Fast Quantum Property Prediction via Deeper 2D and 3D Graph Networks

Meng Liu[*, 1], **Cong Fu**[*, 1], **Xuan Zhang**[1], **Limei Wang**[1], **Yaochen Xie**[1], **Hao Yuan**[1], **Youzhi Luo**[1], **Zhao Xu**[1], **Shenglong Xu**[2], and **Shuiwang Ji**[1]

[*]These authors contributed equally to this work.
[1]Department of Computer Science and Engineering, Texas A&M University
[2]Department of Physics and Astronomy, Texas A&M University
{mengliu,congfu,xuan.zhang,limei,ethanycx,hao.yuan,yzluo,zhaoxu,slxu,sji}@tamu.edu

Abstract

Molecular property prediction is gaining increasing attention due to its diverse applications. One task of particular interests and importance is to predict quantum chemical properties without 3D equilibrium structures. This is practically favorable since obtaining 3D equilibrium structures requires extremely expensive calculations. In this work, we design a deep graph neural network to predict quantum properties by directly learning from 2D molecular graphs. In addition, we propose a 3D graph neural network to learn from low-cost conformer sets, which can be obtained with open-source tools using an affordable budget. We employ our methods to participate in the 2021 KDD Cup on OGB Large-Scale Challenge (OGB-LSC), which aims to predict the HOMO-LUMO energy gap of molecules. Final evaluation results reveal that we are one of the winners with a mean absolute error of 0.1235 on the holdout test set. Our implementation is available as part of the MoleculeX package (`https://github.com/divelab/MoleculeX`).

1 Introduction

Molecular property prediction is of great importance in many applications, such as chemistry, drug discovery, and material science. Many molecular properties, such as the energy and the shape of molecules, could be computed by quantum mechanical simulation methods, such as Density Functional Theory (DFT). However, such methods are computationally expensive. To be specific, it takes several hours to run the DFT computation for a small molecule since it requires geometry optimization to obtain 3D equilibrium structures. Therefore, it is highly desired if we could predict such quantum chemical properties without 3D equilibrium structures of molecules.

Recently, graph deep learning methods have been developed for molecular property prediction [4, 25, 27, 22, 24]. As molecules can be naturally treated as graphs by viewing atoms as nodes and bonds as edges, these methods leverage various graph neural networks (GNNs) [2, 4, 1, 12, 23, 26] to learn from 2D molecular graphs and achieve great success. Nevertheless, these approaches only employ shallow graph neural networks, thus limiting the expressive power and the receptive fields of the models. In addition, the existing methods only focus on 2D molecular graphs without explicitly considering 3D information, which is crucial for determining quantum chemical properties. Hence, it remains challenging to incorporate useful 3D information with an affordable budget since 3D equilibrium structures are usually unavailable and time-consuming to obtain. While there is another line of research that predicts quantum chemical properties given 3D equilibrium structures [21, 13, 19, 9], it is orthogonal to our work since 3D equilibrium structures are not available in this work.

NeurIPS 2021 AI for Science Workshop.

Motivated by the above two challenges, we propose a deeper 2D graph neural network on 2D molecular graphs and a 3D graph neural network on low-cost conformer sets to predict quantum chemical properties. Specifically, we leverage the recent advanced deep graph neural networks, namely DeeperGCN [16] and DAGNN [18], to construct our 2D model. For the 3D model, we obtain the conformer sets using RDKit [15], which takes less than 0.05 seconds for a molecule on average. Afterwards, we deploy a 3D GNN to learn from the conformer sets. Our intuition is to utilize the imprecise but descriptive 3D information contained in the conformer sets to improve the prediction performance for quantum chemical properties. To show the effectiveness of our proposed approach, we conduct experiments on PCQM4M-LSC dataset [7], included in the 2021 KDD Cup on OGB Large-Scale Challenge[1], to predict the HOMO-LUMO energy gap of molecules without given 3D equilibrium structures. The results show that our methods achieve remarkable prediction performance.

