OXFORD

Briefings in Bioinformatics, 00(0), 2021, 1–14

doi: 10.1093/bib/bbab109
Problem Solving Protocol

An effective self-supervised framework for learning expressive molecular global representations to drug discovery

Pengyong Li, Jun Wang, Yixuan Qiao, Hao Chen, Yihuan Yu, Xiaojun Yao, Peng Gao, Guotong Xie and Sen Song

Corresponding authors: Jun Wang, Ping An Healthcare Technology, Chaoyang, 100027 Beijing, China. Tel.: +86 13011158896; E-mail: junwang.deeplearning@gmail.com; Guotong Xie, Ping An Healthcare Technology, Chaoyang, 100027 Beijing, China. Tel.: +86 021-20665549; E-mail: xieguotong@pingan.com.cn; Sen Song, Tsinghua Laboratory of Brain and Intelligence and Department of Biomedical Engineering, Tsinghua University, Haidian, 100084 Beijing, China. Tel.: +86 010-62773357; E-mail: songsen@mail.tsinghua.edu.cn

Abstract

How to produce expressive molecular representations is a fundamental challenge in artificial intelligence-driven drug discovery. Graph neural network (GNN) has emerged as a powerful technique for modeling molecular data. However, previous supervised approaches usually suffer from the scarcity of labeled data and poor generalization capability. Here, we propose a novel molecular pre-training graph-based deep learning framework, named MPG, that learns molecular representations from large-scale unlabeled molecules. In MPG, we proposed a powerful GNN for modelling molecular graph named MolGNet, and designed an effective self-supervised strategy for pre-training the model at both the node and graph-level. After pre-training on 11 million unlabeled molecules, we revealed that MolGNet can capture valuable chemical insights to produce interpretable representation. The pre-trained MolGNet can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of drug discovery tasks, including molecular properties prediction, drug-drug interaction and drug-target interaction, on 14 benchmark datasets. The pre-trained MolGNet in MPG has the potential to become an advanced molecular encoder in the drug discovery pipeline.

Key words: molecular representation; deep learning; graph neural network; self-supervised learning.

Pengyong Li is a Ph.D. candidate in Department of Biomedical Engineering at Tsinghua University. His research interests focus on graph neural network, drug discovery, artificial intelligence and bioinformatics.
Jun Wang received his Ph.D. from Peking University in 2016. He joined IBM Research China as a Research Scientist during 20162018. Since 2018, he is a senior algorithm researcher in PingAn Healthcare Technology. His research interests include Deep Learning and Drug Discovery.
Yixuan Qiao obtained his Master in Operations Research and Cybernetics at Beijing University of Technology. He works as an algorithm engineer in PingAn Healthcare Technology. His main research interests include natural language processing.
Hao Chen obtained his Master in Operations Research and Cybernetics at Beijing University of Technology. He works as an algorithm engineer in PingAn Healthcare Technology. His main research interests include natural language processing.
Yihuan Yu is a Ph.D. candidate at Beijing University of Biomedical Engineering. He is interested in chemistry.
Professor Xiaojun Yao received his Ph.D. degree in Chemoinformatics and Theoretical Chemistry from University Paris 7-Denis Diderot. He works as a professor of Analytical Chemistry and Chemoinformatics at Lanzhou University. His current research interests include computer-aided molecular design, bioinformatics and computational biology.
Peng Gao received his Ph.D. in Traffic and Transportation Engineering from Tongji University in 2010. Since 2018, he works as vice-chief engineer in PingAn Healthcare Technology, focusing on the *R&D* of distributed AI platform SFE. His research interests include Deep Learning and Smart Healthcare.
Guotong Xie received his Ph.D. from Peking University in 2010. He is now working as PingAn Group Chief Healthcare Scientist and leading the *R&D* of PingAn Healthcare Technology. His research interests include Smart Healthcare, Artificial Intelligence and Deep Learning.
Sen Song received his Ph.D. degree in Biology from Brandeis University. He is an associate professor at Tsinghua University. His current research interests include computational neuroscience, neuroinformatics, artificial intelligence, comparative genomics and bioinformatics.
Submitted: 10 February 2021; **Received (in revised form):** 06 March 2021

Introduction

Drug discovery is a complicated systematic project spanned over 10–15 years [20], which is a long journey for a drug from invention to market in practice. Meanwhile, due to the complexity of biological systems and large number of experiments, drug discovery is prone to failure and inherently expensive [5]. To address these issues, many researchers proposed various computer-aided drug discovery (CADD) methods [50] for small molecule drug design in different stages of early preclinical research from hit identification and selection, hit-to-lead optimization, to clinical candidates [26]. Despite the success in assisting drug discovery, many traditional CADD methods based on molecular simulation techniques suffer from huge computation costs and time-consuming procedures, which limits its application in pharmaceutical industry.

