On Graph Neural Network Ensembles for Large-Scale Molecular Property Prediction

Edward Elson Kosasih[1,6][0000−0001−5293−2641], Joaquin Cabezas[3,6][0000−0001−7041−5082], Xavier Sumba[4,6][0000−0002−4475−079X], Piotr Bielak[2,6][0000−0002−1487−2569], Kamil Tagowski[2,6][0000−0003−4809−3587], Kelvin Idanwekhai[5,6][0000−0002−3804−8830], Benedict Aaron Tjandra[1,6], and Arian Rokkum Jamasb[1,6][0000−0002−6727−7579]

[1] University of Cambridge, United Kingdom
[2] Wrocław University of Science and Technology, Wrocław, Poland
[3] Universitat Rovira i Virgili, Tarragona, Spain
[4] McGill University, Canada
[5] Obafemi Awolowo University, Nigeria
[6] ML Collective

Abstract. In order to advance large-scale graph machine learning, the Open Graph Benchmark Large Scale Challenge (OGB-LSC) [8] was proposed at the KDD Cup 2021. The PCQM4M-LSC dataset defines a molecular HOMO-LUMO property prediction task on about 3.8M graphs. In this short paper, we show our current work-in-progress solution which builds an ensemble of three graph neural networks models based on GIN, Bayesian Neural Networks and DiffPool. Our approach outperforms the provided baseline by 7.6%. Moreover, using uncertainty in our ensemble's prediction, we can identify molecules whose HOMO-LUMO gaps are harder to predict (with Pearson's correlation of 0.5181). We anticipate that this will facilitate active learning.

Keywords: Graph Neural Networks · Ensembling · HOMO-LUMO gap.

1 Introduction

Graph-structured data can be found in many application areas, such as social networks, molecules, protein-protein interaction networks or knowledge graphs, and are of particular interest in drug discovery [5].

Despite the success of graph machine learning methods, one should note that the vast majority of papers report results only for small synthetic datasets, whereas real-world datasets can contain several million or even billions of nodes [8]. The most popular approach currently – Graph Neural Networks (GNNs) – are based on a message passing algorithm, where each node aggregates messages from its direct neighbors. However, there is a trade-off between expressivity and scalability. Therefore, applying GNNs to real-world data is a challenging task.

To address this problem, the Open Graph Benchmark Large-Scale Challenge [9] was proposed for the KDD Cup at the KDD 2021 conference. It aims to facilitate building models that can process large-scale graphs while maintaining performance by providing three large graph datasets that encompass node, edge, and graph-level prediction tasks.

In this work, we focus on the PCQM4M-LSC quantum chemistry dataset which defines a graph regression task aiming at predicting the HOMO-LUMO gap for a given molecule. We propose an ensemble method for molecular graphs that relies on GNNs as weak learners. We demonstrate improved predictive performance that outperforms individual GNN models and other baselines by 7.6%. We additionally perform an uncertainty analysis which shows a correlation (Pearson's R: 0.5181) between error and uncertainty.

2 Background

HOMO-LUMO gap is the energy difference between the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO). Although, being able to predict reactivity, photo-chemical or photo-physical molecule properties, the HOMO-LUMO gap cannot be determined experimentally as it is not an actual physical quantity. Traditionally this value is calculated using the Density Functional Theory (DFT) quantum mechanical calculations [10], which can take from hours to days [13]. This can exponentially slow down computational chemistry research on large molecule sets, limiting the ability to study applications in drug discovery, material science and photo-voltaic design.

PCQM4M-LSC is a quantum chemistry dataset [8], curated under the PubChemQC project [14]. It consists of 3,803,453 molecules and total of 55,399,880 edges. Each molecule is provided in the SMILES string format, which is further converted to a graph, along with their ground truth HOMO-LUMO gap. All molecules are identified by their PubChem ID (CID) and split into 80% training, 10% validation and 10% test based on their 2D structural frameworks, ensuring that molecules in validation and test set are structurally different from training data. The regression performance is measured by the Mean Absolute Error (MAE).