2 Methodology

We consider a molecular graph $G = (V, E, \mathbf{X}, \mathbf{E}, \mathbf{A})$, where node set $V = \{1, 2, \ldots, n\}$ denotes atoms and edges $E \subseteq V \times V$ are given by bonds or by connecting atoms considering certain cutoff distance. Without losing generality, we consider each node and edge is associated with a feature vector. To be specific, $\mathbf{X} \in \mathbb{R}^{|V| \times d}$ is the node feature matrix, where each row $\mathbf{x}_i \in \mathbb{R}^d$ represents the d-dimensional feature vector of atom i. $\mathbf{E} \in \mathbb{R}^{|E| \times p}$ denotes the edge feature matrix and each row is a p-dimensional feature vector of an edge. We use $\mathbf{e}_{ij} \in \mathbb{R}^p$ to denote the feature vector of the edge from atom i to j. The connectivity of the graph is described by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$.

2.1 Deeper 2D Model on 2D Molecular Graphs

We aim to construct a deep graph neural network, which can bring us large receptive fields and powerful expressivity. Our model is developed based on the recent progress on deep GNNs, particularly, DeeperGCN [16] and DAGNN [18]. The details of the main components are described below.

GENConv. Most modern GNNs follow a message passing scheme [4, 1]. To be specific, we iteratively update the node representation by aggregating information, *a.k.a.*, message, from neighboring nodes. GENConv [16] also follows such a mechanism. The key characteristic of GENConv is the *SoftMax_Agg* aggregation function. The forward computation of one GENConv layer for each node i is formulated as

$$\begin{aligned} \mathbf{m}_{ji} &= \text{ReLU}(\mathbf{x}_j + \mathbf{e}_{ji}) + \epsilon, \ j \in \mathcal{N}_i \\ \mathbf{x}_i' &= \text{MLP}\left(\mathbf{x}_i + \textit{SoftMax\_Agg}_\beta\left(\{\mathbf{m}_{ji} : j \in \mathcal{N}_i\}\right)\right) \\ \textit{SoftMax\_Agg}_\beta(\ ) &= \sum_{j \in \mathcal{N}_i} \frac{\exp\left(\beta \mathbf{m}_{ji}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\beta \mathbf{m}_{ki}\right)} \ \mathbf{m}_{ji}, \end{aligned} \tag{1}$$

where $\mathbf{m}_{ji}$ represents the message from atom j to atom i. ϵ is a small positive constant. ReLU( ) [14] is the activation function and MLP( ) denotes a multi-layer perceptron. Note that β is a learnable scalar parameter and initialized as 1. As shown in the above equations, in GENConv, we firstly construct messages from neighboring nodes and then aggregate them to obtain the updated representation $\mathbf{x}_i'$ for each node i.

DeeperGCN Layer. Based on GENConv block, DeeperGCN Layer [16] is proposed by further introducing skip connections [5] and the pre-activation technique [6]. Therefore, the resulting DeeperGCN Layer consists of the following components: Normalization $\rightarrow$ Activation $\rightarrow$ GENConv $\rightarrow$ Addition. We apply BatchNorm (BN) [10] as the normalization and ReLU as the activation function.

Virtual Node. Virtual node is found to be effective across various graph-level tasks [4, 8]. Generally, we can augment a graph with a virtual node that communicates with all other nodes, thus better capturing the global or long-range information. The feature of the virtual node is denoted as $\mathbf{g}$ and initialized as a zero-values vector. In each layer, the virtual node feature is updated as

$$\mathbf{g}' = \text{MLP}\left(\text{Readout}_{\text{sum}}\left(\{\mathbf{x}_i, i \in V\}\right) + \mathbf{g}\right), \tag{2}$$

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

where $\text{Readout}_{\text{sum}}(\ )$ represents the summation readout function.

