ChemRL-GEM: Geometry Enhanced Molecular Representation Learning for Property Prediction

Xiaomin Fang (fangxiaomin01@baidu.com)
Baidu Inc.
Lihang Liu
Baidu Inc.
Jieqiong Lei
Tsinghua University
Donglong He
Baidu Inc.
Shanzhuo Zhang
Baidu Inc.
Jingbo Zhou
Baidu Inc.
Fan Wang
Baidu Inc.
Hua Wu
Baidu
Haifeng Wang
Baidu

Article

Keywords: molecular property prediction, drug development, GNN, ChemRL

Posted Date: August 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-670872/v1

Version of Record: A version of this preprint was published at Nature Machine Intelligence on February 7th, 2022. See the published version at https://doi.org/10.1038/s42256-021-00438-4.

ChemRL-GEM: Geometry Enhanced Molecular Representation Learning for Property Prediction

Abstract

Effective molecular representation learning is of great importance to facilitate molecular property prediction, which is a fundamental task for the drug and material industry. Recent advances in graph neural networks (GNNs) have shown great promise in applying GNNs for molecular representation learning. Moreover, a few recent studies have also demonstrated successful applications of self-supervised learning methods to pre-train the GNNs to overcome the problem of insufficient labeled molecules. However, existing GNNs and pre-training strategies usually treat molecules as topological graph data without fully utilizing the molecular geometry information. Whereas, the three-dimensional (3D) spatial structure of a molecule, a.k.a molecular geometry, is one of the most critical factors for determining molecular physical, chemical, and biological properties. To this end, we propose a novel **G**eometry **E**nhanced **M**olecular representation learning method (GEM) for **Chem**ical **R**epresentation **L**earning (ChemRL). At first, we design a geometry-based GNN architecture that simultaneously models atoms, bonds, and bond angles in a molecule. To be specific, we devised double graphs for a molecule: The first one encodes the atom-bond relations; The second one encodes bond-angle relations. Moreover, on top of the devised GNN architecture, we propose several novel geometry-level self-supervised learning strategies to learn spatial knowledge by utilizing the local and global molecular 3D structures. We compare ChemRL-GEM with various state-of-the-art (SOTA) baselines on different molecular benchmarks and exhibit that ChemRL-GEM can significantly outperform all baselines in both regression and classification tasks. For example, the experimental results show an overall improvement of 8.8% on average compared to SOTA baselines on the regression tasks, demonstrating the superiority of the proposed method.

1 Introduction

Molecular property prediction has been widely considered as one of the most critical tasks in computational drug and materials discovery, since many methods rely on predicted molecular properties to evaluate, select and generate molecules [43, 52]. With the development of deep neural networks (DNNs), molecular representation learning exhibits a great advantage over feature engineering-based methods, which has attracted increasing research attention to tackle the molecular property prediction problem.

Recently, graph neural networks (GNNs) for molecular representation learning have become an emerging research area, which regard the topology of atoms and bonds as a graph, and propagate messages of each element to its neighbors [24, 40, 44, 45]. However, one major obstacle to hinder the successful application of GNNs (and DNNs) in molecule property prediction is the scarity of labeled data, which is also a common research challenge in Natural Language Processing (NLP) [10, 22] and Computer Vision (CV) [11, 17] communities. Inspired by the success of self-supervised learning,

Preprint. Under review.

recent studies [23, 40] start to utilize large-scale molecules with self-supervised methodology to pre-train the molecular representation, which have achieved substantial improvements.

However, existing representation methods based on GNNs only encode the topology information of the molecules, neglecting the molecular geometry information, i.e. the three-dimensional (3D) spatial structure of a molecule. Although some works [42, 30] take the atomic distance into edge features to consider partial geometry information, they fail to thoroughly model the geometry information, such as bond-angle information. As a result, existing graph-based molecular representations usually can not distinguish molecules with the same topology but different geometries (i.e., Geometric Isomerism [14]). However, such information plays an important role in determining molecules' physical, chemical, and biological activities. For example, due to the geometrical difference, water solubility (a critical metric of drug-likeness) of the two molecules illustrated in Figure 1 are different, even though they have the same topology. Cis-Platin and trans-Platin are another example that have the same topology but different geometries: cis-Platin is a popular chemotherapy drug used to treat a number of cancers, whereas trans-Platin has no cytotoxic activity [35].

As for the self-supervised learning methods, previous works typically do masking and predicting in nodes, edges or contexts in the topology [23, 40]. We argue that these tasks only enable the model to learn superficial chemical laws such as "which atom/group could be connected to a double bond", but lack the ability to learn the geometry knowledge, such as "the difference between the bond angles of the two molecules" shown in Figure 1. It is also desirable to propose a delicately designed self-supervised learning methods to fully capture the intrinsic differences in atomic interaction forces due to different molecular geometries.

To solve these problems, we propose a novel **G**eometry **E**nhanced **M**olecular representation learning method (GEM) for **Chem**ical **R**epresentation **L**earning (ChemRL). Firstly, to make the message passing sensitive to geometries, we model the atoms, bonds, and bond angles simultaneously by designing a geometry-based GNN architecture (GeoGNN), which contains two graphs: the first encodes the atom-bond relations; the second encodes the bond-angle relations. Secondly, we pre-train the GeoGNN to learn the knowledge of both the chemical laws and the geometries by designing various geometry-level self-supervised learning tasks. To verify the effectiveness of the proposed ChemRL-GEM, we compared it with several SOTA baselines on a dozen of molecular property prediction benchmarks. Our exhaustive experimental study demonstrates the superiority of ChemRL-GEM.

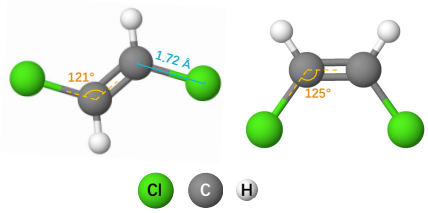


Figure 1: Comparison between two molecules (cis-1,2-DCE and trans-1,2-DCE) with the same topology but different geometries. The two chlorine atoms are on the different sides in the left molecule, while the same sides in the right molecule.

Our contributions can be summarized as follows:

- We propose a geometry-based graph neural network, GeoGNN, to encode both the topology and the geometry information of molecules.
- We introduce multiple geometry-level self-supervised learning tasks to learn the molecular 3D spatial knowledge in addition to other self-supervised learning tasks.
- We evaluated ChemRL-GEM thoroughly on various molecular property prediction datasets. Experimental results demonstrate that ChemRL-GEM significantly outperforms competitive baselines on multiple benchmarks.

2 Preliminaries

2.1 Graph-based Molecular Representation

A molecule consists of atoms, and the neighboring atoms are connected by the chemical bonds, which can be naturally represented by a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a node set and $\mathcal{E}$ is an edge set. An atom in the molecule is regarded as a node $v \in \mathcal{V}$ and a chemical bond in the molecule is regarded as an edge $(u, v) \in \mathcal{E}$ connecting atoms u and v.

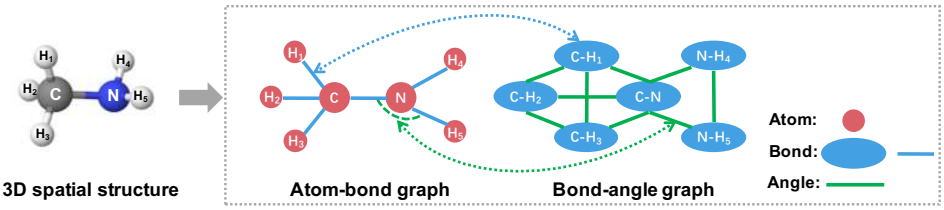


Figure 2: Illustration of the atom-bond graph and the bond-angle graph. The left figure shows the structure of Methanamine in the 3D space. We can easily encode its geometry information with the help of the atom-bond graph that describes the relations between atoms and bonds and the bond-angle graph that describes the relations between bonds and bond angles.