OGB-LSC is a KDD 2021 Cup challenge. The task is to use machine learning (ML) technique on the PCQM4M-LSC dataset to predict HOMO-LUMO gaps in cheaper computational time compared to the traditional expensive DFT approach. In order for ML to be on par with DFT, it has to achieve an MAE of 0.0430.

3 Proposed solution

We experiment with three concepts: Bayesian networks, hierarchical clustering (graph pooling) and ensembling. All our solutions take into account the large scale of the dataset and the computational power required, impacting decisions regarding embedding size and depth.

3.1 Weak learners

GIN-Virtual This is a baseline model that is provided as part of OGB-LSC. Graph Isomorphism Network (GIN) [16] is theoretically proven to achieve maximum discriminative power among GNNs, as it generalizes the Weisfeiler-Lehman (WL) graph isomorphism test. We combine the GIN model with the concept of a virtual node, which is connected to every node in the graph with a special edge type. Such a setting allows information to travel long distances during the propagation phase instead of being limited to a certain number of hops within the graph. The latent dimensionality is set to 600, the number of GNN layers is set to 5. We follow the rest of the configuration provided in [9]. The model output is clamped between 0 and 50, following physical constraints to HOMO-LUMO gaps. The L1 loss is used for training. We call this implementation *GIN-Virtual*.

Table 1: MAE on validation and test splits. Only baseline and ensemble-all test values are known since only this result is released by the organisers. The MAE for each ensemble corresponds to the red bars in Figure 1a. As seen, *ensemble-all* outperforms all other models.

	Model	#Params	Validation MAE	Test MAE
Baselines [8]	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**
Ours	GIN-Virtual-ensemble	6.7M	0.1335	n/a
	GIN-Virtual-BNN-ensemble	6.8M	0.1329	n/a
	GIN-Virtual-DiffPool-ensemble	7.7M	0.1346	n/a
	ensemble-all	21.2M	**0.1290**	**0.1358**

GIN-Virtual with BNN layers The usage of Bayesian Neural Networks (BNNs) both for message passing and the GNN readout stage improves predictive accuracy on the QM9 dataset [12]. Particularly, Bayes by Backpropagation (BBP) [1] outperforms other BNN approaches.

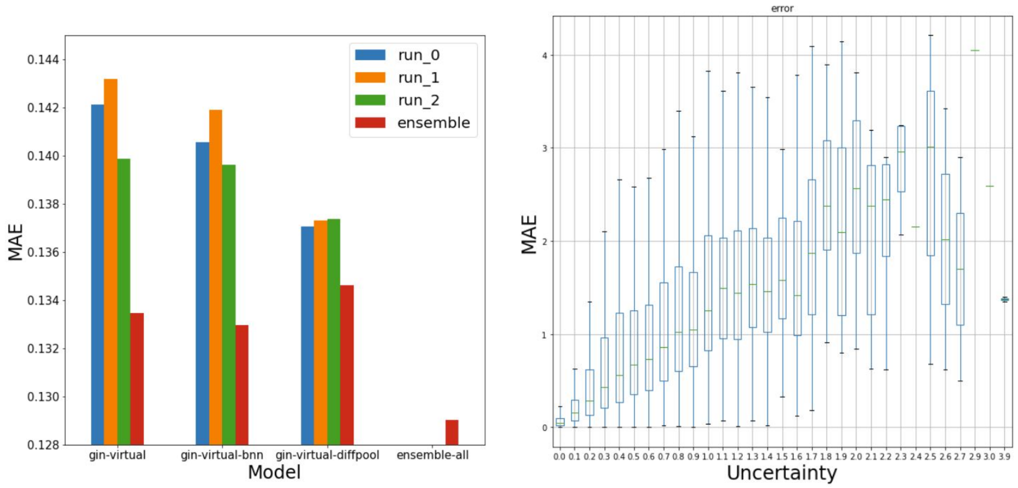
In this work, we implement BBP at the readout stage using torchbnn [7]. We replace the single layer neural network from the baseline GIN-Virtual with 4 Bayesian Linear layers. During training we utilize the Kullback-Leibler Divergence in addition to the L1 loss. We call this implementation *GIN-Virtual-BNN*.

