OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs

Weihua Hu [1] **Matthias Fey** [2] **Hongyu Ren** [1] **Maho Nakata** [3] **Yuxiao Dong** [4] **Jure Leskovec** [1]

Abstract

Enabling effective and efficient machine learning (ML) over large-scale graph data (*e.g.*, graphs with billions of edges) can have a huge impact on both industrial and scientific applications. However, community efforts to advance large-scale graph ML have been severely limited by the lack of a suitable public benchmark. For KDD Cup 2021, we present OGB Large-Scale Challenge (OGB-LSC), a collection of three real-world datasets for advancing the state-of-the-art in large-scale graph ML. OGB-LSC provides graph datasets that are orders of magnitude larger than existing ones and covers three core graph learning tasks—link prediction, graph regression, and node classification. Furthermore, OGB-LSC provides dedicated baseline experiments, scaling up expressive graph ML models to the massive datasets. We show that the expressive models significantly outperform simple scalable baselines, indicating an opportunity for dedicated efforts to further improve graph ML at scale. Our datasets and baseline code are released and maintained as part of our OGB initiative (Hu et al., 2020a). We hope OGB-LSC at KDD Cup 2021 can empower the community to discover innovative solutions for large-scale graph ML.[1]

1. Introduction

Machine Learning (ML) on graphs has attracted immense attention in recent years because of the prevalence of graph-structured data in real-world applications. Modern application domains include web-scale social networks (Ugander et al., 2011), recommender systems (Ying et al., 2018), hyperlinked web documents (Kleinberg, 1999), knowledge graphs (KGs) (Bollacker et al., 2008; Vrandečić & Krötzsch, 2014), as well as molecule simulation data generated by the ever-increasing scientific computation (Chanussot et al., 2020; Nakata & Shimazaki, 2017). These domains involve large-scale graphs with billions of edges or a dataset with millions of graphs. Deploying accurate graph ML at scale will have a huge practical impact, enabling better recommendation results, improved web document search, more comprehensive KGs, and accurate ML-based drug and material discovery.

Table 1. Basic statistics of the three OGB-LSC datasets.

Task type	Dataset	Statistics	
Node-level	`MAG240M-LSC`	#nodes:	244,160,499
		#edges:	1,728,364,232
Link-level	`WikiKG90M-LSC`	#nodes:	87,143,637
		#edges:	504,220,369
Graph-level	`PCQM4M-LSC`	#graphs:	3,803,453
		#edges (total):	55,399,880

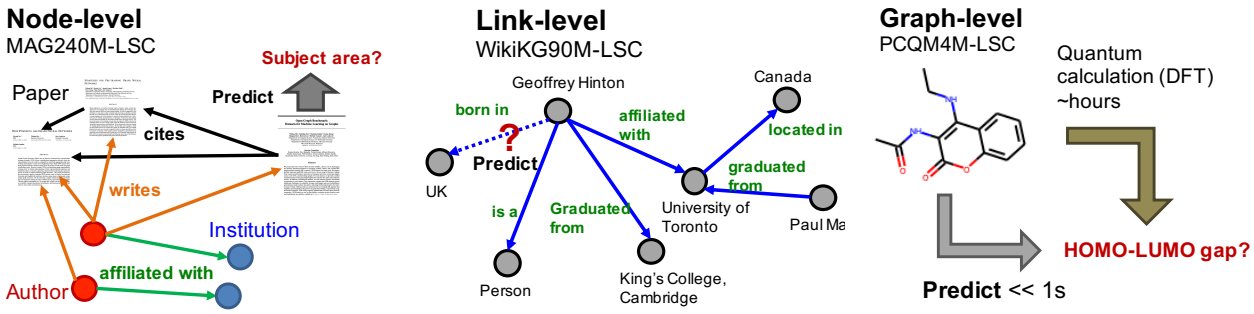
However, community efforts to advance state-of-the-art in large-scale graph ML have been extremely limited. In fact, most of graph ML models have been developed and evaluated on extremely small datasets (Bordes et al., 2013; Morris et al., 2020; Yang et al., 2016).

Handling large-scale graphs is challenging, especially for state-of-the-art expressive Graph Neural Networks (GNNs) (Hamilton et al., 2017; Kipf & Welling, 2017; Velickovic et al., 2018) because they make prediction on each node based on the information from many other nodes. Effectively training these models at scale requires sophisticated algorithms that are well beyond standard SGD over i.i.d. data (Chen et al., 2018; Chiang et al., 2019; Hamilton et al., 2017; Zeng et al., 2020). More recently, researchers improve model scalability by significantly simplifying GNNs (Huang et al., 2020; Rossi et al., 2020; Wu et al., 2019), which inevitably limits their expressive power.

However, in deep learning, it has been demonstrated over and over again that one needs big expressive models and train them on big data to achieve the best performance (Devlin et al., 2018; He et al., 2016; Russakovsky et al., 2015; Vaswani et al., 2017). In graph ML, the trend has been the opposite—models get simplified and less expressive to be able to scale to large graphs. Thus, there is a massive opportunity to move the community to work with realistic and large-scale graph datasets and move the state of the field

[1]Stanford University [2]TU Dortmund University [3]RIKEN [4]Facebook AI. Correspondence to: OGB-LSC Team <ogb-lsc@cs.stanford.edu>.