DAGNN. We apply L DeeperGCN Layers and incorporate the virtual node technique to learn the node representations. We denote the resulting node representation matrix as $\mathbf{Z} \in \mathbb{R}^{|V| \times f}$, where f is the number of hidden dimensions. As shown by [17], the performance could be further improved if we deploy a diffusion algorithm on the resulting node representations $\mathbf{Z}$. In our work, we choose DAGNN [18], which is demonstrated to be effective and can adaptively balance the information from various receptive fields for each node. In the original DAGNN, there are three main steps, including transformation, propagation, and adaptive adjustment. An MLP is used for transformation in the orginal DAGNN. In this work, we remove this transformation step since we can regard the previous DeeperGCN Layers as the transformation. Hence, we deploy K steps of propagation and the adaptive adjustment mechanism to integrate the information of different receptive fields.

Afterwards, we apply a summation readout function to derive a graph-level representation, and then use a linear transformation $\mathbf{W} \in \mathbb{R}^{f \times 1}$ to make prediction for the desired molecular quantum property. Formally, it can be written as

$$\hat{y} = \text{Readout}_{\text{sum}}(\{\mathbf{y}_i, i \in V\}) \mathbf{W} \quad \in \mathbb{R}. \tag{3}$$

In our 2D model, we have L DeeperGCN Layers and K steps of propagation in DAGNN, thereby making the depth, or receptive field, to be $(L + K)$. The overall architecture of our deeper 2D model is illustrated in Figure 2, Appendix A.

2.2 3D Model on Low-Cost Conformer Sets

Although it is natural to view a molecule as a 2D graph defined by atoms and bonds in chemistry, quantum mechanical simulation methods still require the precise 3D structure of a molecule. In particular, the Hamiltonian and the wave functions defining the state of a quantum system are functions of 3D atomic coordinates and are sensitive to small perturbations. While the 3D equilibrium structure of a molecule, from which we compute the quantum properties, is very expensive to obtain, 3D atomic coordinates that are close to equilibrium can be efficiently generated with open-source software [3]. However, we empirically observe that using only one randomly sampled conformer as input leads to unexpected prediction results. Thus, we propose to sample multiple conformers for each molecule and leverage a 3D GNN to learn from the set of conformers. Intuitively, the idea is consistent with the fact that molecules are intrinsically flexible objects with fluctuating 3D structures. Therefore, learning from a set of conformers can reduce the variance in the input space.

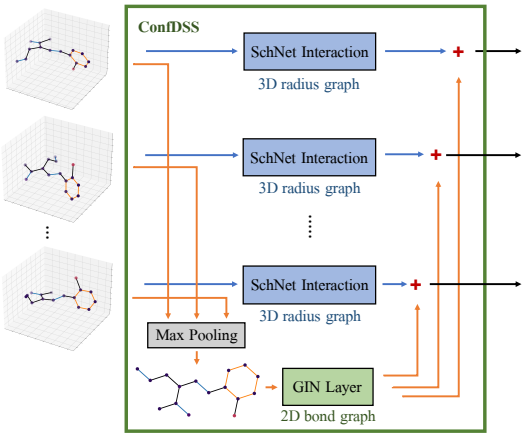

Figure 1: An illustration of the proposed ConfDSS Layer. Details are described in Section 2.2.

Conformer Generation. We generate mutliple conformers for each molecule with RDKit [15] and prune similar conformers with an RMSD cutoff $R = 0.5$Å.

ConfDSS Layer. Inspired by [20], we propose the ConfDSS Layer, as illustrated in Figure 1, to learn from generated conformer sets. Each conformer can be regarded as a set of nodes where each node is associated with a feature vector and a 3D coordinate. Both the input and the output of the ConfDSS Layer are a set of such conformers. Generally, in each ConfDSS Layer, we apply 3D GNN blocks to learn from individual conformers. In addition, as suggested by [20], a 2D GNN block is deployed to their aggregated graph.

