A Key Information about OGB-LSC

Dataset documentation. All of our datasets as well as how to use them through our Python package are documented at https://ogb.stanford.edu/kddcup2021/. Our baseline code to reproduce all the results for each dataset is available at https://github.com/snap-stanford/ogb/tree/master/examples/lsc.

Intended use. OGB-LSC is intended for machine learning and data scientists to develop ML models to tackle the challenge of large-scale graph ML.

Relevant URLs. OGB-LSC maintains the following:

- **Official website** (https://ogb.stanford.edu/kddcup2021/) is the main reference of OGB-LSC. It provides an overview of the OGB-LSC, descriptions of the datasets as well as detailed documentations of how to use the datasets through the OGB Python package. The subpage (https://ogb.stanford.edu/kddcup2021/results/) also contains the leaderboards during the KDD Cup 2021 as well as the technical reports and code provided by the winners.
- **Github repository** (https://github.com/snap-stanford/ogb) hosts the source code for the OGB Python package. OGB-LSC datasets and evaluation are all managed by the Python package. We also release all the baseline code that we used in our experiments.
- **Datasets** are extremely large (around 300GB in total) and are hosted under AWS with the help of the DGL Team. Our users do not need to directly interact with the URL, as the dataset download and processing are all managed by our Python package.
- **Mailing list** (https://groups.google.com/g/open-graph-benchmark) is used for making any announcements about OGB/OGB-LSC.

Hosting and maintenance plan. OGB-LSC's Python package is hosted and version-tracked via Github. All the datasets are hosted under the AWS with the help of the DGL Team. We design the Python package to handle downloading and processing of the datasets. OGB is a community-driven initiative that has been actively maintained by our team members.

Licensing. The OGB Python package uses the MIT license. Each dataset has its own license. Specifically, MAG240M uses ODC-BY, WikiKG90M uses CC-0, and PCQM4M uses CC BY 4.0.

Author statement. We bear all responsibility in case of violation of rights, etc., and confirmation of the data license.

Computing resources. We ran all the experiments on a server with 10 GeForce RTX 2080 GPUs and an Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz.

Limitations. Large-scale graph ML has a wide variety of application domains and there are representative graphs that we cannot cover in the current OGB-LSC datasets. Examples include large-scale recommender systems, social networks, and financial networks. These graphs are hard to obtain due to privacy and cooperative concerns, but we hope to include these realistic large graphs in the future if we have a chance. That being said, it is our hope that many methodological insights on our large graphs (training strategy, GNN architecture, regularization, etc) still transfer well to a variety of large-scale graphs. We leave the thorough investigation to future work.

Potential negative social impacts. All of our datasets are derived from practically-relevant tasks in the real world; hence, developing models and deploying them to the real-world could potentially produce predictions that are influenced by the bias in the datasets. For example, regarding the MAG240M dataset, we may use the resulting paper and author embeddings to perform a variety of downstream ML tasks such as searching for similar papers or recommending author collaboration and paper citations. Thus, it is critical to ensure there is no undesirable bias in the embeddings. There could be also misuse of highly accurate ML models. For instance, regarding the PCQM4M dataset, we

Table 8: **Basic graph statistics of the OGB-LSC datasets.** The last three graph statistics are calculated over the 'standardized' graphs, where the graphs are first converted into undirected and unlabeled homogeneous graphs with duplicated edges removed. The SNAP library (Leskovec and Sosič, 2016) is then used to compute the graph statistics. MAG240M (homo) represents the homogenized MAG240M graph with only paper nodes and citation links. Some graph statistics were omitted due to their high computational cost (the calculation did not complete in two weeks).

Model	#Graphs	Avg #nodes	Avg #edges	Avg deg	Avg clust. coeff.	Avg diameter
MAG240M	1	244,160,499	1,728,364,232	14.15	0.033	—
MAG240M (homo)	1	121,751,666	1,297,748,926	21.30	0.031	—
WikiKG90M	1	87,143,637	504,220,369	10.93	—	—
WikiKG90Mv2	1	91,230,610	601,062,811	12.59	—	—
PCQM4M	3,803,453	14.15	14.57	2.05	0.010	7.96
PCQM4Mv2	3,746,619	14.14	14.56	2.05	0.011	7.95

need to make sure that the trained molecular property predictor is used in the right way to develop useful drugs/materials rather than harmful ones.