[1]https://ogb.stanford.edu/kddcup2021/



(a) MAG240M-LSC (b) WikiKG90M-LSC (c) PCQM4M-LSC

Figure 1. **Overview of the three OGB-LSC datasets, covering node-, link-, and graph-level prediction tasks. (a)** MAG240M-LSC is a heterogeneous academic graph, and the task is to predict the subject areas of papers (*cf.* Section 2.1) situated in the heterogeneous graph. **(b)** WikiKG90M-LSC is a knowledge graph, and the task is to impute missing triplets (*cf.* Section 2.2). **(c)** PCQM4M-LSC is a quantum chemistry dataset, and the task is to predict an important molecular property, the HOMO-LUMO gap, of a given molecule (*cf.* Section 2.3).

forward to where it needs to be.

For KDD Cup 2021, we present a large-scale graph ML competition, OGB Large-Scale Challenge (OGB-LSC), to encourage the development of state-of-the-art graph ML models for massive modern datasets. Specifically, we introduce three large-scale, realistic, and challenging datasets: MAG240M-LSC, WikiKG90M-LSC, PCQM4M-LSC, that are unprecedentedly large in scale (see Table 1) and cover prediction at the level of nodes, links, and graphs, respectively. An overview of the three OGB-LSC datasets is provided in Figure 1. For each dataset, we carefully design its prediction task and data split so that achieving high prediction performance on the task will have direct impact on the corresponding application, as we detail in Section 2.

Beyond providing the datasets, we perform an extensive baseline analysis on each dataset, implementing simple baseline models as well as advanced expressive models at scale. We find that advanced expressive models, despite requiring more efforts to scale up, do benefit from large data and significantly outperform simple baseline models that are easy to scale. All of our baseline code is made publicly available to facilitate public research.[2]

Overall, OGB-LSC will encourage the community to develop and scale up expressive graph ML models, and can yield breakthrough in their performance. We hope OGB-LSC at KDD Cup 2021 will serve as an "ImageNet Large Scale Visual Recognition Challenge" (Russakovsky et al., 2015) in the field of graph ML, encouraging the community to work on the realistic and large-scale graph datasets and significantly advance the state-of-the-art.

[2] https://github.com/snap-stanford/ogb/tree/master/examples/lsc

2. OGB-LSC Datasets

W describe the three datasets in OGB-LSC, covering three key task categories (node-, link-, and graph-level prediction tasks) of ML on graphs.

We particularly emphasize the practical relevance and data split for each dataset, making our task closely aligned to realistic applications. Through our extensive baseline analysis, we show that advanced expressive models tend to give much better performance than simple graph ML models, leaving room for further improvement.

2.1. MAG240M-LSC: Node-Level Prediction

Practical relevance and dataset overview. The volume of scientific publication has been increasing exponentially, doubling every 12 years (Dong et al., 2017). Currently, subject areas of ARXIV papers are manually determined by the paper's authors and ARXIV moderators. An accurate automatic predictor of papers' subject categories not only reduces the significant burden of manual labeling, but can also be used to classify the vast number of non-ARXIV papers, thereby allowing better search and organization of academic papers.

MAG240M-LSC is a heterogeneous academic graph extracted from the Microsoft Academic Graph (MAG) (Wang et al., 2020). Given arXiv papers situated in the heterogeneous graph, whose schema diagram is illustrated in Figure 2, we aim to automatically annotating their topics, *i.e.*, predicting the primary subject area of each ARXIV paper.

Graph. We extract 121M academic papers in English from MAG (version: 2020-11-23) to construct a heterogeneous academic graph. The resultant paper set is written by 122M author entities, who are affiliated with 26K institutes. Among these papers, there are 1.3 billion citation links captured by MAG. Each paper is associated with its natural

language title and most papers' abstracts are also available. We concatenate the title and abstract by period and pass it to a RoBERTa sentence encoder (Liu et al., 2019; Reimers & Gurevych, 2019), generating a 768-dimensional vector for each paper node. Among the 121M paper nodes, approximately 1.4M nodes are arXiv papers annotated with 153 arXiv subject areas, *e.g.*, `cs.LG (Machine Learning)`.

Prediction task and evaluation metric. The task is to predict the primary subject areas of the given arXiv papers, which is cast as an ordinary multi-class classification problem. The metric is the classification accuracy.

