation differs significantly across works, making their results not directly comparable to each other. In OGB, we aim to standardize the scaffold split by adopting its most challenging version where test molecules are maximally diverse.

Baselines. We consider the two representative GNNs: GCN [49] and GIN [94]. We additionally consider augmenting the models with VIRTUAL NODES, where message-passing is performed over an augmented graph with an additional node that is connected to all nodes in the original graph [34, 44, 55]. We use 5-layer GNNs, average graph pooling, a hidden dimensionality of 300, and a tuned dropout ratio of (0, 0.5). To include edge features, we follow Hu et al. [40] and add transformed edge features into the incoming node features.

Results and discussion. Benchmarking results are given in Tables 5 and 6. We see that GIN with both additional features and VIRTUAL NODES provides the best performance in the two datasets. In Appendix F, we show that even for the other MOLECULENET datasets, the additional features provide the best performance.
consistently improve generalization performance. In OGB, we therefore include the additional node/edge features in our molecular graphs.

We further report the performance on the random splitting, keeping the split ratio the same as the scaffold splitting. We find the random split to be much easier than scaffold split. On random splits of ogbg-molhiv and ogbg-molpcba, the best GIN achieves the ROC-AUC of 82.73 ± 0.05 % (5.66 percentage points higher than scaffold) and AP of 34.40 ± 0.05 % (7.37 percentage points higher than scaffold), respectively. The same trend holds true for the other MOLECULENET datasets, e.g., the best GIN performance on the random split of ogbg-moltox21 is 86.03 ± 1.37 %, which is 8.46 percentage points higher than that of the best GIN for the scaffold split (77.57 ± 0.62 % ROC-AUC). These results highlight the challenges of the scaffold split compared to the random split, and opens up a fruitful research opportunity to increase the out-of-distribution generalization capability of GNNs.

6 Conclusions

To enable scalable, robust, and reproducible graph ML research, we introduce the Open Graph Benchmark (OGB)—a diverse set of realistic graph datasets in terms of scales, domains, and task categories. We employ realistic data splits for the given datasets, driven by application-specific use cases. Through extensive benchmark experiments, we highlight that the OGB datasets present significant challenges for ML models to handle large-scale graphs and make accurate prediction under the realistic data splitting scenarios. Altogether, OGB presents fruitful opportunities for future research to push the frontier of graph ML.

OGB is an open-source initiative that provides ready-to-use datasets as well as their data loaders, evaluation scripts, and public leaderboards. We hereby invite the community to develop and contribute state-of-the-art graph ML models at https://ogb.stanford.edu.

Broader Impact

We expect the Open Graph Benchmark (OGB) to have a significant impact on fundamental graph ML research as well as many of its application domains. We also discuss a potential negative impact.

Impact on Graph ML Research

Historically, high-quality and large-scale datasets have played significant roles in advancing research fields (e.g., IMAGENET [23], MS-COCO [58], GLUE benchmark [86], SQUAD [69]). The amount of impact these datasets have brought is enormous, leading to the significant methodological advancements in the respective fields [24, 39].

We expect OGB to be a standard benchmark in graph ML, contributing to the significant advancements of the field. To this end, our datasets are carefully designed to address the two major drawbacks of current graph benchmark datasets, namely (1) small dataset sizes, and (2) unrealistic random splits. Overall, OGB provides a set of diverse, realistic, and large-scale graph datasets to facilitate the development of graph ML models that are scalable and generalizable under realistic data splits, both of which are crucial in practice.

In addition, OGB aims to address the fundamental problem of reproducibility in graph ML research. We promote the reproducibility by standardizing the research pipeline, as illustrated in Figure 2, and provide official leaderboards, for which public code is mandatory to make a submission. Altogether, OGB incentivises researchers to release their code, and allows researchers to easily compare different models under equal settings.

Impact on Diverse Application Domains

In OGB, we have curated graph datasets that are relevant to a variety of practical and realistic application domains, including science (e.g., biology, chemistry), knowledge graphs, academic graphs, and source code ASTs. For example, we provided a biomedical knowledge graph (ogbl-biokg), where algorithmic advances on this dataset can be immediately translated into solutions for problems in precision medicine. In another example, we provide a dataset of 450K molecular graphs (ogbg-molpcba) that have direct implications for drug discovery. In academic domains, we
prepared a variety of prediction tasks (e.g., recommending missing citations as well as future collaborations, predicting paper categories and venues, etc.), solving which can lead to improved scholarly efficiency and to better organization of academic knowledge. In technological domains, OGB includes a dataset of source code snippets (ogbg-code). The development of graph ML models on this dataset can lead to exciting applications for advanced code analysis and retrieval.

To further increase the impact of OGB, all of our datasets are mapped to real entities in the world. For example, each node in the drug-drug interaction network (ogbl-ddi) is mapped to a unique Drug ID in DrugBank [90], each molecule in the molecule datasets (ogbg-mol*) is mapped to a SMILES string that uniquely identifies the molecule, and each node in the paper citation networks (ogbn-arxiv and ogbn-papers100M) is mapped into a real research paper indexed by the Microsoft Academic Graph [87]. Such mappings to real entities allow researchers to draw scientific insight and to augment the graphs with external information.

**Potential Negative Impact**

If OGB becomes the standard de-facto benchmark for graph ML, one potential negative impact is that OGB might contribute to narrowing down the scope of future papers to the tasks and dataset types that have been included in OGB. In order to avoid such a negative impact, we will regularly update our datasets and tasks, based on the input from the community.

**Acknowledgements**

We thank Adrijan Bradaschia and Rok Sosic for their help in setting up the server and website. We also thank Emma Pierson and Shigeru Maya for their suggestions on the paper writing. Finally, we thank the entire community of graph ML for providing valuable feedback to improve OGB. Weihua Hu is supported by Funai Overseas Scholarship and Masason Foundation Fellowship. Matthias Fey is supported by the German Research Association (DFG) within the Collaborative Research Center SFB 876 “Providing Information by Resource-Constrained Analysis”, project A6. Marinka Zitnik is in part supported by NSF IIS-2030459. We gratefully acknowledge the support of DARPA under Nos. FA865018C7880 (ASED), N660011924033 (MCS); ARO under Nos. W911NF-16-1-0342 (MURI), W911NF-16-1-0171 (DURIP); NSF under Nos. OAC-1835598 (CINES), OAC-1934578 (HDR), CCF-1918940 (Expeditions), IIS-2030477 (RAPID); Stanford Data Science Initiative, Wu Tsai Neurosciences Institute, Chan Zuckerberg Biohub, Amazon, Boeing, JPMorgan Chase, Docomo, Hitachi, JD.com, KDDI, NVIDIA, Dell. Jure Leskovec is a Chan Zuckerberg Biohub investigator.

**References**