Graph neural networks (GNNs) can be seen as message passing neural networks [18], which is useful for predicting molecular properties. Following the definitions of the previous GNNs [55], the features of a node v are represented by x_v and the features of an edge (u, v) are represented by x_{uv}. Taking node features, edge features and the graph structure as inputs, a GNN learns the representation vectors of the nodes and the entire graph, where the representation vector of a node v is denoted by h_v and the representation vector of the entire graph is denoted by h_G. A GNN iteratively updates a node's representation vector by aggregating the messages from the node's neighbors. Given a node v, its representation vector $h_v^{(k)}$ at the k-th iteration is formalized by

$$\begin{aligned} a_v^{(k)} &= \textit{AGGREGATE}^{(k)}(\{(h_v^{(k-1)}, h_u^{(k-1)}, x_{uv} | u \in \mathcal{N}(v)\}), \\ h_v^{(k)} &= \textit{COMBINE}^{(k)}(h_v^{(k-1)}, a_v^{(k)}). \end{aligned} \tag{1}$$

where $\mathcal{N}(v)$ is the set of neighbors of node v, $\textit{AGGREGATE}^{(k)}$ is the aggregation function for aggregating messages from a node's neighborhood, and $\textit{COMBINE}^{(k)}$ is the update function for updating the node representation. We initialize $h_v^{(0)}$ by the feature vector of node v, i.e., $h_v^{(0)} = x_v$.

READOUT function is introduced to integrate the nodes' representation vectors at the final iteration so as to gain the graph's representation vector h_G, which is formalized as

$$h_G = \textit{READOUT}(h_v^{(K)} | v \in \mathcal{V}), \tag{2}$$

where K is the number of iterations. In most cases, *READOUT* is a permutation invariant pooling function, such as summation and maximization. The graph's representation vector h_G can then be used for downstream task predictions.

2.2 Pre-training Methods for GNNs

In the molecular representation learning community, recently several works [48, 23, 40] have explored the power of self-supervised learning to improve the generalization ability of GNN models on downstream tasks. They mainly focus on two kinds of self-supervised learning tasks: the node-level (edge-level) tasks and the graph-level tasks.

The node-level self-supervised learning tasks are devised to capture the local domain knowledge. For example, some studies randomly mask a portion of nodes or sub-graphs and then predict their properties by the node/edge representation. The graph-level self-supervised learning tasks are used to capture the global information, like predicting the graph properties by the graph representation. Usually, the graph properties are domain-specific knowledge, such as experimental results from biochemical assays or the existence of molecular functional groups.

3 The ChemRL-GEM Framework

This section introduces the details of our proposed **G**eometry **E**nhanced **M**olecular representation learning method (GEM) for **Chem**ical **R**epresentation **L**earning, which includes two parts: a novel geometry-based GNN and various geometry-level self-supervised learning tasks.

3.1 Geometry-based Graph Neural Network

We propose a **Geo**metry-based **G**raph **N**eural **N**etwork (GeoGNN) that encodes the molecular geometries by modeling the atom-bond-angle relations, while traditional GNNs only consider the relations between atoms and bonds.

For a molecule, we denote the atom set as $\mathcal{V}$, the bond set as $\mathcal{E}$, and the bond angle set as $\mathcal{A}$. We introduce atom-bond graph G, and bond-angle graph H for each molecule, as illustrated in Figure 2. The atom-bond graph is defined as $G = (\mathcal{V}, \mathcal{E})$, where atom $u \in \mathcal{V}$ is regarded as the node of G and bond $(u, v) \in \mathcal{E}$ as the edge of G, connecting atom u and atom v. Similarly, the bond-angle graph is defined as $H = (\mathcal{E}, \mathcal{A})$, where bond $(u, v) \in \mathcal{E}$ is regarded as the node of H and bond angle $(u, v, w) \in \mathcal{A}$ as the edge of H, connecting bond (u, v) and bond (v, w). We use x_u as the initial features of atom u, x_{uv} as the initial features of bond (u, v) and x_{uvw} as the initial features of bond angle (u, v, w). The atom-bond graph G and the bond-angle graph H, as well as atom features, bond features and bond angle features are taken as the inputs of GeoGNN.

GeoGNN learns the representation vectors of atoms and bonds iteratively. For the k-th iteration, the representation vectors of atom u and bond (u, v) are denoted by h_u and h_{uv}, respectively. We initialize $h_u^{(0)} = x_u$ and $h_{uv}^{(0)} = x_{uv}$. In order to connect the atom-bond graph G and bond-angle graph H, the representation vectors of the bonds as the communication links between G and H, as shown in Figure 3. More concretely, the bonds' representation vectors are learned by aggregating the messages from the neighboring bonds and corresponding bond angles in the bond-angle graph H. Then, the learned bond representation vectors are taken as edge features of the atom-bond graph G and help to learn the atoms' representation vectors.

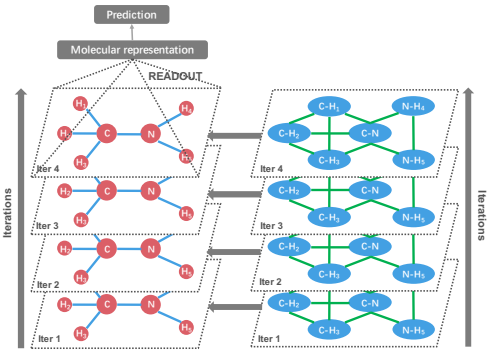


Figure 3: Overall architecture of GeoGNN

Given bond (u, v), its representation vector $h_{uv}^{(k)}$ at the k-th iteration is formalized by

$$\begin{aligned} a_{uv}^{(k)} =& \textit{AGGREGATE}_{\textit{bond-angle}}^{(k)}(\{(h_{uv}^{(k-1)}, h_{uw}^{(k-1)}, x_{wuv}) : w \in \mathcal{N}(u)\} \\ & \cup \{(h_{uv}^{(k-1)}, h_{vw}^{(k-1)}, x_{uvw}) : w \in \mathcal{N}(v)\}), \\ h_{uv}^{(k)} =& \textit{COMBINE}_{\textit{bond-angle}}^{(k)}(h_{uv}^{(k-1)}, a_{uv}^{(k)}). \end{aligned} \tag{3}$$