To understand the relation between the prediction task and the heterogeneous graph structure, we analyze the graph homophily (McPherson et al., 2001)—tendency of two adjacent nodes to share the same labels—to better understand the interplay between heterogeneous graph connectivity and the prediction task. Homophily is normally analyzed over a homogeneous graph, but we extend the analysis to the heterogenous graph by considering meta-paths (Sun et al., 2011)—a path consisting of a sequence of relations defined between different node types. Given a meta-path, we can say two nodes are adjacent if they are connected by the meta-path. Table 2 shows the homophily for different kinds of meta-paths with different levels of connection strength. Compared to the direct citation connection (*i.e.*, P-P), certain meta-paths (*i.e.*, P-A-P) give rise to much higher degrees of homophiliness, while other meta-paths (*i.e.*, P-A-I-A-P) provide much less homophily. As homophily is the central graph property exploited by many graph ML models, we believe that discovering essential heterogeneous connectivity is important to achieve good performance on this dataset.

Table 2. **Analysis of graph homophily for different meta-paths connecting 1,251,341 arXiv papers (only train+validation).** Connection strength indicates the number of different possible paths along the template meta-path, *e.g.*, meta-path "Paper-Author-Paper (P-A-P)" with connection strength 3 means that at least 3 authors are shared for the two papers of interest. Homophily ratio is the ratio of two nodes having the same target labels.

Meta-path	Connect. strength	Homophily ratio (%)	#Edges
P-P	1	57.80	2,017,844
P-A-P	1	46.12	88,099,071
	2	57.02	12,557,765
	4	64.03	1,970,761
	8	66.65	476,792
	16	70.46	189,493
P-A-I-A-P	1	3.83	159,884,165,669
	2	4.61	81,949,449,717
	4	5.69	33,764,809,381
	8	6.85	12,390,929,118
	16	7.70	4,471,932,097
All pairs	0	1.99	782,926,523,470

Dataset split. We split the data according to time. Specifically, we train models on arXiv papers published until 2018, validate the performance on the 2019 papers, and finally test the performance on the 2020 papers. The split reflects the practical scenario of helping the authors and moderators annotate the subject areas of the newly-published arXiv papers.

Baseline. We consider a broad range of graph ML models for our initial benchmarking efforts in both homogeneous (where only paper to paper relations are considered) and full heterogeneous settings. For both settings, we convert the directed graph into an undirected graph for simplicity. First, for the homogeneous setting, we benchmark the simple baseline models: graph-agnostic MLP, Label Propagation, and the recently-proposed simplified graph methods: SGC (Wu et al., 2019), SIGN (Rossi et al., 2020) and MLP+C&S (Huang et al., 2020), which are inherently scalable by decoupling predictions from propagation. Furthermore, we benchmark state-of-the-art expressive GNNs trained with neighborhood sampling (NS) (Hamilton et al., 2017), where we recursively sample 25 neighbors in the first layer and 15 neighbors in the second layer during training time. At inference time, we sample at most 160 neighbors for each layer. Here, we benchmark two types of strong models: the GraphSAGE (Hamilton et al., 2017) model (performing mean aggregation and utilizing skip-connections), and the more advanced Graph Attention Network (GAT) model (Velickovic et al., 2018). For the full heterogeneous setting, we follow Schlichtkrull et al. (2018) and learn distinct weights for each individual relation type (denoted by R-GraphSAGE and R-GAT, where "R" stands for "Relational"). We obtain the input features of authors and institutions by averaging the features of papers belonging to the same author and institution, respectively. The models are trained with NS. We note that the expressive GNNs trained with NS require more efforts to scale up, but are more expressive than the simple baselines.

Hyper-parameters. Hyper-parameters are selected based on their best validation performance. For all the models without NS, we tuned the hidden dimensionality $\in \{128, 256, 512, 1024\}$, MLP depth $\in \{1, 2, 3, 4\}$, dropout ratio $\in \{0, 0.25, 0.5\}$, propagation layers (for SGC, SIGN, and C&S) $\in \{2, 3\}$. For all the GNN models with NS, we use a hidden dimensionality of 1024. We make use of batch normalization (Ioffe & Szegedy, 2015) and ReLU activation in all models. All models were trained using the Adam optimizer (Kingma & Ba, 2015).

Discussion. Validation and test performances of all models

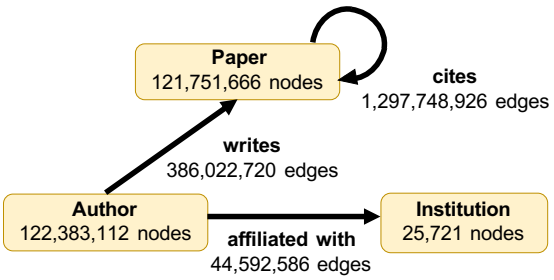


Figure 2. **A schema diagram of `MAG240M-LSC`.** Paper nodes have text features and publication year information.

Table 3. **Results of `MAG240M-LSC` measured by the accuracy.** R-GRAPHSAGE and R-GAT utilize the full heterogeneous graph information, while the other models operate on the homogeneous paper citation graph.