The interdisciplinary studies between artificial intelligence (AI) and drug discovery have received increasing attention due to the superior speed and performance. Many AI technologies have been successfully applied in a variety of tasks for drug discovery, suchas molecular properties prediction [13], drug-drug interaction (DDI) [47] and drug-target interaction (DTI) prediction [2, 11]. One of the fundamental challenges for these studies is how to learn expressive representations from molecular structures [68]. In the early years, molecular representations are based on handcrafted features such as molecular descriptors or fingerprints [67]. Most traditional machine learning methods have revolved around feature engineering for these molecular representations. In contrast, there has been a surge of interest in molecular representation learned by deep neural networks, from fitting raw inputs to the specific task-related targets. Recently, among the promising deep learning architectures, graph neural network (GNN), such as message passing neural network (MPNN) [14] has gradually emerged as a powerful candidate for modeling molecular data. Because a molecule is naturally a graph that consists of atoms (nodes) connected through chemical bonds (edges), it is ideally suited for GNN. Up to now, various GNN architectures have been proposed [14, 16, 28, 56] and achieved great progress in drug discovery [63]. However, there are some limitations that need to be addressed. Challenges for deep learning in molecular representation mainly arise from the scarcity of labeled data, as lab experiments are expensive and time-consuming. Thus, training datasets in drug discovery are usually limited in size, and GNNs tend to overfit them, resulting in learned representations that lack generalizability [22, 46].

One way to alleviate the need for large labeled datasets is to pre-train a model on unlabeled data via self-supervised learning, and then transfer the learned model to downstream tasks [35]. These methods have been widely applied and have made a massive breakthrough in computer vision (CV) and natural language processing (NLP) [1, 17, 29], such as BERT [1]. Some recent works have employed self-supervised learning to pre-train a language model on Simplified Molecular-Input Line-Entry System (SMILES) [60] for learning molecular representation, such as pre-training BERT by regarding SMILES as sequences [8, 21, 41, 59], and pre-training an autoencoder on reconstructing SMILES [15, 62, 66]. Recently, some researchers began to study the pre-training strategies on molecular graph data [22, 34, 46] due to the superior performance of GNN. However, graph data are often more complicated than image and text data because of the variable topological structures, introducing challenges to adopting a self-supervised learning method to the molecular graph directly. Nowadays, some researchers also begin to leverage contrastive learning [35] to empower GNNs to learn the representations for graph data [42, 52, 57], and achieve state-of-the-art performance on unsupervised graph learning. However, most of them, such as InfoGraph [52], usually employ the batch-wise positive/negative samples generation for contrastive discrimination, which bring huge computation costs and are unsuited for pre-training on large-scale datasets, while large-scale dataset is essential for pre-training [1, 4]. Inspired by language model, some simple self-supervised methods for pre-training on large-scale datasets have been proposed, such as N-gram [34], AttrMasking [22], ContextPredict [22] and MotifPredict [46]. However, these methods mainly focus on node-level representation learning and do not explicitly learn a global graph-level representation, resulting in limited gains in graph-level tasks (e.g. molecular classification). Moreover, Hu *et al.* [22] has shown that pre-trained GNNs with pure graph-level or node-level strategy give limited improvements and sometimes lead to negative transfer on many downstream tasks. Thus, it is desirable to develop an efficient graph-level self-supervised strategy.

To address the above issues, we proposed a novel MPG-based deep learning framework, named MPG. In MPG, we first developed a novel GNN that integrates the powerful capacity of MPNN [14] and Transformer [55] to learn molecular representation, called MolGNet. More importantly, we proposed a graph-level self-supervised strategies—Pairwise Half-graph Discrimination (PHD), which is conceptually simple and empirically powerful. We combined PHD with AttrMasking [22] to jointly pre-train our MolGNet model on the node and graph-level. After pre-training MolGNet on 11 million unlabeled molecules, we firstly investigated what our model in MPG learned. We found that the pre-trained MolGNet can capture meaningful patterns of molecules, including molecular scaffold and some quantum properties, to produce interpretable and expressive representation. Moreover, we conducted extensive experiments to evaluate our MPG on a wide range of drug discovery tasks, including molecular properties prediction, DTI and DDI, with 14 widely used datasets. The experimental results show that our MPG advanced the state-of-the-art performances on multiple drug discovery tasks, demonstrating the great capacity and generalizability of MPG. In summary, our MPG learns meaningful and expressive molecular representations from large scale unlabeled molecules, and it lays a foundation for the application of self-supervised learning in drug discovery pipeline.

Methods

The MPG framework

There are two critical aspects to achieving the proposed MPG framework: one is to design a powerful model capable of capturing valuable information from molecular structures; another is to propose an effective self-supervised strategy for pre-training the model. We will introduce the MolGNet model and pre-training strategies in MPG (Figure 1).