For individual conformers, in order to capture the 3D information, we do not use the graph topology information given by bonds. Instead, we construct a spatial graph where edges are defined between atom pairs closer than a cutoff distance. Individual conformers are then processed by SchNet interaction blocks [21], which can consider the 3D information contained in the conformers. Skip connections are applied after the SchNet blocks. The parameters of the SchNet interaction blocks are shared across conformers.

Note that the topology information and edge features of the aggregated graph are still defined by bonds. The node features of the aggregated graph are obtained by performing set-level max pooling over the node features of the conformer set. The aggregated graph has the same number of nodes as each conformer. A GIN layer [26] is then applied to the aggregated graph. Afterwards, we add the node features from the aggregated graph back to the node features of each output conformers to explicitly incorporate the set-level information into each conformer. Similar to our 2D model, we also incorporate the virtual node technique in the ConfDSS layer.

3D Model. As shown in Figure 3, Appendix B, the final 3D model is composed of 5 ConfDSS Layers follwed by the ReLU activation. For the input of the first ConfDSS layer, all conformers share the same set of node features and become different in latter layers due to their different 3D coordinates. The final output conformer set is aggregated into a single feature vector representing the input molecule by performing conformer-level readout and set-level max pooling. Finally, we apply an MLP to this feature vector for prediction.

3 Experiments

Dataset and setup. We evaluate our methods on PCQM4M-LSC [7], which is collected for graph-level quantum chemical property prediction, specifically, HOMO-LUMO energy gap prediction without given 3D equilibrium structures. It contains over 3.8M molecules in total. The training/validation/test split of PCQM4M-LSC is $80\%/10\%/10\%$ and the labels of the test set are not publicly available. For comparing with baselines, we train our model on the given training set and compare the performance on the standard validation set, given that the results are shown to be consistent across the validation set and the hidden test set [7]. The implementation details are described in Appendix C.

Table 1: Results on PCQM4M-LSC in terms of MAE [eV] on the validation set. The results of baselines are obtained from [7]. * represents that the MAE is computed on the molecules that have conformers generated by RDKit. There are around 0.1% molecules in the original validation set, for which RDKit cannot generate conformers. The result is still convincing since the proportion of missing data is negligible.

Method	#Params	Validation
GCN	2.0M	0.1684
GCN-Virtual	4.9M	0.1510
GIN	3.8M	0.1536
GIN-Virtual	6.7M	0.1396
Our 2D model	34.1M	**0.1278**
Our 3D model	9.0M	**0.1295**
Our 2D model without DAGNN	34.1M	0.1350

Baselines. Following [7], we consider GCN [12], GIN [26], GCN-Virtual, and GIN-Virtual as baselines. The last two models denotes the variants with adding the virtual node.

Results. The results obtained by training on the given training set and then evaluating on the standard validation set are shown in Table 1. Both our 2D and 3D models outperform baselines with obvious margins. These demonstrate the effectiveness of developing deep and large models and incorporating low-cost descriptive 3D information for large-scale quantum property prediction. We also conduct an ablation study to investigate the improvement of including DAGNN. We construct a model by removing DAGNN from our 2D model. As shown in Table 1, we observe that the performance degrades a lot without DAGNN. This shows the surprising improvement of incorporating DAGNN, considering that there are only f, *i.e.*, 600 in our setting, learnable parameters in this DAGNN component.

To participate in the 2021 KDD Cup on OGB large-Scale Challenge, we submit the ensemble results of multiple models trained on our new splits. The details are included in Appendix D.

4 Conclusion

To predict molecular quantum chemical properties without known 3D equilibrium structures, in this work, we propose a deeper 2D GNN to endow us larger receptive fields and more expressivity on 2D molecular graphs. In addition, to explicitly consider 3D information, we propose a 3D GNN to learn from low-cost conformer sets, which can be obtained in an affordable budget. We perform experiments on the 2021 KDD Cup on OGB Large-Scale Challenge, in which the task is to predict the HOMO-LUMO energy gap of molecules. The results demonstrate that our proposed 2D and 3D methods are effective. We also repartition the available data and adopt ensemble strategy to further improve the prediction performance, which is shown to be helpful according to our experiments.