Model	#Params	Validation	Test
MLP	0.5M	52.67	52.73
LABELPROP	0	58.44	56.29
SGC	0.7M	65.82	65.29
SIGN	3.8M	66.64	66.09
MLP+C&S	0.5M	66.98	66.18
GRAPHSAGE (NS)	4.9M	67.32	66.25
GAT (NS)	4.9M	67.71	66.63
R-GRAPHSAGE (NS)	12.2M	70.21	68.94
R-GAT (NS)	12.3M	**70.48**	**69.49**

considered are shown in Table 3. First, the graph-agnostic MLP and Label Propagation algorithm perform poorly, indicating that both graph structure and feature information are indeed important for the given task. Across the graph ML models operating on the homogeneous paper graph, GNNs with NS perform the best, with slight gains compared to their simplified versions. In particular, the advanced expressive graph attention aggregation is favourable compared to the uniform mean aggregation in GRAPHSAGE. Furthermore, considering all available heterogeneous relational structure in the heterogeneous graph setting yields significant improvements, with performance gains up to 3 percentage points. Again, the advanced attention aggregation provides favorable performance.

Overall, our experiments highlight the benefits of developing and evaluating advanced expressive models on the larger scale. However, due to the exponential neighbor expansion in NS, we were only able to train the expressive GNNs up to two layers, while the other simplified models utilize three-layer propagation to achieve their best performance. It is an interesting direction to explore how to train deeper and even more expressive GNNs and whether they can yield further improvements.

Table 4. **Analysis of how entities in training and validation triplets connect to the training KG in the `WikiKG90M-LSC` dataset.** For head (resp. tail) in each triplet, we count the number of training triplets (excluding the triplet of interest) that also contain the entity as head (resp. tail). A large (resp. small) count value implies that the triplet is densely (resp. sparsely) connected to the training KG. As the summary statistics over the training and validation triplets, we report the average and the Proportion below Threshold (PbT), *i.e.*, percentage of triplets whose counts are below a threshold.

Split	Connectivity of triplets to training KG (head)			
	Avg	PbT≤ 5	PbT≤ 2	PbT≤ 1
Train	28.0	43.4%	24.6%	8.8%
Validation	6.5	74.6%	48.5%	32.1%
Split	**Connectivity of triplets to training KG (tail)**			
	Avg	**PbT≤ 25**	**PbT≤ 10**	**PbT≤ 5**
Train	3,394,933	21.2%	12.3%	7.8%
Validation	341,743	28.9%	24.3%	21.0%

2.2. `WikiKG90M-LSC`: Link-Level Prediction

Practical relevance and dataset overview. Large encyclopedic Knowledge Graphs (KGs), such as Wikidata (Vrandečić & Krötzsch, 2014) and Freebase (Bollacker et al., 2008), represent factual knowledge about the world through triplets connecting different entities, *e.g.*, *Hinton* $\xrightarrow{citizenof}$ *Canada*. They provide rich structured information about many entities, aiding a variety of knowledge-intensive downstream applications such as information retrieval, question answering (Singhal, 2012), and recommender systems (Guo et al., 2020). However, these large KGs are known to be far from complete (Min et al., 2013), missing many relational information between entities. Using ML to automatically impute missing triplets significantly reduces the manual curation of knowledge and provides a more comprehensive KG, which in turn improves the aforementioned downstream applications.

`WikiKG90M-LSC` is a Knowledge Graph (KG) extracted from the *entire* Wikidata knowledge base. The task is to automatically impute missing triplets that are not yet present in the current KG. Accurate imputation models can be readily deployed on the Wikidata to improve its coverage.

Graph. Each triplet (head, relation, tail) in `WikiKG90M-LSC` represents an Wikidata claim, where head and tail are the Wikidata items, and relation is the Wikidata predicate. We extracted triplets from the public Wikidata dump downloaded at three time-stamps: September 28, October 26, and November 23 of 2020, for training, validation, and testing, respectively. We retain all the entities and relations in the September dump, resulting in 87,143,637 entities, 1,315 relations, and 504,220,369 triplets in total.

Table 5. **Textual representation of validation triplets whose head entities only *appear once* as head in the training KG.**

Head	Relation	Tail
Food and drink companies of Bulgaria	combines topics	Bulgaria
Performing arts in Denmark	combines topics	performing arts
Anglicanism in Grenada	combines topics	Anglicanism
Chuan Li	occupation	researcher
Petra Junkova	given name	Petra

In addition to extracting triplets, we provide text features for entities and relations. Specifically, each entity/relation in Wikidata is associated with a title and a short description, *e.g.*, one entity is associated with the title 'Geoffrey Hinton' and the description 'computer scientist and psychologist'. Similar to `MAG240M-LSC`, we provide RoBERTa embeddings (Liu et al., 2019; Reimers & Gurevych, 2019) as node and edge features.[3]

Prediction task and evaluation metric. The task is the KG completion, *i.e.*, given a set of training triplets, predict a set of test triplets. For evaluation, we follow the protocol similar to how KG completion is evaluated (Bordes et al., 2013).[4] Specifically, for each validation/test triplet, (head, relation, tail), we corrupt tail with randomly-sampled 1000 negative entities, *e.g.*, tail_neg, such that (head, relation, tail_neg) does not appear in the train/validation/test KG. The model is asked to rank the 1001 candidates (consisting of 1 positive and 1000 negatives) for each triplet and predict the top 10 entities that are most likely to be positive. The goal is to rank the ground-truth positive entity as high in the rank as possible, which is measured by Mean Reciprocal Rank (MRR). [5]

Dataset split. We split the triplets according to time, simulating a realistic KG completion scenario of imputing missing triplets not present at a certain timestamp. Specifically, we construct three KGs using the aforementioned September, October, and November KGs, where we only retain entities and relation types that appear in the earliest September KG. We use the triplets in the September KG for training, and use the additional triplets in the October and November KGs for validation and test, respectively.