MolGNet As shown in Figure 1c, MolGNet is composed of a stack of $N = 5$ identical layers; each layer performs a shared message passing operation for $T = 3$ times recurrently to enable larger receptive fields with less parameters. The message passing operation [14] at each time step t contains a message calculation function $\boldsymbol{M}$ and a vertex update function $\boldsymbol{U}$, where $\boldsymbol{M}$ aggregates the information of neighbors and $\boldsymbol{U}$ uses the aggregated information to update nodes' states. Formally, these two components work sequentially to update the hidden state x_i^t

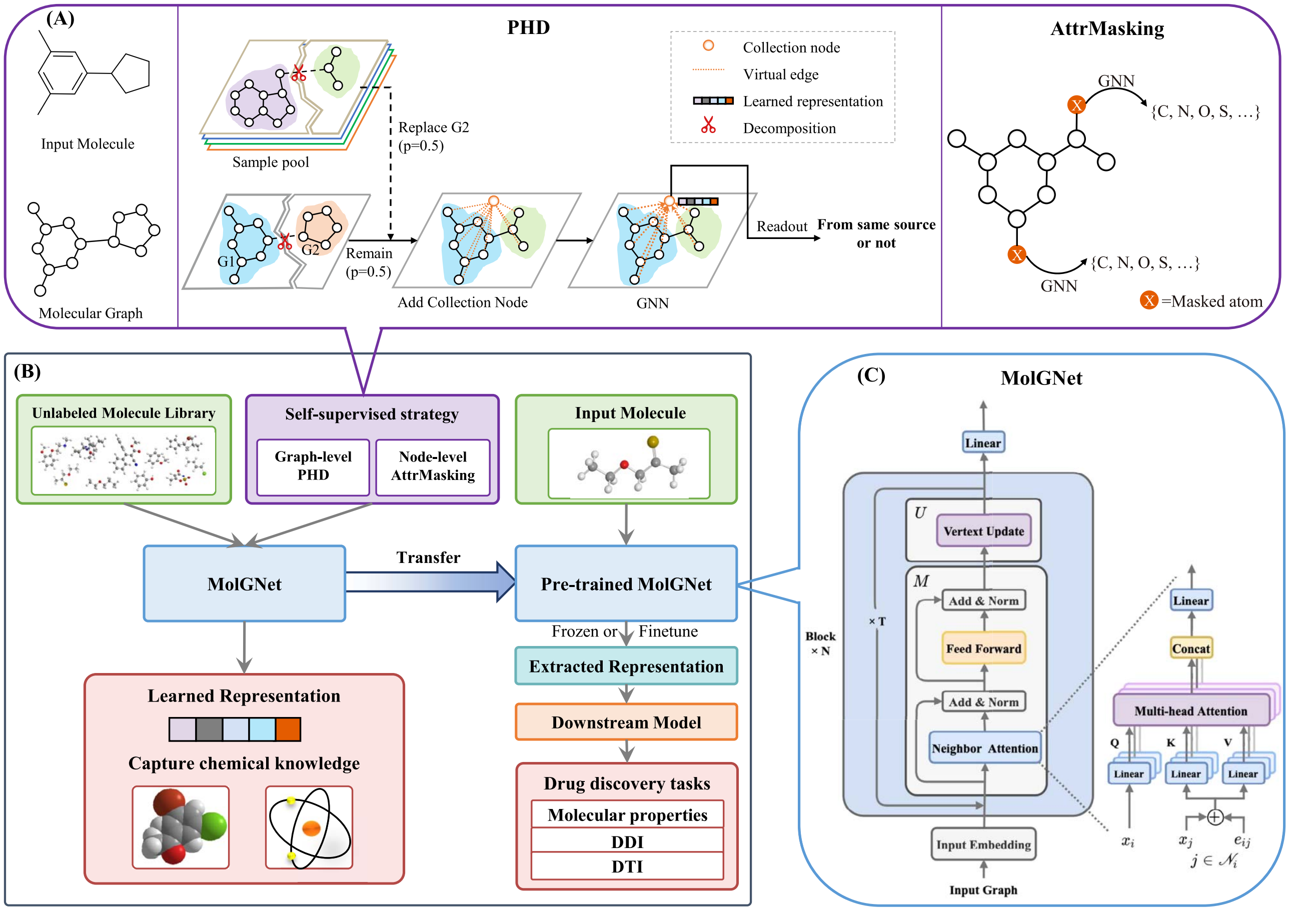


Figure 1. **The overview of the MPG framework.** The bottom left sub-figure (**B**) illustrates the workflow of MPG framework. The MPG framework includes two key components—MolGNet and self-supervised strategies. The architecture of MolGNet is shown in the bottom right sub-figure (**C**). The top sub-figure (**A**) illustrates the schemes of our self-supervised strategy, including PHD we proposed and AttrMasking for pre-training the GNN model.

at each node according to the message passing mechanism. That is

$$m_i^t = \mathbf{M}(\{x_i^{t-1}, x_j^{t-1}, e_{ij}\}, j \in \mathcal{N}_i) \tag{1}$$

$$x_i^t = \mathbf{U}(h_i^{t-1}, m_i^t) \tag{2}$$

where $\mathcal{N}_i$ represents the neighbors of node i, e_{ij} denotes the edge between the node i and node j, vertex update function $\mathbf{U}$ is a gated recurrent unit (GRU) network [9], h_i^{t-1} is the hidden state of $\mathbf{U}$, and h_i^0 is the initial atom representation x_i^0. Specifically, $\mathbf{M}$ has two sub-layers. The first sub-layer conducts the *neighbor attention module* we proposed to extract the information from neighbor nodes and edges, and the second sub-layer is a fully connected feed-forward network. We employ a residual connection around each of the two sub-layers to avoid over-smoothing issues [30, 32], followed by layer normalization. To facilitate these residual connections, all sub-layers in the model produce outputs of dimension $d = 768$. More details about the components of MolGNet can be found in the next section.