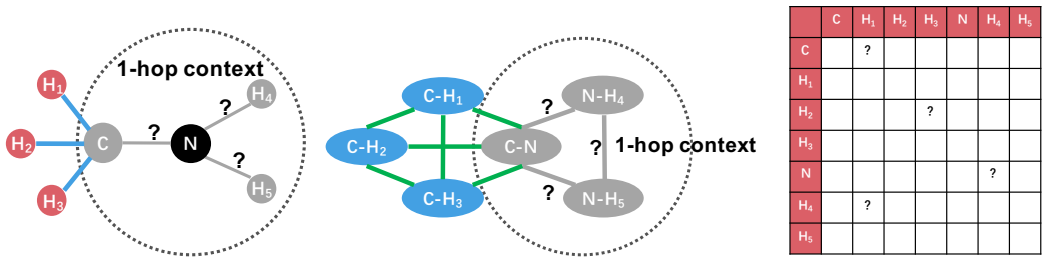
Here, $\mathcal{N}(u)$ and $\mathcal{N}(v)$ denote the neighboring atoms of atom u and atom v, respectively. $\{(u, w) : w \in \mathcal{N}(u)\} \cup \{(v, w) : w \in \mathcal{N}(v)\}$ are the neighboring bonds of bond (u, v). $\textit{AGGREGATE}_{\textit{bond-angle}}$ is the message aggregation function, and $\textit{COMBINE}_{\textit{bond-angle}}$ is the update function for bond-angle graph H. In this way, the information from the neighboring bonds and the corresponding bond angles is aggregated into $a_{uv}^{(k)}$. Then, the representation vector of bond (u, v) is updated according to the aggregated information. With the learned representation vectors of the bonds from bond-angle graph $\mathcal{H}$, given an atom u, its representation vector $h_u^{(k)}$ at the k-th iteration can be formalized as

$$\begin{aligned} a_u^{(k)} &= \textit{AGGREGATE}_{\textit{atom-bond}}^{(k)}(\{(h_u^{(k-1)}, h_v^{(k-1)}, h_{uv}^{(k-1)}) : v \in \mathcal{N}(u)\}), \\ h_u^{(k)} &= \textit{COMBINE}_{\textit{atom-bond}}^{(k)}(h_u^{(k-1)}, a_u^{(k)}). \end{aligned} \tag{4}$$

Similarly, $\mathcal{N}(u)$ denotes the neighboring atoms of atom u, $\textit{AGGREGATE}_{\textit{atom-bond}}$ is the message aggregation function for atom-bond graph G, and $\textit{COMBINE}_{\textit{atom-bond}}$ is the update function. For atom u, messages are aggregated from the neighboring atoms and the corresponding bonds. Note that, the messages of the bonds are learned from the bond-angle graph H. Then, the aggregated messages update the representation vector of atom u.

The representation vectors of the atoms at the final iteration are integrated to gain the molecular representation vector h_G by the *READOUT* function, which is formalized as

$$h_G = \textit{READOUT}(h_u^{(K)} | u \in \mathcal{V}), \tag{5}$$



(a) Predict bond lengths (b) Predict bond angles (c) Predict atomic distances

Figure 4: Demonstration of geometry-level self-supervised learning tasks. (The black circle represents the selected atom. The gray circles and lines represent the neighboring masked atoms, bonds, and bond angles.)

where K is the number of iterations. The molecule's representation vector h_G is used to predict the molecular properties.

3.2 Geometry-level Self-supervised Learning Tasks

To further boost the generalization ability of GeoGNN, we propose three geometry-level self-supervised learning tasks to pre-train GeoGNN: 1) the bond lengths prediction; 2) the bond angles prediction; 3) the atomic distance matrices prediction. The bond lengths and bond angles describe the local spatial structures, while the atomic distance matrices describe the global spatial structures.

3.2.1 Local Spatial Structures

The bond lengths and the bond angles are the most important molecular geometrical parameters. The bond length is the average distance between two joint atoms in a molecule, reflecting the bond strength between the atoms, while the bond angle is the angle connecting two consecutive bonds, including three atoms, describing the local spatial structure of a molecule.

In order to learn the local spatial structures, we construct self-supervised learning tasks that predict the bond lengths and bond angles. Firstly, for a molecule, we randomly select 15% of atoms. For each selected atom, we extract 1-hop neighboring atoms and bonds, as well as the bond angles formed by that selected atom. Secondly, we mask the features of these atoms, bonds, and bond angles in the 1-hop context. The representation vectors of the extracted atoms and bonds at the final iteration of GeoGNN are used to predict the extracted bond lengths and bond angles. Figure 4(a) and Figure 4(b) show the self-supervised learning tasks based on bond lengths and bond angles. More concretely, for a selected atom v, the loss functions of the self-supervised tasks of local geometry information are defined as follow:

$$\begin{aligned} L_{length}(v) &= \frac{1}{|\mathcal{N}(v)|}(f_{length}(h_u^{(K)}, h_v^{(K)}) - l_{uv})^2, u \in \mathcal{N}(v); \\ L_{angle}(v) &= \frac{1}{|\mathcal{N}(v)|^2}(f_{angle}(h_u^{(K)}, h_v^{(K)}, h_w^{(K)}) - \phi_{uvw})^2, u, w \in \mathcal{N}(v). \end{aligned} \tag{6}$$

where $f_{length}(\cdot)$ is the network predicting the bond lengths, and $f_{angle}(\cdot)$ is the network predicting the bond angles. l_{uv} denotes the length of the bond connecting atom u and atom v and ϕ_{uvw} denotes the degree of the bond angle connecting bonds (u, v) and (v, w). The task of predicting the local spatial structures can be seen as a node-level self-supervised learning task.

3.2.2 Global Spatial Structures

Except for the tasks for learning local spatial structures, we also design the atomic distance matrices prediction task for learning the global molecular geometry. We construct the atomic distance matrix for each molecule based on the 3D coordinates of the atoms. Then, we predict the elements in the distance matrix, shown in Figure 4(c). We use d_{uv} to denote the distance between two atoms u and v in the molecule. Note that, for two molecules with the same topological structures, the spatial distances between the corresponding atoms could vary greatly. Thus, for a molecule, rather than take

predicting atomic distance matrix as a regression problem, we take it as a multi-class classification problem by discretizing the atomic distances. The loss function is defined as

$$L_{distance} = \frac{1}{|\mathcal{V}|^2} \sum_{u,v \in \mathcal{V}} -bin^T(d_{uv}) \cdot log(f^T_{distance}(h^{(K)}_u, h^{(K)}_v)) \tag{7}$$

where $f_{distance}(\cdot)$ is the network predicting the distribution of atomic distances, the $bin(\cdot)$ function is used to discretize the atomic distance d_{uv} into a one-hot vector, and $log(\cdot)$ is the logarithmic function. The task predicting the bond lengths can be seen as a special case of the task predicting the atomic distances. The former focuses more on the accurate local spatial structures, while the latter focuses more on the distribution of the global spatial structures. To pre-train GeoGNN, we consider both the local spatial structures and global spatial structures for each molecule by summing up the loss functions defined in Equation 6 and Equation 7.

4 Experiments

To thoroughly evaluate the performance of ChemRL-GEM, we compare it with multiple state-of-the-art (SOTA) methods on 12 benchmark datasets from MoleculeNet [53] with various molecular property prediction tasks, such as physical, chemical, and biophysics mechanics. The source codes for the experiments will be released to ensure reproducibility when the paper is published.

4.1 Pre-training Settings

Datasets. We use 20 million unlabelled molecules sampled from Zinc15 [46], a public access database that contains purchasable "drug-like" compounds, to pre-train GeoGNN. We randomly sample 90% of the molecules for training and the remaining for evaluation.

Self-supervised Learning Task Settings. We utilize the geometry-level and graph-level tasks to pre-train GeoGNN. For the geometry-level tasks, we utilize the Merck molecular force field [20] function from the RDKit[1] package, an open-source cheminformatics toolkit [29], to obtain the simulated 3D coordinates of the atoms in the molecules. The geometric features of the molecule, including bond lengths, bond angles and atomic distance matrices, are calculated by the simulated 3D coordinates. For the graph-level tasks, we predict two kinds of molecular fingerprints: 1) The Molecular ACCess System (MACCS) key [12]; 2) The extended-connectivity fingerprint (ECFP) [39].

4.2 Molecular Property Prediction Settings

Datasets and Splitting Method. We conduct experiments on multiple molecular benchmarks from the MoleculeNet [53], including both classification and regression tasks [47, 33, 38, 16, 25, 28]. Following the previous work [23], we split all the 12 datasets with scaffold split [37], which splits molecules according to the their scaffold (molecular substructure). Rather than splitting the dataset randomly, scaffold split is a more challenging splitting method, which can better evaluate the generalization ability of the models on out-of-distribution data samples. Based on scaffold split, we split the molecules in each dataset into training set, validation set, and test set by the ratio of 8:1:1. We run each method for 100 epochs on the training set in the training process and then select the epoch according to the validation set. The selected epoch is evaluated on the test set.