We analyze the time split in Table 4. We observe that head entities of validation triplets are much less well-connected to the training KG compared to training triplets, which is expected since new triplets tend to be added for less popular entities. The sparse connectivity to the training KG suggests that learning signals for predicting validation (and test) triplets are extremely sparse. Nonetheless, even for the sparsely-connected triplets, we find the textual information provides important clues, as illustrated in Table 5. Hence, we expect that advanced graph models that effectively incorporate textual information will be key to achieve good performance on the challenging time split.

Baseline. We consider two representative KG embedding models: TransE (Bordes et al., 2013) and ComplEx (Trouillon et al., 2016). These models define their own *decoders* to score knowledge triplets using the corresponding entity and relation embeddings. For instance, TransE uses $-\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|_2$ as the decoder, where $\boldsymbol{h}$, $\boldsymbol{r}$, and $\boldsymbol{t}$ are embeddings of head, relation, and tail, respectively. For the encoder function (mapping each entity and relation to its embedding), we consider the following three options. **Shallow:** We use the distinct embedding for each entity and relation, as normally done in KG embedding models. **RoBERTa:** We use two MLP encoders (one for entity and another for relation) that transform the RoBERTa features into entity and relation embeddings. This method has an extremely small number of learnable parameters since the MLPs are shared across all entities and relations. **Concat:** To enhance the expressive power of the previous encoder, we augment it with shallow embeddings. Specifically, we concatenate the shallow learnable embeddings into the RoBERTa features, and use the MLPs to transform the concatenated vectors to get the final embeddings. This way, the MLP encoders can adaptively utilize the RoBERTa features and the shallow embeddings to fit the large amount of triplet data. To implement our baselines, we utilize DGL-KE (Zheng et al., 2020).

Hyper-parameters. For the loss function, we use the negative sampling loss from Sun et al. (2019), where we pick margin γ from $\{1,4,8,10,100\}$. In order to balance the performance and the memory cost, we use the embedding dimensionality of 200 for all the models. The learning rate is chose from $\{0.01, 0.1\}$ and we train all models for 1M steps.

Discussion. Table 6 shows the validation and test performance of the six different models, *i.e.*, combination of two decoders (TransE and ComplEx) and three encoders (Shallow, RoBERTa, and Concat). Notably, in terms of the encoders, we see that the most expressive Concat

[3]We concatenate the title and description with comma, *e.g.*, 'Geoffrey Hinton, computer scientist and psychologist', and pass the sentence to a RoBERTa sentence encoder (Note that the RoBERTa model was trained before September 2020, so there is no obvious information leak). The title or/and description are sometimes missing, in which case we simply use the blank sentence to replace it.

[4]One difference compared to the standard KG completion evaluation (Bordes et al., 2013) is that we evaluate the corruption of only tail entities. This is because in Wikidata, (head, relation, tail) represents a particular claim about the head entity. Hence, predicting the tail from the head and relation, *e.g.*, (Geoffrey Hinton, citizen of, ?), is often more natural and useful than predicting head entity from the tail and relation, *e.g.*, (?, citizen of, Canada).

[5]Note that this is more strict than the standard MRR since there is no partial score for positive entities being ranked outside of top 10.

Table 6. **Results of `WikiKG90M-LSC` measured by Mean Reciprocal Rank (MRR).**

Model	#Params	Validation	Test
TransE-Shallow	17.4B	0.7559	0.7412
ComplEx-Shallow	17.4B	0.6142	0.5883
TransE-RoBERTa	0.3M	0.6039	0.6288
ComplEx-RoBERTa	0.3M	0.7052	0.7186
TransE-Concat	17.4B	**0.8494**	0.8548
ComplEx-Concat	17.4B	0.8425	**0.8637**

outperforms both Shallow and RoBERTa, indicating that both the textual information (captured by the RoBERTa embeddings) and structural information (captured by node-wise learnable embeddings) are useful in predicting validation and test triplets. In terms of the decoders, TransE and ComplEx show similar performance with the Concat encoder, while they show somewhat mixed results with the Shallow and RoBERTa encoders.

Overall, our experiments suggest that the expressive encoder that combines both textual information and structural information gives the most promising performance. In the KG completion literature, the design of the encoder has been much less studied compared to the decoder designs. Therefore, we believe there is a huge opportunity in scaling up more advanced encoders, especially GNNs (Schlichtkrull et al., 2018), to further improve the performance on this dataset.