Self-supervised strategies Most of the tasks in chemistry (*e.g.* molecular properties prediction) crucially rely on globally inherent molecular characteristics. However, to the best of our knowledge, the current pre-training strategies on large-scale molecular graph mainly focus on node-level representation learning [22, 46]. Here, we proposed a self-supervised pre-training strategy, named PHD, that explicitly pre-trains a GNN at the graph-level. Inspired by contrastive learning [35], the key idea of PHD strategy (Figure 1a) is to learn to compare two half-graphs (each decomposed from a graph sample) and discriminate whether they come from the same source (binary classification). If we assume that the two half-graphs from the same source can be combined into a valid molecule and the two half-graphs from the different source cannot, PHD is to identify the molecular validity by combining two half-graphs, which might teach the network to capture some molecular intrinsic patterns. In particular, we employ a virtual node, called the collection node, to integrate the information of two half-graphs based on the message passing of GNN. The representation of the collection node, serving as the global representation of the given two half-graphs, learns to predict whether two half-graphs are from the same source via maximum likelihood estimation. In order to perform well on the PHD task, it requires the learned collection node representations to encode global information that is capable of discriminating the similarity and dissimilarity between pairs of half-graphs. More details about the implementation of PHD can be found in the PHD strategy section. Moreover, we incorporated our PHD strategy with a recently proposed node-level strategy—AttrMasking [22] for joint pre-training to take full advantage of structural graph information and avoid the negative transfer [22]. Briefly, AttrMasking is designed to predict the masked node's type, as shown in Figure 1a.

In the next section, we firstly introduce the essential components of MolGNet, then we describe the self-supervised strategy—PHD in detail.

MolGNet model

MolGNet consists of three key components: graph attention module, feed forward network and vertex update function. We will elaborate on these three components below.

Neighbor attention module

The input to *neighbor attention module* at time step t is a set of atom representation $\boldsymbol{x} = \{x_1^{t-1}, \cdots, x_N^{t-1}\}, x_i^{t-1} \in \mathbb{R}^d$ and a set of bond representation $\boldsymbol{e} = \{\cdots, e_{i,j}, \cdots\}, e_{i,j} \in \mathbb{R}^d$. The module captures the interaction information between the atom and its neighbors (including its neighbor atoms and neighbor edges) to produce a message representation for each node $\boldsymbol{m} = \{m_1^t, \cdots, m_N^t\}$, $m_i^t \in \mathbb{R}^d$.

For each atom i, the *neighbor attention module* first adds atom i's neighbor atom representation x_j^t with the edge $e_{i,j}$ between them to represent the neighbor information I_j^t, that is:

$$I_j^t = x_j^{t-1} + e_{i,j} \tag{3}$$

Given the neighbor information and atom representation, the module performs scaled dot-product attention [55] on the atoms—a shared attention mechanism computes the attention score. Formally, we first map the node x_i^t into query Q_i^t, and map its neighbor information I_j^t into key K_j^t and value V_j^t, respectively, computed by:

$$Q_i^t = \mathbf{W}_q x_i^{t-1} \tag{4}$$

$$K_j^t = \mathbf{W}_k I_j^t \tag{5}$$

$$V_j^t = \mathbf{W}_v I_j^t \tag{6}$$

where $\mathbf{W}_k$, $\mathbf{W}_q$ and $\mathbf{W}_v$ are the learnable weight matrices shared across all nodes, the dimension of Q_i^t and K_j^t is d_k, and the dimension of V_j^t is d. We compute the dot products of the query and key to indicate the importance of neighbor information to node i. To avoid that the dot products grow large in magnitude, we scale the dot products by $\frac{1}{\sqrt{d_k}}$. That is:

$$s_{i,j}^t = \frac{Q_i^t K_j^{t^T}}{\sqrt{d_k}} \tag{7}$$

To make coefficients easily comparable across different nodes, we then normalize them across all choices of j using the softmax function:

$$a_{i,j}^t = \text{softmax}(s_{i,j}^t) = \frac{e^{s_{i,j}^t}}{\sum_{j\in\mathcal{N}_i} e^{s_{i,j}^t}} \tag{8}$$

where $\mathcal{N}_i$ stands for the neighbors of node i.