GIN-Virtual with DiffPool Local pooling operators (known from Convolutional Neural Networks) are used to coarsen intermediate layer outputs. Similar approaches can be applied for GNNs, coarsening the graph into a smaller set of supernodes. Multiscale coarsening of graph data is a crucial component of scale separation priors needed as part of a geometric deep learning architecture [2].

In this work, we implement a recently proposed coarsening technique called DiffPool [17]. We add a DiffPool layer after GIN message passing layers to coarsen the graph into 5 nodes. Afterwards, we run an additional GIN layer on the coarsened graph. Finally, we concatenate both graph embeddings from the original and coarsened graph, and feed them into the readout layer. We call this implementation *GIN-Virtual-DiffPool*.

3.2 GNN ensemble

Ensembling is a technique to combine prediction outputs of several machine learning models, either with different architecture or similar, but trained from different initial conditions. In [3] the authors explain three possible reasons why ensembling often improves the performance compared to any single weak learner. An ensemble: (1) reduces the risk of choosing the wrong model, (2) provides better approximation as a result of training from different initial conditions and (3) forms a weighted sum representation that might not be achievable by any single weak learner.



(a) Ensembling significantly improves performance compared to individual weak learners. Each model is trained three times from different initial conditions (*run_{0,1,2}*).

(b) Prediction error tends to increase as uncertainty increases. Explain this is obtained from calculating uncertainty across the 9 different trials.

Fig. 1: Ensemble MAE results and uncertainty.

In this work, we perform two types of ensembling across the three aforementioned models. First, we train each individual model (weak learner) three times, and then take the mean of their predictions as the ensemble score. We call these, *GIN-Virtual-ensemble*, *GIN-Virtual-BNN-ensemble* and *GIN-Virtual-DiffPool-ensemble* respectively. Second, we combine and calculate the mean of results across all models and trials. We call this *ensemble-all*.

4 Experimental results

Both *GIN-Virtual* and *GIN-Virtual-BNN* are implemented in PyTorch-Geometric [4], while *GIN-Virtual-DiffPool* is implemented with DGL [15]. We run our experiments on a Google Cloud Platform Compute Engine machine of type n1-high-mem-16 with 16 vCPUs, 104 GB memory and 1 NVIDIA Tesla T4 GPU. Our code implementation can be found in our github[7].

[7] https://github.com/cuent/ogb-kdd21

4.1 Baseline performance

The PCQM4M-LSC dataset comes with a set of benchmark Graph Neural Networks models [8]. Graph Convolutional Network (GCN) [11] and GIN perform best in this task. Moreover, adding a virtual node [6] further improves results. Table 1 (upper part) shows the performance on both validation and test splits. GIN with addition of the virtual node provides the best performance.

4.2 Ensemble performance

All models are trained on the 3 million molecules from the training dataset and evaluated on 300,000 validation and another 300,000 test molecules. The labels in the test set are kept by the organisers of the challenge. Hence, we report full results only for the validation set. For the test set, we have only values for the baselines and our submission – *ensemble-all*. We rank 21st out of the top 49 results shown in the leaderboard[8].

Figure 1a and Table 1 (bottom part) show that ensembling of individual types of weak learners outperforms any one of them. Moreover, taking an ensemble across all models and trials brought the MAE down to 0.1290. This is 7.6% better than the reported baseline performance of 0.1396 [8].

4.3 Uncertainty Analysis

We also analyse whether there is any correlation between disagreement among weak learners and the task performance accuracy. We use such disagreement as a proxy for measure of uncertainty. This information, if true, would be useful to identify molecular graphs that are harder to model and predict. We anticipate that such uncertainty value could be used for future active learning tasks.

We measure uncertainty as the standard deviation of the weak learners' predictions and find that there is indeed a tendency for performance error to increase as ensembling uncertainty goes up (see Figure 1b). The Pearson's correlation of this trend is 0.5181.