B Basic Graph Statistics of the Datasets

The basic graph statistics of the OGB-LSC datasets are provided in Table 8.

C Details about Dataset Updates after the KDD Cup 2021

MAG240M updates. The MAG240M dataset itself has not been changed. The only update is on the test set. In Table 8, we report the test-dev accuracy of all the models.

WikiKG90Mv2 updates. The WikiKG90M dataset has been updated to WikiKG90Mv2. Below we summarize the updates we have applied to the dataset.

- **No candidate tails provided.** The most important update is that we do not provide any candidate tail entities for validation/test triples. Hence, a model needs to predict the target tail entity out of all the entities in Wikidata.
- **Created from more recent Wikidata.** The WikiKG90Mv2 is based on the public Wikidata dump downloaded at three time-stamps: May 17th, June 7th, and June 28th of 2021, for training, validation, and testing, respectively. We retain all the entities and relations in the September dump, resulting in 91,230,610 entities, 1,387 relations, and 601,062,811 triplets in total.
- **A better text encoder used.** The text features of WikiKG90Mv2 are obtained by using MPNet (Reimers and Gurevych, 2019; Song *et al.*, 2020), which is shown to be significantly better sentence encoder (Reimers and Gurevych, 2019).
- **Balancing relation types in validation/test triples.** On the new Wikidata dumps, we found the relation types of the raw validation/test triples are highly-skewed; the most frequent relation, "cites work (P2860)", occupies 60% and 85% of the entire validation and test triples, respectively. To test a model's capability to perform well across all types of relations, we subsample 15,000 triples from the entire validation/test triples such that the resulting relation counts are proportional to the cubic-root of the original relation counts.

In Table 10, we show head entities that have very sparse connection in the training KG. We see that textual features could provide important signals for predicting these triples.

We perform an extensive baseline analysis on WikiKG90Mv2. We used the same set of hyper-parameters and baseline models as our original WikiKG90M. Different from WikiKG90M, WikiKG90Mv2 does not provide any candidate tail entities. A naïvely approach is to use the entire entities as the tail candidates. However, this approach does not scale well to a KG with nearly 90M entities because we need to predict scores for all the 90M entities for every triple. Nonetheless, in practice, most of the entities are obvious negatives: *e.g.*, for the relation type "is located in", any

entities that are not locations can be easily filtered out as negatives. Based on the the above intuition, we consider the relation-specific tail candidate sets. Specifically, on training triples, we pre-compute 20K most frequent tail entities for each relation and treat them as candidate tail entities for that relation. At inference time, we use our KG model to score among those relation-specific candidates.

The results are provided in Table 10. Overall, we observe that the relative trends are similar to the original `WikiKG90Mv2`. Especially the CONCAT encoder provides the best MRR performance. Different from `WikiKG90M`, the MRR score on the new `WikiKG90Mv2` is far perfect score of 1 and leaves a lot of room for improvement. Overall, we believe it is promising to explore how to quickly generate a small number of high-quality candidate tail entities out of all the entities so that KG models only need to score a much fewer number of candidate entities.

`PCQM4Mv2` updates. The `PCQM4M` dataset has been updated to `PCQM4Mv2`. Below we summarize the updates we have applied to the dataset.

- **3D molecular structures provided.** We additionally provide 3D structures for training molecules. These structures are calculated by DFT and are obtained together with the HOMO-LUMO gap.
- **SMILES strings are partly updated.** In the process of preparing the 3D structures, we found a subtle mismatch between SMILES strings (*i.e.*, 2D molecular graphs) and the HOMO-LUMO gap for about 10% of the entire molecules. Specifically, the SMILES strings can be changed in the course of DFT's geometry optimization, but in `PCQM4M`, we provided the *initial* SMILES strings. In the updated `PCQM4Mv2`, we provide SMILES strings corresponding to the *final* optimized 3D structures. Note that the HOMO-LUMO gap was calculated by DFT based on the final 3D structures (Nakata and Shimazaki, 2017); hence, it makes more sense to correspond the HOMO-LUMO gap with the SMILES string associated with the final 3D structures.
- **Number of molecules decreased slightly.** As a result of the SMILES update, some molecules can no longer be parsed by the commonly-used chemistry toolkit, *i.e.*, `rdkit` (Landrum *et al.*, 2006). As a result, the total number of molecules has been slightly reduced to 3,746,619.
- **Split ratio changed.** For `PCQM4Mv2`, we set the split ratio for train/validation/test-dev/test-challenge to 90/2/4/4. The split is still done by PubChem compound ID so that there is no test label leakage, *i.e.*, all the test molecules in `PCQM4Mv2` is in the test split of `PCQM4M`.