GNN Architecture. We use the *AGGREGATE* function and *COMBINE* function defined in Graph Isomorphism Network (GIN) [55]. To further improve the performance, residual connections [21], layer normalization [1], and graph normalization [7] are incorporated into GIN. Also, we use the average pooling as the *READOUT* function to obtain the graph representation.

Hyper-parameters and Evaluation Metrics. We use Adam Optimizer[26] with learning rate of 0.001 for all our models. For each dataset, we train the model with batch size of 32. As suggested by the MoleculeNet [53], we use the average ROC-AUC [4] as the evaluation metric for the 6 binary classification datasets. With respect to the regression datasets, for FreeSolv [34], ESOL [9], and Lipo [15], we use Root Mean Square Error (RMSE), and for QM7 [3], QM8 [36], and QM9 [41], we use Mean Average Error (MAE). We execute 4 independent runs for each method and report the mean and the standard deviation of the metrics.

[1] http://www.rdkit.org

Table 1: Overall performance for molecular property prediction.

Regression (Lower is better)						
	RMSE			MAE		
Dataset	ESOL	FreeSolv	Lipo	QM7	QM8	QM9
#Molecules	1128	642	4200	6830	21786	133885
#Prediction tasks	1	1	1	1	12	12
D-MPNN [56]	$1.050_{(0.008)}$	$2.082_{(0.082)}$	$0.683_{(0.016)}$	$103.5_{(8.6)}$	$0.0190_{(0.0001)}$	$0.00814_{(0.00001)}$
AttentiveFP [54]	$0.877_{(0.029)}$	$2.073_{(0.183)}$	$0.721_{(0.001)}$	$72.0_{(2.7)}$	$0.0179_{(0.0001)}$	$0.00812_{(0.00001)}$
N-Gram$_{RF}$ [32]	$1.074_{(0.107)}$	$2.688_{(0.085)}$	$0.812_{(0.028)}$	$92.8_{(4.0)}$	$0.0236_{(0.0006)}$	$0.01037_{(0.00016)}$
N-Gram$_{XGB}$ [32]	$1.083_{(0.082)}$	$5.061_{(0.744)}$	$2.072_{(0.030)}$	$81.9_{(1.9)}$	$0.0215_{(0.0005)}$	$0.00964_{(0.00031)}$
PretrainGNN [23]	$1.100_{(0.006)}$	$2.764_{(0.002)}$	$0.739_{(0.003)}$	$113.2_{(0.6)}$	$0.0200_{(0.0001)}$	$0.00922_{(0.00004)}$
GROVER$_{base}$ [40]	$0.983_{(0.090)}$	$2.176_{(0.052)}$	$0.817_{(0.008)}$	$94.5_{(3.8)}$	$0.0218_{(0.0004)}$	$0.00984_{(0.00055)}$
GROVER$_{large}$ [40]	$0.895_{(0.017)}$	$2.272_{(0.051)}$	$0.823_{(0.010)}$	$92.0_{(0.9)}$	$0.0224_{(0.0003)}$	$0.00986_{(0.00025)}$
ChemRL-GEM	$\mathbf{0.798_{(0.029)}}$	$\mathbf{1.877_{(0.094)}}$	$\mathbf{0.660_{(0.008)}}$	$\mathbf{58.9_{(0.8)}}$	$\mathbf{0.0171_{(0.0001)}}$	$\mathbf{0.00746_{(0.00001)}}$

Classification (Higher is better)							
Dataset	BACE	BBBP	ClinTox	SIDER	Tox21	ToxCast	Avg
#Molecules	1513	2039	1478	1427	7831	8575	-
#Prediction tasks	1	1	2	27	12	617	-
D-MPNN [56]	$0.809_{(0.006)}$	$0.710_{(0.003)}$	$0.906_{(0.006)}$	$0.570_{(0.007)}$	$0.759_{(0.007)}$	$0.655_{(0.003)}$	0.735
AttentiveFP [54]	$0.784_{(0.022)}$	$0.643_{(0.018)}$	$0.847_{(0.003)}$	$0.606_{(0.032)}$	$0.761_{(0.005)}$	$0.637_{(0.002)}$	0.713
N-Gram$_{RF}$ [32]	$0.779_{(0.015)}$	$0.697_{(0.006)}$	$0.775_{(0.040)}$	$0.668_{(0.007)}$	$0.743_{(0.004)}$	-	-
N-Gram$_{XGB}$ [32]	$0.791_{(0.013)}$	$0.691_{(0.008)}$	$0.875_{(0.027)}$	$0.655_{(0.007)}$	$0.758_{(0.009)}$	-	-
PretrainGNN [23]	$0.845_{(0.007)}$	$0.687_{(0.013)}$	$0.726_{(0.015)}$	$0.627_{(0.008)}$	$0.781_{(0.006)}$	$0.657_{(0.006)}$	0.721
GROVER$_{base}$ [40]	$0.826_{(0.007)}$	$0.700_{(0.001)}$	$0.812_{(0.030)}$	$0.648_{(0.006)}$	$0.743_{(0.001)}$	$0.654_{(0.004)}$	0.730
GROVER$_{large}$ [40]	$0.810_{(0.014)}$	$0.695_{(0.001)}$	$0.762_{(0.037)}$	$0.654_{(0.001)}$	$0.735_{(0.001)}$	$0.653_{(0.005)}$	0.718
ChemRL-GEM	$\mathbf{0.856_{(0.011)}}$	$\mathbf{0.724_{(0.004)}}$	$0.901_{(0.013)}$	$\mathbf{0.672_{(0.004)}}$	$\mathbf{0.781_{(0.001)}}$	$\mathbf{0.692_{(0.004)}}$	**0.771**

Baselines. We compare the proposed method with various competitive baselines. D-MPNN [56] and AttentiveFP [54] are the GNNs without pre-training, while N-Gram [32], PretrainGNN [23], and GROVER [40] are the methods with pre-training. N-Gram assembles the node embeddings in short walks in the graph to obtain the graph representation and then leverages Random Forest or XGBoost to predict the molecular properties. PretrainGNN implements several types of self-supervised learning tasks, among which we report the best result. GROVER integrates GNN into Transformer with the context prediction task and the functional motif prediction task, and we report the results of GROVER$_{base}$ and GROVER$_{large}$ with different network capacity.

More network and experimental details can be found in the Appendices.

4.3 Overall Performance

The overall performance of ChemRL-GEM along with other methods on the molecular property prediction benchmarks is summarized in Table 1[2], where the SOTA results are shown in bold and the cells in gray indicate the previous SOTA results. The numbers in brackets are the standard deviation. From Table 1, we have the following observations: 1) ChemRL-GEM achieves SOTA results on 11/12 datasets. On the regression tasks, ChemRL-GEM achieves an overall relative improvement of 8.8% on average compared to the previous SOTA results in each dataset. While on the classification tasks, it achieves an overall relative improvement of 3.7% on the average ROC-AUC compared to the previous SOTA result from D-MPNN. 2) Since the tasks of the regression datasets, such as the water solubility prediction in the ESOL dataset and the electronic properties prediction in the QM7 dataset, are much more correlated to the molecular geometries than the classification tasks, ChemRL-GEM achieves more considerable improvement on the regression datasets compared to the classification datasets. 3) ChemRL-GEM does not achieve SOTA results in the Clintox. It is possible that the Clintox dataset is highly unbalanced, with only 9 positive samples in the test set, which may cause

[2]Since N-Gram on ToxCast is too time-consuming, we are not able to finish in time but will be added later.