5 Conclusion

While graph machine learning has been extensively studied recently, a vast majority of papers have only reported results for small datasets [8]. To encourage development of more scalable models, the OGB-LSC challenge has been released during the KDD Cup at KDD 2021 Conference. We addressed the PCQM4M-LSC molecular property prediction task and proposed a GNN ensemble model that outperforms the provided baseline by 7.6%. Additionally, we found that our model's uncertainty informs the error in prediction with a Pearson's correlation of 0.5181. We anticipate that this information would be useful for future active learning framework.

6 Credit

We would like to thank ML Collective for supporting and funding our research.

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

References

1. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight Uncertainty in Neural Networks. arXiv:1505.05424 [cs, stat] (May 2015), http://arxiv.org/abs/1505.05424, arXiv: 1505.05424
2. Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P.: Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. arXiv:2104.13478 [cs, stat] (Apr 2021), http://arxiv.org/abs/2104.13478, arXiv: 2104.13478
3. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Proceedings of the First International Workshop on Multiple Classifier Systems. pp. 1–15. MCS '00, Springer-Verlag, Berlin, Heidelberg (Jun 2000)
4. Fey, M., Lenssen, J.E.: Fast Graph Representation Learning with PyTorch Geometric. arXiv:1903.02428 [cs, stat] (Apr 2019), http://arxiv.org/abs/1903.02428, arXiv: 1903.02428
5. Gaudelet, T., Day, B., Jamasb, A.R., Soman, J., Regep, C., Liu, G., Hayter, J.B.R., Vickers, R., Roberts, C., Tang, J., Roblin, D., Blundell, T.L., Bronstein, M.M., Taylor-King, J.P.: Utilizing graph machine learning within drug discovery and development. Briefings in Bioinformatics (05 2021). https://doi.org/10.1093/bib/bbab159, https://doi.org/10.1093/bib/bbab159, bbab159
6. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1263–1272. PMLR (06–11 Aug 2017), http://proceedings.mlr.press/v70/gilmer17a.html
7. Harry: Harry24k/bayesian-neural-network-pytorch (Jun 2021), https://github.com/Harry24k/bayesian-neural-network-pytorch, original-date: 2019-09-04T14:28:12Z
8. Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., Leskovec, J.: OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs. arXiv:2103.09430 [cs] (Mar 2021), http://arxiv.org/abs/2103.09430, arXiv: 2103.09430
9. Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., Leskovec, J.: Ogb-lsc: A large-scale challenge for machine learning on graphs. arXiv preprint arXiv:2103.09430 (2021)
10. Huang, Y., Rong, C., Zhang, R., Liu, S.: Evaluating frontier orbital energy and homo/lumo gap with descriptors from density functional reactivity theory. Journal of Molecular Modeling **23**(1), 3 (Dec 2016). https://doi.org/10.1007/s00894-016-3175-x, https://doi.org/10.1007/s00894-016-3175-x
11. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: Proceedings of the 5th International Conference on Learning Representations. ICLR '17 (2017), https://openreview.net/forum?id=SJU4ayYgl
12. Lamb, G., Paige, B.: Bayesian Graph Neural Networks for Molecular Property Prediction. arXiv:2012.02089 [cs, q-bio] (Nov 2020), http://arxiv.org/abs/2012.02089, arXiv: 2012.02089
13. Morgante, P., Peverati, R.: The devil in the details: What everybody should know when running dft calculations (Nov 2019). https://doi.org/10.26434/chemrxiv.10187756.v1, 10.26434/chemrxiv.10187756.v1
14. Nakata, M., 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). https://doi.org/10.1021/acs.jcim.7b00083, https://doi.org/10.1021/acs.jcim.7b00083, pMID: 28481528
15. Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., Zhang, Z.: Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. arXiv:1909.01315 [cs, stat] (Aug 2020), http://arxiv.org/abs/1909.01315, arXiv: 1909.01315

16. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=ryGs6iA5Km

17. Ying, R., You, J., Morris, C., Ren, X., Hamilton, W.L., Leskovec, J.: Hierarchical Graph Representation Learning with Differentiable Pooling. arXiv:1806.08804 [cs, stat] (Feb 2019), http://arxiv.org/abs/1806.08804, arXiv: 1806.08804