Acknowledgements

This work was supported in part by National Science Foundation grants IIS-2006861, IIS-1955189, IIS-1908198, DBI-1922969, and National Institutes of Health grant 1R21NS102828.

References

[1] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[2] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[3] Jean-Paul Ebejer, Garrett M Morris, and Charlotte M Deane. Freely available conformer generation methods: how good are they? *Journal of chemical information and modeling*, 52(5):1146–1158, 2012.

[4] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th international conference on machine learning*, pages 1263–1272, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[7] Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

[8] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, 2020.

[9] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C Lawrence Zitnick. Forcenet: A graph neural network for large-scale quantum calculations. *arXiv preprint arXiv:2103.01436*, 2021.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[13] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[15] Greg Landrum et al. RDKit: Open-source cheminformatics. 2006.

[16] Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 2020.

[17] Weibin Li, Shanzhuo Zhang, Lihang Liu, Zhengjie Huang, Jieqiong Lei, Xiaomin Fang, Shikun Feng, and Fan Wang. Molecule representation learning by leveraging chemical information. *Technical Report*, 2021.

[18] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 338–348, 2020.

[19] Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021.

[20] Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. In *International Conference on Machine Learning*, pages 6734–6744. PMLR, 2020.

[21] KT Schütt, P-J Kindermans, Huziel E Sauceda, S Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 992–1002. Neural Information Processing Systems (NIPS) Foundation, 2018.

[22] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackerman, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representation*, 2018.

[24] Zhengyang Wang, Meng Liu, Youzhi Luo, Zhao Xu, Yaochen Xie, Limei Wang, Lei Cai, and Shuiwang Ji. Advanced graph and sequence neural networks for molecular property prediction and drug discovery. *arXiv preprint arXiv:2012.01981*, 2020.

[25] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[26] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[27] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

Appendix

A. An Illustration of Our 2D Model

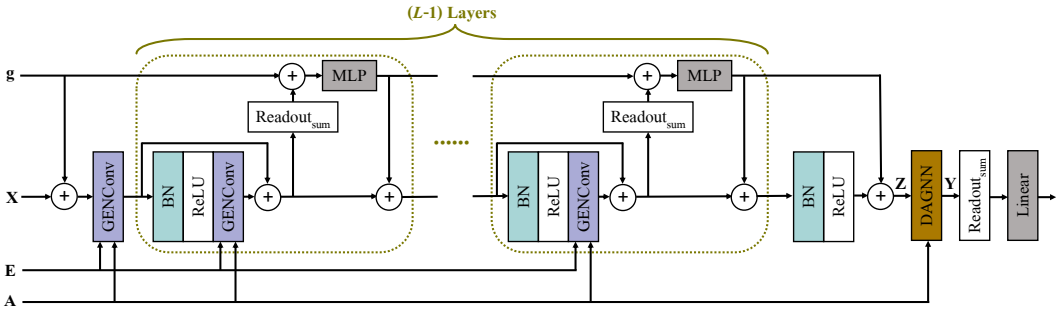

Figure 2: An illustration of our deeper 2D model. Details are described in Section 2.1.

B. An Illustration of Our 3D Model

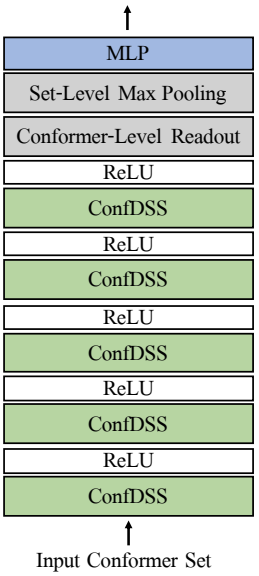

Figure 3: An illustration of our 3D model. Details are described in Section 2.2.