Table 2: Performance of different GNN architectures on the regression datasets.

	RMSE			MAE		
Method	ESOL	FreeSolv	Lipo	QM7	QM8	QM9
GIN [55]	$1.067_{(0.051)}$	$2.346_{(0.122)}$	$0.757_{(0.022)}$	$110.3_{(7.2)}$	$0.0199_{(0.0002)}$	$0.00886_{(0.00005)}$
GAT [50]	$1.556_{(0.085)}$	$3.559_{(0.050)}$	$1.021_{(0.029)}$	$103.0_{(4.4)}$	$0.0224_{(0.0005)}$	$0.01117_{(0.00018)}$
GCN [27]	$1.211_{(0.052)}$	$3.174_{(0.308)}$	$0.773_{(0.007)}$	$100.0_{(3.8)}$	$0.0203_{(0.0005)}$	$0.00923_{(0.00019)}$
D-MPNN [56]	$1.050_{(0.008)}$	$2.082_{(0.082)}$	$0.683_{(0.016)}$	$103.5_{(8.6)}$	$0.0190_{(0.0001)}$	$0.00814_{(0.00009)}$
AttentiveFP [54]	$0.877_{(0.029)}$	$2.073_{(0.183)}$	$0.721_{(0.001)}$	$72.0_{(2.7)}$	$0.0179_{(0.0001)}$	$0.00812_{(0.00001)}$
GTransformer [40]	$2.298_{(0.118)}$	$4.480_{(0.155)}$	$1.112_{(0.029)}$	$161.3_{(7.1)}$	$0.0361_{(0.0008)}$	$0.00923_{(0.00019)}$
GeoGNN	$\mathbf{0.832_{(0.010)}}$	$\mathbf{1.857_{(0.071)}}$	$\mathbf{0.666_{(0.015)}}$	$\mathbf{59.0_{(3.4)}}$	$\mathbf{0.0173_{(0.0004)}}$	$\mathbf{0.00746_{(0.00003)}}$

Table 3: Performance of different pre-training strategies on the regression datasets.

	RMSE			MAE		
Pre-train Method	ESOL	FreeSolv	Lipo	QM7	QM8	QM9
w/o pre-train	$0.832_{(0.010)}$	$1.857_{(0.071)}$	$0.666_{(0.015)}$	$59.0_{(3.4)}$	$0.0173_{(0.0004)}$	$0.00746_{(0.00003)}$
Context+Graph	$0.837_{(0.027)}$	$1.982_{(0.098)}$	$0.664_{(0.011)}$	$72.1_{(2.3)}$	$\mathbf{0.0171_{(0.0003)}}$	$0.00748_{(0.00005)}$
Graph	$0.815_{(0.025)}$	$1.950_{(0.069)}$	$0.665_{(0.012)}$	$63.1_{(2.8)}$	$0.0174_{(0.0002)}$	$0.00750_{(0.00001)}$
Geometry	$0.825_{(0.017)}$	$\mathbf{1.701_{(0.147)}}$	$\mathbf{0.660_{(0.021)}}$	$\mathbf{58.2_{(0.5)}}$	$\mathbf{0.0171_{(0.0001)}}$	$\mathbf{0.00734_{(0.00003)}}$
Geometry+Graph	$\mathbf{0.798_{(0.029)}}$	$1.876_{(0.094)}$	$\mathbf{0.660_{(0.008)}}$	$58.9_{(0.8)}$	$\mathbf{0.0171_{(0.0001)}}$	$0.00746_{(0.00001)}$

unstable results. We also conduct experiments on other splitting methods, and ChemRL-GEM still achieves SOTA results. Please refer to the Appendix B for more details.

4.4 Ablation Studies of ChemRL-GEM

Contribution of GeoGNN. We investigate the effect of GeoGNN on regression datasets which are more related to the molecular geometries. GeoGNN is compared with multiple GNN architectures, including the commonly used GNN architectures, GIN [55], GAT [50], and GCN [27], as well as the architectures specially designed for molecular representation, D-MPNN [56], AttentiveFP [54], and GTransformer [40]. From Table 2, we can conclude that GeoGNN significantly outperforms other GNN architectures on all the regression datasets since GeoGNN utilizes a clever architecture that incorporates geometrical parameters even though the 3D coordinates of the atoms are simulated. The overall relative improvement is 7.9% compared to the best results of previous methods.

Although the simulated 3D coordinates by RDKit are not accurate, they still provide the coarse information about the 3D conformation. Furthermore, we instead utilize the 3D coordinates of the molecules provided by QM9, which are much more accurate. Surprisingly GeoGNN with the accurate 3D coordinates achieves an average MAE of 0.00652, while GeoGNN with the coarse 3D coordinates achieves an average MAE of 0.00746. Such improvement further demonstrates the increasing power of GeoGNN on learning molecular representations when providing more accurate 3D coordinates.

Contribution of Geometry-level Tasks. To study the effect of the proposed geometry-level self-supervised learning tasks, we apply different types of self-supervised learning tasks to pre-train GeoGNN on the regression datasets. In Table 3, *w/o pre-training* denotes the GeoGNN network without pre-training, *Geometry* denotes our proposed geometry-level tasks, *Graph* denotes the graph-level task that predicts the molecular fingerprints, and *Context* [40] denotes a node-level task that predicts the atomic context. In general, the methods with geometry-level tasks are better than that without it. Furthermore, *Geometry* performs better than *Geometry+Graph* in the regression tasks, which may due to the weak connection between molecular fingerprints and the regression tasks.

5 Related Work

5.1 Molecular Representation

Current molecular representations can be categorized into three types: molecular fingerprints, sequence-based representations and graph-based representations.

Molecular Fingerprints. Molecular fingerprints, such as ECFP [39] and MACCS [12], are commonly used for molecular representations by traditional machine learning methods [6, 8, 13, 24], which encode a molecule into a sequence of bits according to the molecules' topological substructures. However, molecular fingerprints lack the ability to represent complex global structures, since they only focus on the local substructures.

Sequence-based Representations. Some studies [19, 24] take SMILES strings [51] that describe the molecules by strings as inputs, and leverage sequence-based models, such as Recurrent Neural Networks and Transformer [57, 49], to learn the molecular representations. However, the same molecule could be represented by more than two SMILES strings, resulting in ambiguity of the representation. Besides, it is laborious for sequence-based representation to model some molecular topological structures, such as rings.

Graph-based Representations. Many works [24, 40, 44, 45, 18] have showcased the great potential of graph neural networks on modeling molecules by taking each atom as a node and each chemical bond as an edge. For example, Attentive FP [54] proposes to extend graph attention mechanism in order to learn aggregation weights. Furthermore, several works [42, 30] start to take the atomic distance into edge features to consider partial geometry information. However, they still lack the ability to model the full geometry information due to the shortage of traditional GNN architecture.

5.2 Pre-training for Graph Neural Networks

Self-supervised learning [10, 11, 17, 22, 31] has achieved great success in natural language processing (NLP), computer vision (CV), and other domains, which trains unlabeled samples in a supervised manner to alleviate the over-fitting issue and improve data utilization efficiency. Recently, some studies [23, 40] apply self-supervised learning methods to GNNs for molecular property prediction to overcome the insufficiency of the labeled samples. These works learn the molecular representation vectors by exploiting the node-level and graph-level tasks, where the node-level tasks learn the local domain knowledge by predicting the node properties and the graph-level tasks learn the global domain knowledge by predicting biological activities. Although existing self-supervised learning methods can boost the generalization ability, they neglect the spatial knowledge that is strongly related to the molecular properties.