2.3. `PCQM4M-LSC`: Graph-Level Prediction

Practical relevance and dataset overview. Density Functional Theory (DFT) is a powerful and widely-used quantum physics calculation that can accurately predict various molecular properties such as the shape of molecules, reactivity, responses by electromagnetic fields (Helgaker et al., 2014). However, DFT is time-consuming and takes up to several hours per small molecule. Using fast and accurate ML models to approximate DFT enables diverse downstream applications, such as property prediction for organic photovaltaic devices (Cao & Xue, 2014) and structure-based virtual screening for drug discovery (Ferreira et al., 2015).

`PCQM4M-LSC` is a quantum chemistry dataset originally curated under the PubChemQC project (Nakata, 2015; Nakata & Shimazaki, 2017). Based on the PubChemQC, we define a meaningful ML task of predicting DFT-calculated HOMO-LUMO energy gap of molecules given their 2D molecular graphs. The HOMO-LUMO gap is one of the most practically-relevant quantum chemical properties of molecules since it is related to reactivity, photoexcitation, and charge transport (Griffith & Orgel, 1957). Moreover, predicting the quantum chemical property only from 2D molecular graphs without their 3D equilibrium structures is also practically favorable. This is because obtaining 3D equilibrium structures requires DFT-based geometry optimization, which is expensive on its own.

Graph. We provide molecules as the SMILES strings (Weininger, 1988), from which 2D molecule graphs (nodes are atoms and edges are chemical bonds) as well as molecular fingerprints (hand-engineered molecular feature developed by the chemistry community) can be obtained. By default, we follow OGB (Hu et al., 2020a) to convert the SMILES string into a molecular graph representation, where each node is associated with a 9-dimensional feature (*e.g.*, atomic number, chirality) and each edge comes with a 3-dimensional feature (*e.g.*, bond type, bond stereochemistry). Hu et al. (2020a) show that using the rich graph features beyond the atomic number and the bond type is often effective in improving generalization capability of GNNs, although the optimal set of input graph features remains to be explored.

Prediction task and evaluation metric. The task is graph regression: predicting the HOMO-LUMO energy gap in electronvolt (eV) given 2D molecular graphs. Mean Absolute Error (MAE) is used as evaluation metric.

Dataset split. Following OGB, we adopt the scaffold split with ratio 80/10/10 (Wu et al., 2018). Specifically, we split the molecules based on their 2D structural frameworks, resulting in validation and test molecules that are structurally very different from training ones. Prediction over out-of-distribution molecular structure is commonplace in ML-based virtual screening. This is because training molecules represent an extremely tiny and biased subset of the entire chemical space (estimated to be around 10^{60} molecules) (Reymond & Awale, 2012), to which fast ML models are applied for virtual screening.

Baseline. We benchmark two types of models: a simple MLP over the Morgan fingerprint (Morgan, 1965) and more advanced GNN models. For GNNs, we use the four strong models developed for graph-level prediction: Graph Convolutional Network (GCN) (Kipf & Welling, 2017) and Graph Isomorphism Network (GIN) (Xu et al., 2019), as well as their variants, GCN-virtual and GIN-virtual, which augment graphs with a virtual node that is bidirectionally connected to all nodes in the original graph (Gilmer et al., 2017). Adding the virtual node is shown to be effective across a wide range of graph-level prediction datasets in OGB (Hu et al., 2020a). Edge features are incorporated following Hu et al. (2020b). At inference time, the model output is clamped between 0 and 50.[6]

[6]These values are set because HOMO/LUMO gap is always positive and the maximum target value in train and validation sets is 47.02. The clamping is crucial to avoid model's anomalously large/small prediction on a small number molecules that skews the

Hyper-parameters. For the MLP over Morgan fingerprint, we set the fingerprint dimensionality to be 2048, and tune the fingerprint radius $\in \{2, 3\}$, as well as MLP's hyper-parameters: hidden dimensionality $\in \{1200, 1600\}$, number of hidden layers $\in \{2, 4, 6\}$, and dropout ratio $\in \{0, 0.2\}$. For GNNs, we tune hidden dimensionality, *i.e.*, width $\in \{300, 600\}$, number of GNN layers, *i.e.*, depth $\in \{3, 5\}$. Simple summation is used for graph-level pooling and dropout rate is set to 0. For all MLPs (including GIN's), we use batch normalization (Ioffe & Szegedy, 2015) and ReLU activation. For all models, we use the Adam optimizer (Kingma & Ba, 2015), batch size of 256, and train models for 100 epochs. The initial learning rate is set to 0.001, which is multiplied by 0.25 every 30 epochs. For our ablation study using only 10% of training data (hence, 1 epoch is 10 times shorter), we train models for 1000 epochs and anneal the learning rate every 300 epochs.

Discussion. The validation and test results are shown in Table 7. We see both the GNN models significantly outperform the simple fingerprint baseline. Expressive GNNs (GIN and GIN-VIRTUAL) outperform less expressive ones (GCN and GCN-VIRTUAL); especially, the most advanced and expressive GIN-VIRTUAL model significantly outperforms the other GNNs. Nonetheless, the current performance is still much worse than the chemical accuracy of 0.043eV—an indicator of practical usefulness established by the chemistry community.