C. Implementation Details

We follow OGB [8, 7] to initialize the node features and edge features for molecular graphs. To be specific, each node is associated with a 9-dimensional feature, including atom type, chirality, *etc.*. Each edge has a 3-dimensional feature, containing bond type, stereochemistry, and conjugation. For our 2D model, we adopt $L = 16$ DeeperGCN Layers and set $K = 5$ in DAGNN. We set the number of hidden dimensions to be 600. We apply dropout to the MLP in Eq. (2) with 0.25 dropout rate. We train the model with the Adam optimizer [11] for 100 epochs. The initial learning rate is 0.001 and decays to 25% every 30 epochs. For our 3D model, we train the model with the Adam optimizer for 60 epochs. We also use 0.001 as the initial learning, but it decays to 10% every 40 epochs. We use up to 20 conformers for each molecule during training and 40 conformers per molecule for prediction. The training batch size is set to 256 for both our 2D and 3D models. Our implementation is available as part of the MoleculeX[2] software package under the *BasicProp/kddcup2021* folder.

[2] `https://github.com/divelab/MoleculeX`

Table 2: Results on PCQM4M-LSC in terms of validation MAE [eV] on the 5 new splits. As the results in Table 1, results of our 3D model are still computed on the molecules that have conformers generated by RDKit. If a molecule do not have available conformers, we ignore the 3D model prediction in the ensemble model.

Method	1st split	2nd split	3rd split	4th split	5th split
Our 2D model	0.1217	0.1181	0.1201	0.1222	0.1192
	0.1191	0.1188	0.1186	0.1181	0.1194
	0.1176	0.1190	0.1191	0.1188	0.1183
	0.1190	0.1185	0.1192	0.1194	0.1198
Our 3D model	0.1214	0.1208	0.1216	0.1222	0.1214
Ensemble	0.1117	0.1113	0.1120	0.1111	0.1114

D. Experiments for the 2021 KDD Cup on OGB large-Scale Challenge,

New splits. Notably, we observe that it is useful to include more available data for training. Hence, we consider using the original validation set for training. On the other hand, model selection is also necessary because the performance could fluctuate over different training epochs. Therefore, we randomly divide the original validation set into 5 sets. Each of them could be used for validation, and the rest are added into the training set. In this case, we can obtain 5 different $88\%/2\%/10\%$ splits. For improving performance on the hidden test set, we can train multiple models on these 5 splits and use the ensemble of the predictions of these models on test set. In this case, we can leverage all the available data, including the original training and validation set, for training, while conducting model selection at the same time using the new 2% validation set on the corresponding split. The reason why we divide the original validation set into 5 sets is that the variance brought by the random division is small enough and the numerical result in terms of MAE is similar to the original validation set for a given trained model. For example, a GIN-virtual model trained on the original training set achieves 0.1396 MAE on the original validation set and 0.1393 0.1413 MAE on random 20% of the original validation set.

Results. As discussed above, we obtain 5 new splits by moving partial validation data to training set. We train multiple models on such 5 splits and evaluate them on their corresponding validation sets. To be specific, we train 4 2D models and 1 3D model on each split. The results are summarized in Table 2. According to the results of individual models, we can observe that the improvement owned to more training data is remarkable, compared to the results of our 2D and 3D model in Table 1. In addition, we evaluate the ensemble of predictions obtained by multiple 2D and 3D models on validation set for each split. We find that the ensemble improves the performance obviously, and we also observe that the ensemble of different models outperforms the ensemble of several identical models if the number of used models is the same, which indicates that the 2D and 3D models capture different and complementary information for predicting the HUMO-LOMO energy gap.

For predicting the hidden test set, we ensemble the predictions of the multiple models, including 4 2D models and 1 3D model, on each split. Afterwards, we take the average over the predictions obtained from all 5 splits. The MAE of our final prediction on the whole hidden test set is 0.1235, evaluated by the OGB team.