6 Conclusions and Future Work

Efficient molecular representation learning is crucial for molecular property prediction. Existing works that apply GNNs and pre-training methods for molecular property prediction fail to fully utilize the molecular geometries described by bonds, bond angles, and other geometrical parameters. To this end, we design a geometry-based GNN for learning the atom-bond-angle relations that utilize the information of the bond angles by introducing a bond-angle graph on top of the atom-bond graph. Moreover, multiple geometry-level self-supervised learning methods are constructed to predict the molecular geometries to capture spatial knowledge. In order to verify the effectiveness of ChemRL-GEM, extensive experiments were conducted, comparing it with multiple competitive baselines. ChemRL-GEM significantly outperforms other methods on 12 benchmarks. In the future, other geometric parameters, such as torsional angles, will be considered to further boost the molecular representation capacity. We will also study the performance of applying molecular representation to other molecule-related problems, including predicting drug-target interaction and the interactions between molecules.

Appendices

A Details of GeoGNN

A.1 GNN Architecture

As introduced in the main body, GeoGNN consists of two stacks of GeoGNN blocks, one for the atom-bond graph and the other for the bond-angle graph. The architecture of GeoGNN are shown in the bottom of Figure 5, where *Layer Norm* [1], *Graph Size Norm* [7] and *Residual Connection* [21] are popular tricks commonly used in GNNs. *GIN* [55], a convolutional block for message passing, is utilized as the backbone of GeoGNN. In our experiments, the *AGGREGATE* function and *COMBINE* function in *GIN* are defined as

$$
\begin{aligned}
AGGREGATE_{bond\text{-}angle}: a_{uv}^{(k)} = & \sum_{w \in \mathcal{N}(u)} (h_{uv}^{(k-1)} + h_{uw}^{(k-1)} + x_{wuv}) \\
& + \sum_{w \in \mathcal{N}(v)} (h_{uv}^{(k-1)} + h_{vw}^{(k-1)} + x_{uvw}), \\
COMBINE_{bond\text{-}angle}: h_{uv}^{(k)} = & MLP(a_{uv}^{(k)}), \\
AGGREGATE_{atom\text{-}bond}: a_{u}^{(k)} = & \sum_{v \in \mathcal{N}(u)} (h_{u}^{(k-1)} + h_{v}^{(k-1)} + h_{uv}^{(k-1)}), \\
COMBINE_{atom\text{-}bond}: h_{u}^{(k)} = & MLP(a_{u}^{(k)}).
\end{aligned}
$$

where *AGGREGATE* function summarizes the node features and the edge features, while *COMBINE* function is a 2-layer Multi Layer Perceptron (MLP) with hidden size of 32. We use 8 GeoGNN blocks for atom-bond graph and bond-angle graph and the hidden size is set to be 32.

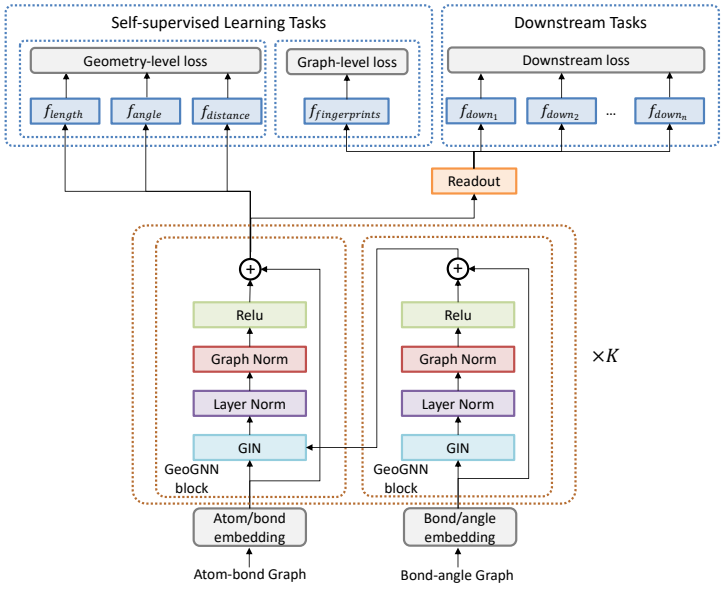


Figure 5: Architecture of GeoGNN.

On top of GeoGNN, different headers are designed for different self-supervised tasks and downstream tasks while all the tasks share the same foundation, i.e., GeoGNN, as illustrated in Figure 5. More specifically, for the geometry-level tasks, the headers are applied upon the node representations $h_u^{(K)}$:

$$
\begin{aligned}
f_{length}(h_u^{(K)}, h_v^{(K)}) = & MLP(Concat(h_u^{(K)}, h_v^{(K)})), \\
f_{angle}(h_u^{(K)}, h_v^{(K)}, h_w^{(K)}) = & MLP(Concat(h_u^{(K)}, h_v^{(K)}, h_w^{(K)})), \\
f_{distance}(h_u^{(K)}, h_v^{(K)}) = & MLP(Concat(h_u^{(K)}, h_v^{(K)})).
\end{aligned}
$$

Table 4: Input features of GeoGNN

Feature type	Feature	Description	Size
atom	atom type	type of atom (e.g., C, N, O), by atomic number (one-hot)	119
	aromaticity	whether the atom is part of an aromatic system (one-hot)	2
	formal charge	electrical charge (one-hot)	16
	chirality tag	CW, CCW, unspecified or other (ont-hot)	4
	degree	number of covalent bonds (one-hot)	11
	number of hydrogens	number of bonded hydrogen atoms (one-hot)	9
	hybridization	sp, sp^2, sp^3, sp^3d, or sp^3d^2 (one-hot)	5
bond	bond dir	begin dash, begin wedge, etc. (one-hot)	7
	bond type	single, double, triple or aromatic (one-hot)	4
	in ring	whether the bond is part of a ring (one-hot)	2
	bond length	bond length (float)	-
bond angle	bond angle	bond angle (float)	-

where *MLP* is a 2-layer MLP network with hidde size of 256 and *Concat* is the concatenation operation. While for the graph-level tasks and downstream tasks, the headers are applied upon the graph representations h_G:

$$f_{\textit{fingerprints}}(h_G) = \textit{Linear}(h_G),$$
$$f_{\textit{down}_n}(h_G) = \textit{MLP}(h_G).$$

where the *MLP* is a 3-layer MLP network with hidden size of 128, and $f_{\textit{down}_1}$ to $f_{\textit{down}_N}$ in Figure 5 represent N different downstream tasks.

A.2 Input Features

As stated in Section 3.1, the input features of GeoGNN can be categorized into three parts: the atom features, bond features, and bond angle features, shown in Table 4. All these features are extracted by RDKit [29] with the help of the Merck molecular force field function. Among them, bond lengths and bond angles are the continuous features, while the others are the discrete features. For the continuous features, we use the Radial Basis Functions [5] to expand each continuous value x into a feature vector e of dimension M:

$$e_m(x) = \exp(-\gamma||x - \mu_m||^2), \tag{8}$$

where γ controls the shape of the radial kernel, and we set $\gamma = 10$. $\{\mu_m\}$ is a list of centers ranging from the minimum value to the maximum value of corresponding features with stride of 0.1. Besides, for the discrete features, they are converted into one-hot vectors according to their vocabulary size.