Similar to `PCQM4M`, we also provide our baseline analysis on the updated `PCQM4Mv2` dataset. At inference time, we clamped the output values to be between 0 and 20, which prevents our models from predicting erroneous values for some test molecules. We show the results in Tables 12 and 13. We found that all the models were able to achieve lower MAE compared to `PCQM4M`, probably because we have fixed the mismatch bug described above. Beyond the overall better MAE, we see that the trend in model performance is mostly preserved; larger and more expressive GNN models achieve better results. For the GNNs, we observe that the depth helps more than width. Interestingly, too-wide models often make unstable prediction on validation molecules.

Table 9: **Results of `MAG240M` measured by the accuracy (%).** R-GRAPHSAGE/-GAT utilize the full heterogeneous graph information, while the other models operate on the homogeneous paper citation graph. Test accuracy is evaluated on the *test-dev set.*

Model	#Params	Validation	Test-dev
MLP	0.5M	52.67	52.76
LABELPROP	0	58.44	56.38
SGC	0.7M	65.82	65.30
SIGN	3.8M	66.64	66.03
MLP+C&S	0.5M	66.98	66.05
GRAPHSAGE (NS)	4.9M	66.79	66.21
GAT (NS)	4.9M	67.15	66.71
R-GRAPHSAGE (NS)	12.2M	69.86	68.78
R-GAT (NS)	12.3M	**70.02**	**69.31**
KDD 1ST: BD-PGL	Ensemble		**75.39**
KDD 2ND: ACADEMIC	Ensemble		**75.07**
KDD 3RD: SYNERISE AI	Ensemble		**74.57**

Table 10: **Textual representation of validation triplets whose head entities only *appear once* as head in the training `WikiKG90Mv2`.**

Head	Relation	Tail
Herbert Hoover's Inaugural Address	country	United States of America
Jussi Award for Best Sound Recording	instance of	class of award
organ dose	calculated from	absorbed dose
British Endurance Racing Team	country	United Kingdom
Knee bursae	anatomical location	knee
Churches in Dekanat Leuchtenberg	is a list of	church building
web content management system	model item	workflow management system
Stephan von Divonne	given name	Stephan
Minecraft mod	depends on software	Minecraft
beer pouring	uses	beer engine

Table 11: **Results of `WikiKG90Mv2` measured by the Mean Reciprocal Rank (MRR).**

Model	#Params	Validation	Test-dev
TRANSE-SHALLOW	18.2B	0.1103	0.0824
COMPLEX-SHALLOW	18.2B	0.1150	0.0985
TRANSE-MPNET	0.3M	0.1128	0.0860
COMPLEX-MPNET	0.3M	0.1258	0.0988
TRANSE-CONCAT	18.2B	**0.2060**	**0.1761**
COMPLEX-CONCAT	18.2B	**0.2048**	**0.1761**

Table 12: **Results of `PCQM4Mv2` 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-dev
MLP-FINGERPRINT	16.1M	0.1753	0.1760
GCN	2.0M	0.1379	0.1398
GCN-VIRTUAL	4.9M	0.1153	0.1152
GIN	3.8M	0.1195	0.1218
GIN-VIRTUAL	6.7M	**0.1083**	**0.1084**
MLP-FINGERPRINT (10% train)	16.1M	0.2429	0.2445
GIN-VIRTUAL (10% train)	6.7M	0.1442	0.1446
Chemical accuracy (goal)	–	0.0430	

Table 13: **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	0.1753
	1600	**4**	11.0M	0.1752
	1600	2	5.8M	0.1954
	1200	6	9.7M	0.1804
GIN-VIRTUAL	**600**	**5**	6.7M	0.1083
	600	3	3.7M	0.1239
	300	5	1.7M	0.1100
	300	3	1.0M	0.1181