Once obtained, the normalized attention coefficients together with neighbor values V_j are used to apply weighted summation operation, to derive the message representation m_i^t for every node:

$$m_i^t = \sum_{j\in\mathcal{N}_i} a_{i,j}^t V_j^t \tag{9}$$

The *neighbor attention module* also employs multi-head attention to stabilize the learning process of self-attention, that is, K independent attention mechanisms execute the transformation of Equation 9, and then their features are concatenated, fed into a linear transformation, resulting in the following output representation:

$$m_i^t = \mathbf{W}_m ||_k^K \sum_{j\in\mathcal{N}_i} a_{i,j}^{t,k} V_j^{t,k} \tag{10}$$

where $\|$ represents concatenation, $a_{i,j}^{t,k}$ is the normalized attention coefficients computed by the kth attention mechanism, $V_j^{t,k}$ is the corresponding neighbor value, W_m is a learnable weight matrix shared across all nodes.

Feed-forward network

To extract a deep representation of the message and increase the expression power of the model, we feed the message representation extracted by *neighbor attention module* into a fully connected feed-forward network. This network consists of two linear transformations with a Gaussian error linear unit (GELU) [19] activation in between.

$$m_i^t = \mathbf{W}_2 \sigma(\mathbf{W}_1 m_i^t + b_1) + b_2 \tag{11}$$

where $W_1 \in \mathbb{R}^{d_{ff}\times d}$ and $W_2 \in \mathbb{R}^{d\times d_{ff}}$ are learnable weight matrices, σ is GELU activation function. In our experiments, the dimension d_{ff} is four times of d, that is 3072 ($d = 768$).

Vertex update function

Based on the properly represented neighbor message m_i^t, our model MolGNet employs a GRU network [9] to update the atom's representation x_i^t, computed by:

$$r_i^t = sigmoid(\mathbf{W}_{mr} m_i^t + b_{mr} + \mathbf{W}_{xr} h_i^{t-1} + b_{hr}) \tag{12}$$

$$u_i^t = sigmoid(\mathbf{W}_{mu} m_i^t + b_{mu} + \mathbf{W}_{xu} h_i^{t-1} + b_{hu}) \tag{13}$$

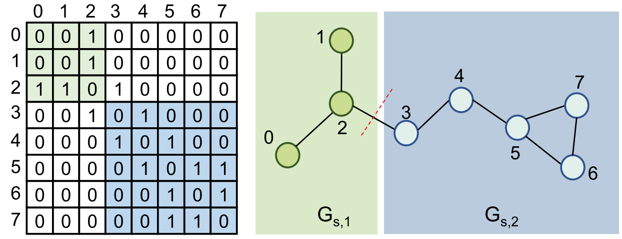
$$x_i^t = tanh(\mathbf{W}_{in} m_i^t + b_{in} + r_i^t * (\mathbf{W}_{hn} h_i^{t-1} + b_{hn})) \tag{14}$$

$$h_i^t = (1 - u_i^t) * x_i^{t-1} + u_i^t * x_i^t \tag{15}$$

where h_i^t is the hidden state of atom i in GRU at time t, h_i^{t-1} is the hidden state at time $t-1$, the initial hidden state h_i^0 is the atom representation x_i^0, and r_i^t and u_i^t are the reset and update gate, respectively. * is the Hadamard product.

PHD strategy

Simply, PHD task is designed to discriminate whether two half-graphs coming from the same source. As shown in Figure 1, the graph is firstly decomposed into two half-graphs, one of these two half-graphs has a 0.5 possibility to be replaced by a half-graph disconnected from another graph which constitutes the



$$\begin{array}{c|cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 7 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

Figure 2. The graph decomposition sample. The left sub-figure is the adjacency matrix of the graph in the right sub-figure, where the green and blue represent the decomposed two half-graphs.

negative sample, otherwise the positive sample. We employed the cross-entropy loss function instead of noise contrastive estimation (NCE) [39] for simple computation to optimize the parameters of the network as follows:

$$L = -\sum_{i=1}^{m} y\log(p) + (1-y)\log(1-p) \tag{16}$$

where m is the number of samples. After pre-training, the collection node embedding can be regarded as a graph-level representation for the graph and used for downstream tasks. In addition, graph representation can also be obtained by averaging the nodes' embeddings or other global graph pooling methods.

In the following sections, we describe the important components of PHD in detail.

Graph decomposition and negative sampling

We decompose the graph into two half-graphs to generate the half-graph pairs, serving as the positive sample, and replace one of the half-graphs to produce the negative sample. As the example shown in Figure 2, given a graph $G = (V, E)$, where V represents nodes and E represents edges. A sampled node v_3 is employed as the border node to separate G into two half-graphs $G_{s,1}$ and $G_{s,2}$, where $G_{s,1}$ contains nodes $\{v_0, v_1, v_2\}$ and $G_{s,2}$ contains nodes $\{v_3, v_4, \cdots, v_7\}$. The edges in these two half-graphs correspond to the top-left sub-matrix and bottom-right sub-matrix of the adjacency matrix respectively. In order to produce half-graphs with balanced and various size, the border node index is randomly sampled in the range of 1/3 to 2/3 of the total number of nodes.