B Details of Experimental Settings

B.1 Dataset Description

We select 6 molecular regression datasets and 6 molecular classification datasets from MoleculeNet [53] as the benchmarks. These benchmarks can be categorized into the physical chemistry, quantum mechanics, biophysics, and physiology, as shown in Table 5. The benchmarks of physical chemistry and quantum mechanics are more related to molecular geometries compared to those of biophysics and physiology.

The descriptions of various benchmarks are listed as follows:

- `ESOL` [9] contains water solubility data (log solubility in mols per liter) for common organic small molecules. It is a standard dataset that is widely used to estimate solubility directly.
- `FreeSolv` [34] contains the experimental values of free energy of hydration of small molecules in water. The values are all obtained through molecular dynamics simulations.
- `Lipophilicity` [15] is collected from the ChEMBL database [15], containing the experimental results of the octanol or water partition coefficient, which reflects the solubility of the molecules.

Table 5: Dataset details

Task Type	Metric	Category	Dataset	# Tasks	# Compounds
Regression	RMSE	Physical chemistry	ESOL	1	1,128
			FreeSolv	1	642
			Lipophilicity	1	4,200
	MAE	Quantum mechanics	QM7	1	6,830
			QM8	12	21,786
			QM9	12	133,885
Classification	ROC-AUC	Biophysics	BACE	1	1,513
		Physiology	BBBP	1	2,039
			ClinTox	2	1,478
			SIDER	27	1,427
			Tox21	12	7,831
			ToxCast	617	8,575

- `QM7` [3] is a subset of GDB-13, which provides information about the spatial structures of the molecules. It records various electronic properties that are stable and synthetically obtainable, such as HOMO and LUMO determined by *ab-initio* density function theory (DFT), and atomization energy.
- `QM8` [36] uses a variety of quantum mechanics methods to calculate the electronic spectrum and excited state energy of small molecules.
- `QM9`[3] [41] provides multiple data information on geometry, energy, electronic and thermodynamic properties of small molecules calculated by DFT.
- `BACE` [47] records molecules with 2D structures and properties, which provides the qualitative (binary label) binding results on inhibitors of human β-secretase 1.
- `BBBP` [33] is a dataset that contains molecules with measured permeaility property of penetrating the blood-brain barrier.
- `ClinTox` [16] includes drugs approved by the FDA and those that have failed clinical trials for toxicity reasons.
- `SIDER` [28] is a database of marketed drugs and adverse drug reactions (ADR), grouped into 27 system organ classes.
- `Tox21` [25] measures the toxicity of compounds on 12 different targets, including nuclear receptors and stress response pathways. It has been used in the 2014 Tox21 Data Challenge as a public database.
- `ToxCast` [38] includes toxicology results of thousands of molecules. It provides multiple toxicity labels by running high-throughput screening experiments on a large library of chemicals.

B.2 Dataset Splitting Methods

There are several popular splitting methods used in molecular property prediction datasets, including random splitting, scaffold splitting [2] and random scaffold splitting. Scaffold splitting and random scaffold splitting offer more challenging yet realistic ways of splitting by keeping the molecules with the same scaffold in either the train, validation, or the test set. Since the random scaffold splitting introduces more randomness into the evaluation of different methods, we adopt the scaffold splitting in the experiments of our main body. At the same time, we additionally conduct experiments using the random scaffold splitting on the classification datasets following the same experimental settings used in GROVER [40], and again ChemRL-GEM achieves the SOTA results with an overall relative improvement of 2.0% compared to the previous SOTA results on all the datasets, as shown in Table 6. Note that, the results of the baselines are directly copied from [40].

One more thing to mention is that we inspect several public implementations of baselines on scaffold calculation and find that some of them consider the chirality of molecules and some do not. Such

[3]Since the ranges of different targets in QM9 vary largely, we only take the "homo", "lumo", and "gap" targets to calculate the average MAE in our experiments.

Table 6: The performance of classification datasets under random scaffold splitting

Classification (Higher is better)							
Dataset	BACE	BBBP	ClinTox	SIDER	Tox21	ToxCast	Avg
D-MPNN [56]	$0.852_{(0.053)}$	$0.919_{(0.030)}$	$0.897_{(0.040)}$	$0.632_{(0.023)}$	$0.826_{(0.023)}$	$0.718_{(0.011)}$	0.807
AttentiveFP [54]	$0.863_{(0.015)}$	$0.908_{(0.050)}$	$0.933_{(0.020)}$	$0.605_{(0.060)}$	$0.807_{(0.020)}$	$0.579_{(0.001)}$	0.783
N-Gram$_{XGB}$ [32]	$0.876_{(0.035)}$	$0.912_{(0.013)}$	$0.855_{(0.037)}$	$0.632_{(0.005)}$	$0.769_{(0.027)}$	-	-
PretrainGNN [23]	$0.851_{(0.027)}$	$0.915_{(0.040)}$	$0.762_{(0.058)}$	$0.614_{(0.006)}$	$0.811_{(0.015)}$	$0.714_{(0.019)}$	0.778
GROVER$_{base}$ [40]	$0.878_{(0.016)}$	$0.936_{(0.008)}$	$0.925_{(0.013)}$	$0.656_{(0.006)}$	$0.819_{(0.020)}$	$0.723_{(0.010)}$	0.823
GROVER$_{large}$ [40]	$0.894_{(0.028)}$	$0.940_{(0.019)}$	$0.944_{(0.021)}$	$0.658_{(0.023)}$	$0.831_{(0.025)}$	$0.737_{(0.010)}$	0.834
ChemRL-GEM	$\mathbf{0.925_{(0.010)}}$	$\mathbf{0.953_{(0.007)}}$	$\mathbf{0.977_{(0.019)}}$	$\mathbf{0.663_{(0.014)}}$	$\mathbf{0.849_{(0.003)}}$	$\mathbf{0.742_{(0.004)}}$	**0.852**

inconsistence results in a big difference in the splitting train, validation, and test sets. To make the comparison fair, we use the scaffold calculation that considers molecular chirality and apply it to all the experiments. Furthermore, to ensure the data alignment, we summarize the statistics of the labels on each benchmarks. For the regression tasks, we summarize the minimum, maximum, and mean values of the labels, as shown in Figure 7, while for the classification tasks, we summarize the ratios of positive and negative samples as shown in Figure 8. Since there are *Nan* values in Tox21 and ToxCast, the positive ratio and the negative ratio can not be summed up to be 1 in these two datasets.

Table 7: Statistics of label values in 6 regression datasets

Dataset	Train Set			Validation Set			Test Set		
Label value	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
ESOL	-1.16e+1	1.58e+0	-2.87e+0	-9.33e+0	1.10e+0	-3.77e+0	-8.80e+0	1.07e+0	-3.80e+0
FreeSolv	-2.36e+1	3.16e+0	-3.26e+0	-2.55e+1	2.55e+0	-6.05e+0	-1.81e+1	3.43e+0	-5.88e+0
Lipo	-1.50e+0	4.50e+0	2.16e+0	-1.10e+0	4.49e+0	2.20e+0	-1.30e+0	4.50e+0	2.36e+0
QM7	-2.19e+3	-4.05e+2	-1.55e+3	-2.19e+3	-1.00e+3	-1.55e+3	-2.06e+3	-8.26e+2	-1.48e+3
QM8	-3.00e-3	7.11e-1	1.31e-1	-4.50e-7	5.86e-1	1.28e-1	-2.43e-5	5.41e-1	1.32e-1
QM9	-4.29e-1	6.22e-1	7.00e-3	-3.15e-1	3.75e-1	9.00e-3	-3.67e-1	3.88e-1	6.00e-3

Table 8: Ratios of positive and negative samples in 6 classification datasets

Dataset	Train Set		Validation Set		Test Set	
Ratio	Positive	Negative	Positive	Negative	Positive	Negative
BACE	0.3967	0.6033	0.8609	0.1391	0.5329	0.4671
BBBP	0.8406	0.1594	0.3971	0.6029	0.5294	0.4706
ClinTox	0.5068	0.4932	0.5034	0.4966	0.5034	0.4966
SIDER	0.5642	0.4358	0.5918	0.4082	0.5706	0.4294
Tox21	0.0599	0.7880	0.0731	0.6872	0.0717	0.6796
ToxCast	0.0230	0.2766	0.0232	0.2104	0.0324	0.2355

B.3 Pre-training and Downstream Fine-tuning

On the pre-training stage, we utilize the distributed training with 8 GPUs and $batch = 512$ for each GPU. We set the dropout rate as 0.2, the vocabulary size for discretizing atomic distance as 30, and the mask ratio as 0.15. Adam optimizer [26] with learning rate of 0.005 is utilized and we train 20 epochs for each pre-training method.