In the same Table 7, we show our ablation, where we use only 10% of data to train the GIN-VIRTUAL model. We see the performance significantly deteriorate, indicating the importance of training the model on large data.

Finally, in Table 8, we show the relation between model sizes and validation performance. We see that the largest models always achieve the best performance. In fact, we observe the same trend for the other GNN models, as well as GIN-VIRTUAL trained on only 10% of training data.

Overall, we find that advanced, expressive, and large GNN model gives the most promising performance on the PCQM4M-LSC dataset, although the performance still needs to be improved for practical use. We believe further advances in advanced modeling, expressive architectures, and larger model sizes could yield breakthrough in large-scale molecular property prediction—the holy-grail in graph-level ML tasks.

3. OGB-LSC Python Package

To make our datasets easily accessible, we have integrated them into the OGB Python package.[7] As shown in the Code Snippet 1, this allows our large datasets to be downloaded and processed with a few lines of code. In addition, for each dataset, we provide the evaluator object that evaluates and saves model's predictions (on the hidden test set) in a standardized way, making the test submission easy and reliable. In our website,[8] we provide the detailed tutorial of the package usage for each dataset.

Table 7. **Results of PCQM4M-LSC measured by MAE [eV].** The lower, the better. Ablation study of using only 10% of training data is also shown. Chemical accuracy indicates the final goal for practical usefulness.

Model	#Params	Validation	Test
MLP-FINGERPRINT	16.1M	0.2044	0.2068
GCN	2.0M	0.1684	0.1838
GCN-VIRTUAL	4.9M	0.1510	0.1579
GIN	3.8M	0.1536	0.1678
GIN-VIRTUAL	6.7M	**0.1396**	**0.1487**
MLP-FINGERPRINT (10% train)	6.8M	0.2708	0.2657
GIN-VIRTUAL (10% train)	6.7M	0.1790	0.1887
Chemical accuracy (goal)	–	–	0.0430

Table 8. **Model size and the MAE performance [eV].** For both models, the width indicates the hidden dimensionality. For GIN-VIRTUAL, the depth represents the number of GNN layers, while for the MLP-FINGERPRINT, the depth represents the the number of hidden layers in MLP.

Model	Width	Depth	#Params	Validation
MLP-FINGERPRINT	**1600**	**6**	16.1M	**0.2044**
	1600	**4**	11.0M	**0.2044**
	1600	2	5.8M	0.2220
	1200	6	9.7M	0.2083
GIN-VIRTUAL	**600**	**5**	6.7M	**0.1410**
	600	3	3.7M	0.1462
	300	5	1.7M	0.1442
	300	3	1.0M	0.1512

4. Conclusion

Modern applications of graph ML involve large-scale graph data with billions of edges or millions of graphs. ML advances on large graph data have been limited due to the lack of a suitable benchmark. For KDD Cup 2021, we present OGB-LSC, with the goal of advancing state-of-the-art in large-scale graph ML. OGB-LSC provides the three large-scale realistic benchmark datasets, covering the core graph ML tasks of node classification, link prediction, and graph regression. We perform dedicated baseline analysis, scaling up advanced graph models to large graphs. We show that advanced and expressive models can significantly outperform

entire result.

[7] https://github.com/snap-stanford/ogb

[8] https://ogb.stanford.edu/kddcup2021/

```
### Here we focus on MAG240M-LSC.
### Dataset object
>>> from ogb.lsc import MAG240MDatase
>>> dataset = MAG240MDataset()
# Get split
>>> dataset.get_idx_split('train')
# Get feature
>>> dataset.paper_feat

### Evaluator object
>>> from ogb.lsc import MAG240MEvaluator
>>> evaluator = MAG240MEvaluator()
# standardized evaluation
>>> evaluator.eval(...)
# saving the test prediction
>>> evaluator.save_test_submission(...)
```

. Code Snippet 1: **OGB Python package**

simpler baseline models, suggesting opportunities for further dedicated effort to yield even better performance. We hope OGB-LSC facilitates dedicated community efforts to tackle the important but challenging problem of large-scale graph ML.

Acknowledgement

We thank Michele Catasta for helpful discussion, Shigeru Maya for motivating the project, Adrijan Bradaschia for setting up the server for the project, and Amit Bleiweiss and Benjamin Braun for providing helpful feedback on our baseline code.

Stanford University is supported by DARPA under Nos. 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, JPMorgan Chase, Docomo, Hitachi, JD.com, KDDI, NVIDIA, Dell, Toshiba, Intel, and UnitedHealth Group. 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. Hongyu Ren is supported by Masason Foundation Fellowship and Apple PhD Fellowship. Jure Leskovec is a Chan Zuckerberg Biohub investigator.

Our baseline code and Python package are built on top of excellent open-source software, including NUMPY (Harris et al., 2020), PYTORCH (Paszke et al., 2017), PYTORCH GEOMETRIC (Fey & Lenssen, 2019), DGL (Wang et al., 2019), and DGL-KE (Zheng et al., 2020).