For negative sampling, we randomly sample another graph in the dataset and separate it into two half-graphs using the above method, and $G_{s,2}$ is replaced with one of these two half-graphs to generate a negative sample. How negative samples are generated can have a large impact on the quality of the learned embeddings. It may drive the model to estimate whether the two graphs can be combined into a valid graph. In this way, the model can learn the valuable graph-level features of graphs from the nodes and edges which are essential for the downstream tasks.

Virtual collection node

The half-graph pair obtained via the above approach are two independent graphs without any connection. We concatenate these two half-graphs into a single whole graph, and introduce a virtual collection node to derive the global graph-level representation by aggregating each node's information. The collection node is linked with all the other nodes by virtual directed edges, pointing from the other nodes to the collection node. During the message passing process of GNN, the collection node learns its representation from all the other nodes but does not affect the feature update procedure of them. Consequently, the collection node's feature can grasp the global representation of the half-graphs pair and be fed into a feed-forward neural network for the final prediction.

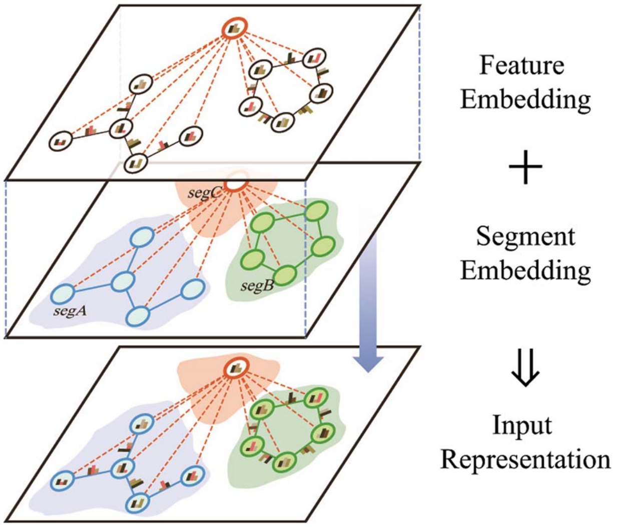


Figure 3. The input representation of graph data is constructed by summing two parts: feature embedding and segment embedding. (**A**) Feature embedding: a set of node and edge features go through the embedding transformation to describe a graph. (**B**) Segment embedding: a learned segmentation embedding to every node and every edge indicating which half-graph it belongs to, different colors represent different segmentation.

Input representation

As shown in Figure 3, the input representation consists of two parts: feature embedding and segment embedding. A graph is generally described by a set of node features and edge features as shown in Table S1. Besides the feature embedding, we add a learned segmentation embedding to every node and every edge indicating which half-graph it belongs to. The final input representation is constructed by summing the segment embedding and feature embedding. In this way, the model could distinguish the nodes and edges from different segments, thus enables simultaneous input of two graphs.

Results

MPG captures meaningful patterns of molecules

To pre-train the MolGNet in MPG, we first constructed a large-scale dataset that contains 11 million molecules from ZINC[51] and ChEMBL[12] datasets. To preserve the diversity, the filtered molecules from ZINC cover a wide range of molecular weight (≥200 Daltons) and LogP (≥1). Each molecule is represented by a set of atom features and a set of bond features (Table S1). We leveraged AttrMasking and our PHD strategies to train the MolGNet jointly. The hyperparameter settings and the learning curves are listed in Figure S1 and Table S2, respectively. After pre-training, we attempted to test whether the pre-trained MolGNet can learn the intrinsic patterns underlying the global molecular characteristic or the interactions between atoms.

To intuitively observe what features the model learned, we visualized the representation extracted by the pre-trained models and tried to explore whether the molecular representations derived from our model can capture chemical knowledge. First, we investigated whether MPG can discriminate the valid molecules from the invalid molecules by their structures, which is the most basic ability for a chemist. The invalid molecular structures conflict with the standard chemical knowledge, such as incorrect valence for atoms. Here, we randomly selected 1000 molecules from the ZINC dataset and disturbed the molecular structures to produce the invalid molecules by shuffling atom features. For each valid and invalid molecule, we extracted the collection node's embedding from the last layer of pre-trained MolGNet as the molecular representation. Once obtained, the representations of both valid and invalid molecules are visualized in the projected 2D space by uniform manifold approximation and projection (UMAP) [38]. We also performed the same analysis on the MolGNet model that was not pre-trained for comparison. As shown in Figure 4a and 4b, non-pre-trained MolGNet shows no obvious cluster, and the molecules overlap without meaningful patterns. After pre-training, the model separated the molecules into two distinct clusters corresponding to valid and invalid molecules (The DB index [10] was decreased from 33.59 to 0.14, indicating a more appropriate separation), demonstrating that the pre-trained model can identify whether the molecule is valid.