On the downstream fine-tuning stage, we utilize a single card to train each dataset. Different batch sizes are selected for different datasets: $batch = 256$ for the large-size datasets QM8 and QM9; $batch = 128$ for the medium-size datasets Tox21 and ToxCast; $batch = 32$ for all the other datasets. We use Adam optimizer and train 100 epochs for each model. As the downstream tasks are sensitive to hyper-parameters, we apply a grid search on the dropout rate and the learning rate. For the dropout rate, we search $\{0.1, 0.2, 0.5\}$. For the learning rate, we consider GeoGNN body and the downstream headers separately, where we search body-header learning rate pairs: $\{(0.001, 0.001), (0.004, 0.004), (0.0001, 0.001)\}$.

References

[1] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[2] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[3] Lorenz C Blum and Jean-Louis Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society*, 131(25):8732–8733, 2009.

[4] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.*, 30(7):1145–1159, 1997.

[5] Martin D Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[6] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

[7] Yihao Chen, Xin Tang, Xianbiao Qi, Chun-Guang Li, and Rong Xiao. Learning graph normalization for graph neural networks. *CoRR*, abs/2009.11746, 2020.

[8] Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, 57(8):1757–1772, 2017.

[9] John S. Delaney. ESOL: estimating aqueous solubility directly from molecular structure. *J. Chem. Inf. Model.*, 44(3):1000–1005, 2004.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[11] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015.

[12] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.*, 42(5):1273–1280, 2002.

[13] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2224–2232, 2015.

[14] N Farrell, TG Appleton, Y Qu, JD Roberts, AP Soares Fontes, KA Skov, P Wu, and Y Zou. Effects of geometric isomerism and ligand substitution in bifunctional dinuclear platinum complexes on binding properties and conformational changes in dna. *Biochemistry*, 34(47):15480–15486, 1995.

[15] Anna Gaulton, Louisa J. Bellis, A. Patrícia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.*, 40(Database-Issue):1100–1107, 2012.

[16] Kaitlyn M Gayvert, Neel S Madhukar, and Olivier Elemento. A data-driven approach to predicting successes and failures of clinical trials. *Cell chemical biology*, 23(10):1294–1301, 2016.

[17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations, 2018.

[18] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 2017.

[19] Garrett B Goh, Nathan Hodas, Charles Siegel, and Abhinav Vishnu. Smiles2vec: Predicting chemical properties from text representations. 2018.

[20] Thomas A. Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.*, 17(5-6):490–519, 1996.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[22] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654, 2020.

[23] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[24] Kexin Huang, Tianfan Fu, Cao Xiao, Lucas Glass, and Jimeng Sun. Deeppurpose: a deep learning based drug repurposing toolkit. *CoRR*, abs/2004.08919, 2020.

[25] R Huang, M Xia, DT Nguyen, et al. Editorial: Tox21 challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental toxicants and drugs. front environ sci 5: 3. *Tox21 Challenge to Build Predictive Models of Nuclear Receptor and Stress Response Pathways as Mediated by Exposure to Environmental Toxicants and Drugs*, 5(3):5, 2017.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[27] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[28] Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. The SIDER database of drugs and side effects. *Nucleic Acids Res.*, 44(Database-Issue):1075–1079, 2016.

[29] Greg Landrum et al. Rdkit: Open-source cheminformatics. 2006.

[30] Jintang Li, Kun Xu, Liang Chen, Zibin Zheng, and Xiao Liu. Graphgallery: A platform for fast benchmarking and easy development of graph neural networks based intelligent software. *CoRR*, abs/2102.07933, 2021.

[31] Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guotong Xie, and Sen Song. Learn molecular representations from large-scale unlabeled molecules for drug discovery. *CoRR*, abs/2012.11175, 2020.

[32] Shengchao Liu, Mehmet Furkan Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8464–8476, 2019.

[33] Ines Filipa Martins, Ana L. Teixeira, Luis Pinheiro, and André O. Falcão. A bayesian approach to *in Silico* blood-brain barrier penetration modeling. *J. Chem. Inf. Model.*, 52(6):1686–1697, 2012.

[34] David L. Mobley and J. Peter Guthrie. Freesolv: a database of experimental and calculated hydration free energies, with input files. *J. Comput. Aided Mol. Des.*, 28(7):711–720, 2014.

[35] Tal Peleg-Shulman, Yousef Najajreh, and Dan Gibson. Interactions of cisplatin and transplatin with proteins: Comparison of binding kinetics, binding sites and reactivity of the pt-protein adducts of cisplatin and transplatin towards biological nucleophiles. *Journal of inorganic biochemistry*, 91(1):306–311, 2002.

[36] Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza, and O Anatole Von Lilienfeld. Electronic spectra from tddft and machine learning in chemical space. *The Journal of chemical physics*, 143(8):084111, 2015.

[37] Bharath Ramsundar, Peter Eastman, Patrick Walters, and Vijay Pande. *Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more*. " O'Reilly Media, Inc.", 2019.

[38] Ann M Richard, Richard S Judson, Keith A Houck, Christopher M Grulke, Patra Volarath, Inthirany Thillainadarajah, Chihae Yang, James Rathman, Matthew T Martin, John F Wambaugh, et al. Toxcast chemical landscape: paving the road to 21st century toxicology. *Chemical research in toxicology*, 29(8):1225–1251, 2016.

[39] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, 2010.

[40] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[41] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.*, 52(11):2864–2875, 2012.

[42] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 991–1001, 2017.

[43] Jie Shen and Christos A Nicolaou. Molecular property prediction: recent trends in the era of artificial intelligence. *Drug Discovery Today: Technologies*, 2020.

[44] Hiroyuki Shindo and Yuji Matsumoto. Gated graph recursive neural networks for molecular property prediction. *CoRR*, abs/1909.00259, 2019.

[45] Zeren Shui and George Karypis. Heterogeneous molecular graph neural networks for predicting molecule properties. In Claudia Plant, Haixun Wang, Alfredo Cuzzocrea, Carlo Zaniolo, and Xindong Wu, editors, *20th IEEE International Conference on Data Mining, ICDM 2020, Sorrento, Italy, November 17-20, 2020*, pages 492–500. IEEE, 2020.

[46] Teague Sterling and John J. Irwin. ZINC 15 - ligand discovery for everyone. *J. Chem. Inf. Model.*, 55(11):2324–2337, 2015.

[47] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10):1936–1949, 2016.

[48] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[50] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2017.

[51] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, 1988.

[52] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 2020.

[53] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: A benchmark for molecular machine learning. *CoRR*, abs/1703.00564, 2017.

[54] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 63(16):8749–8760, 2020. PMID: 31408336.

[55] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[56] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2019. PMID: 31361484.

[57] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *CoRR*, abs/1409.2329, 2014.