The HOKUSAI facility was used to perform some of the quantum calculations. This work was supported by the Japan Society for the Promotion of Science (JSPS KAKENHI Grant no. 18H03206). We are also grateful to Maeda Toshiyuki for helpful discussions.

References

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge. In *Special Interest Group on Management of Data (SIGMOD)*, pp. 1247–1250. AcM, 2008.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2787–2795, 2013.

Cao, W. and Xue, J. Recent progress in organic photovoltaics: Device architecture and optical design. *Energy Environ. Sci.*, 7:2123–2144, 2014. doi: 10.1039/C4EE00260A. URL http://dx.doi.org/10.1039/C4EE00260A.

Chanussot, L., Das, A., Goyal, S., Lavril, T., Shuaibi, M., Riviere, M., Tran, K., Heras-Domingo, J., Ho, C., Hu, W., Palizhati, A., Sriram, A., Wood, B., Yoon, J., Parikh, D., Zitnick, C. L., and Ulissi, Z. The Open Catalyst 2020 (oc20) dataset and community challenges. *arXiv preprint arXiv:2010.09990*, 2020.

Chen, J., Ma, T., and Xiao, C. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

Chiang, W.-L., Liu, X., Si, S., Li, Y., Bengio, S., and Hsieh, C.-J. Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 257–266, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dong, Y., Ma, H., Shen, Z., and Wang, K. A century of science: Globalization of scientific collaborations, citations, and innovations. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1437–1446. ACM, 2017.

Ferreira, L. G., Dos Santos, R. N., Oliva, G., and Andricopulo, A. D. Molecular docking and structure-based drug design strategies. *Molecules*, 20(7):13384–13421, 2015. ISSN 1420-3049. doi: 10.3390/molecules200713384.

URL https://www.mdpi.com/1420-3049/20/7/13384.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML)*, pp. 1273–1272, 2017.

Griffith, J. and Orgel, L. Ligand-field theory. *Quarterly Reviews, Chemical Society*, 11(4):381–393, 1957.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1025–1035, 2017.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Helgaker, T., Jorgensen, P., and Olsen, J. *Molecular electronic-structure theory*. John Wiley & Sons, 2014.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2020b.

Huang, Q., He, H., Singh, A., Lim, S.-N., and Benson, A. R. Combining label propagation and simple models out-performs graph neural networks. *arXiv preprint arXiv:2010.13993*, 2020.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

McPherson, M., Smith-Lovin, L., and Cook, J. M. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

Min, B., Grishman, R., Wan, L., Wang, C., and Gondek, D. Distant supervision for relation extraction with an incomplete knowledge base. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 777–782, 2013.

Morgan, H. L. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Nakata, M. the PubChemQC Project: A Large Chemical Database from the First Principle Calculations. *AIP Conf. Proc.*, 1702:090058, 2015. doi: 10.1186/1758-2946-3-4. URL http://dx.doi.org/10.1063/1.4938866.

Nakata, M. and Shimazaki, T. Pubchemqc project: A large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. 11 2019. URL https://arxiv.org/abs/1908.10084.

Reymond, J.-L. and Awale, M. Exploring chemical space for drug discovery using the chemical universe database. *ACS chemical neuroscience*, 3(9):649–657, 2012.

Rossi, E., Frasca, F., Chamberlain, B., Eynard, D., Bronstein, M., and Monti, F. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

Singhal, A. Introducing the knowledge graph: things, not strings. *Official google blog*, 5:16, 2012.

Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003, 2011.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations (ICLR)*, 2019.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, pp. 2071–2080, 2016.

Ugander, J., Karrer, B., Backstrom, L., and Marlow, C. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Vrandečić, D. and Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57 (10):78–85, 2014.

Wang, K., Shen, Z., Huang, C., Wu, C.-H., Dong, Y., and Kanakia, A. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020.

Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., Huang, Z., Guo, Q., Zhang, H., Lin, H., Zhao, J., Li, J., Smola, A. J., and Zhang, Z. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL https://arxiv.org/abs/1909.01315.

Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Wu, F., Zhang, T., Souza Jr, A. H. d., Fifty, C., Yu, T., and Weinberger, K. Q. Simplifying graph convolutional networks. In *International Conference on Machine Learning (ICML)*, 2019.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019.

Yang, Z., Cohen, W. W., and Salakhutdinov, R. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning (ICML)*, pp. 40–48, 2016.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 974–983, 2018.

Zeng, H., Zhou, H., Srivastava, A., Kannan, R., and Prasanna, V. GraphSaint: Graph sampling based inductive learning method. In *International Conference on Learning Representations (ICLR)*, 2020.

Zheng, D., Song, X., Ma, C., Tan, Z., Ye, Z., Dong, J., Xiong, H., Zhang, Z., and Karypis, G. Dgl-ke: Training knowledge graph embeddings at scale. *arXiv preprint arXiv:2004.08532*, 2020.