Second, we tested whether MPG can encode the scaffold information from molecular structure. The scaffold is an essential concept in chemistry to represent the core structure of a molecule, which provides a basis for systematic investigations of molecular cores and building blocks [3, 23]. Here, we visualized the representation of the molecules with different scaffolds by UMAP. Specifically, we chose 10 most common scaffolds from the ZINC dataset and randomly sampled 1000 molecules for each selected scaffold, resulting in 10000 molecules labeled with 10 different scaffolds. Similarly, the collection node's embedding is regarded as the representation for the molecule. Figure 4c and 4d show the distributions of the representations of molecules produced by the MolGNet with or without the pre-training scheme. Compared with the non-pre-trained MolGNet, the pre-trained MolGNet shows more distinctive clusters corresponding to the 10 molecular scaffolds. It indicates that the pre-trained model is capable of capturing globally inherent molecular characteristics. This capacity may be because our PHD strategy prompts MolGNet to perceive global structural insights or chemical rules, which identifies the scaffolds to accurately discriminate whether two sub-graphs are homologous. The molecules with different scaffolds usually have very different properties, thus our MPG could provide high-quality representations for the downstream tasks. It should be noted that hand-crafted molecular fingerprints [18] such as EFCP and MACCS can also encode scaffold information, but compared to fingerprints, the molecular representation learned by our MPG does not rely on expert knowledge and might encode more valuable and flexible information that remains to be explored (More discussion about molecular representation refers to the Discussion section).

Finally, we conducted a case study to investigate the interpretation of MPG in a more fine-grained way. We colored each atom of selected molecules with the attention weights on the collection node obtained from the last layer of the pre-trained MolGNet. The attention weights represent the contribution of atoms to the global feature. To see whether these attention scores are near related to the critical structural factor of molecules, we plotted the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO) obtained from density functional theory (DFT) calculations for molecules. Surprisingly, we could always find some heads whose attention weights coincided with the areas to which the HOMO and LUMO are distributed (Figure 5). HOMO and LUMO represent the energy required to extract or inject an electron from/to a molecule, respectively, which have crucial effects on the molecular properties, such as redox ability, optical properties and chemical reactivity. In summary, our MolGNet can leverage valuable chemistry knowledge to guide the generation of molecular representations.

Overall, MPG has been proved to be capable of learning interpretable molecular representations that capture some common sense in chemistry, which might bridge the gap between the pre-training and downstream tasks to boost performances.

MPG advances the state-of-the-art in molecular properties prediction

Quantitative structure-activity relationship (QSAR) analysis, aiming at screening large libraries of molecules with desired properties, has emerged as a powerful computational approach in drug discovery [7]. This section comprehensively evaluates our MPG on nine widely used datasets covering various molecular properties, including physical chemistry, biophysics and physiology properties. Details about data sets are referred to Supplementary Information (SI). To offer a fair comparison, we followed the same experimental setting as the previous best method—GROVER [46]. We added a randomly initialized linear classifier on top of the graph-level representations obtained by our pre-trained MolGNet, and fine-tuned the model using the training sets of downstream task datasets. The optimal hyperparameter settings and learning curves are listed in Table S3 and Figure S2, respectively. To compare with molecular SMILES encoding, we also pre-trained a masked language model on SMILES strings named SMILES-BERT according to [59], and fine-tuned this pre-trained model on the datasets above. More details about the SMILE-BERT are referred to SI.

Table 1 summarizes the results that compare MPG with previous self-supervised methods and supervised methods on molecular properties prediction. It shows that our MPG achieves state-of-the-art performance on 7 out of 9 data sets. Compared to molecular fingerprint and supervised neural networks without pre-training, MPG significantly outperforms them on every dataset. In particular, MPG achieved larger gains on small data sets, such as ClinTox, BACE, FreeSolv and ESOL, confirming that our MPG can boost the performance on the tasks with few labeled data. Mol2Vec, N-GRAM and SMILES-BERT are inspired by NLP approches to pre-train a model on the sequential representation. These methods leverage pre-training strategies but failed to explicitly encode the topological structure information of molecules, which greatly harms the performance on the molecular properties prediction. Compared to previous best methods–GROVER, the overall improvement is 13.9% (0.9% on classification tasks and 26.9% on regression tasks). Meanwhile, GROVER contains 100 million parameters, while MolGNet contains 53 million parameters. Better performance with less parameters demonstrates the effectiveness of our MPG. These improvements could be attributed to the self-supervised strategy we proposed. The self-supervised strategy in GROVER only focuses on local structure learning. In contrast, our strategy enables the model to capture more valuable information at both node and graph-level.

Inspired by the impressive performance on molecular properties prediction, we took part in an open task released by MIT J-Clinic recently (https://www.aicures.mit.edu/tasks), aiming at

Table 1. The performance comparison on molecular properties prediction

Methods	Classification (AUC-ROC)						Regression (RMSE)		
Dataset	Tox21	ToxCast	SIDER	ClinTox	BACE	BBBP	FreeSolv	ESOL	Lipo
# Molecules	7831	8575	1427	1478	1513	2039	642	1128	4200
ECFP [45]	$0.760_{(0.009)}$	$0.615_{(0.017)}$	$0.630_{(0.019)}$	$0.673_{(0.031)}$	$0.861_{(0.024)}$	$0.783_{(0.050)}$	$5.275_{(0.751)}$	$2.359_{(0.454)}$	$1.188_{(0.061)}$
TF_Robust [43]	$0.698_{(0.012)}$	$0.585_{(0.031)}$	$0.607_{(0.033)}$	$0.765_{(0.085)}$	$0.824_{(0.022)}$	$0.860_{(0.087)}$	$4.122_{(0.085)}$	$1.722_{(0.038)}$	$0.909_{(0.060)}$
GraphConv [28]	$0.772_{(0.041)}$	$0.650_{(0.025)}$	$0.593_{(0.035)}$	$0.845_{(0.051)}$	$0.854_{(0.011)}$	$0.877_{(0.036)}$	$2.900_{(0.135)}$	$1.068_{(0.050)}$	$0.712_{(0.049)}$
Weave [27]	$0.741_{(0.044)}$	$0.678_{(0.024)}$	$0.543_{(0.034)}$	$0.823_{(0.023)}$	$0.791_{(0.008)}$	$0.837_{(0.065)}$	$2.398_{(0.250)}$	$1.158_{(0.055)}$	$0.813_{(0.042)}$
SchNet [48]	$0.767_{(0.025)}$	$0.679_{(0.021)}$	$0.545_{(0.038)}$	$0.717_{(0.042)}$	$0.750_{(0.033)}$	$0.847_{(0.024)}$	$3.215_{(0.755)}$	$1.045_{(0.064)}$	$0.909_{(0.098)}$
MPNN [14]	$0.808_{(0.024)}$	$0.691_{(0.013)}$	$0.595_{(0.030)}$	$0.879_{(0.054)}$	$0.815_{(0.044)}$	$0.913_{(0.041)}$	$2.185_{(0.952)}$	$1.167_{(0.430)}$	$0.672_{(0.051)}$
DMPNN [68]	$0.826_{(0.023)}$	$0.718_{(0.011)}$	$0.632_{(0.023)}$	$0.897_{(0.040)}$	$0.852_{(0.053)}$	$0.919_{(0.030)}$	$2.177_{(0.914)}$	$0.980_{(0.258)}$	$0.653_{(0.046)}$
MGCN [36]	$0.707_{(0.016)}$	$0.663_{(0.009)}$	$0.552_{(0.018)}$	$0.634_{(0.042)}$	$0.734_{(0.030)}$	$0.850_{(0.064)}$	$3.349_{(0.097)}$	$1.266_{(0.147)}$	$1.113_{(0.041)}$
AttentiveFP [65]	$0.807_{(0.020)}$	$0.579_{(0.001)}$	$0.605_{(0.060)}$	$0.933_{(0.020)}$	$0.863_{(0.015)}$	$0.908_{(0.050)}$	$2.030_{(0.420)}$	$0.853_{(0.060)}$	$0.650_{(0.030)}$
TrimNet [31]	$0.812_{(0.019)}$	$0.652_{(0.032)}$	$0.606_{(0.006)}$	$0.906_{(0.017)}$	$0.843_{(0.025)}$	$0.892_{(0.025)}$	$2.529_{(0.111)}$	$1.282_{(0.029)}$	$0.702_{(0.008)}$
Mol2Vec [25]	$0.805_{(0.015)}$	$0.690_{(0.014)}$	$0.601_{(0.023)}$	$0.828_{(0.023)}$	$0.841_{(0.052)}$	$0.876_{(0.030)}$	$5.752_{(1.245)}$	$2.358_{(0.452)}$	$1.178_{(0.054)}$
N-GRAM [33]	$0.769_{(0.027)}$	-	$0.632_{(0.005)}$	$0.855_{(0.037)}$	$0.876_{(0.035)}$	$0.912_{(0.013)}$	$2.512_{(0.190)}$	$1.100_{(0.160)}$	$0.876_{(0.033)}$
SMILES-BERT [59]	$0.803_{(0.010)}$	$0.655_{(0.010)}$	$0.568_{(0.031)}$	$\mathbf{0.985}_{(0.014)}$	$0.849_{(0.021)}$	$\mathbf{0.959}_{(0.009)}$	$2.974_{(0.510)}$	$0.841_{(0.096)}$	$0.666_{(0.029)}$
HU. et.al. [22]	$0.811_{(0.015)}$	$0.714_{(0.019)}$	$0.614_{(0.006)}$	$0.762_{(0.058)}$	$0.851_{(0.027)}$	$0.915_{(0.040)}$	-	-	-
GROVER [46]	$0.831_{(0.025)}$	$0.737_{(0.010)}$	$0.658_{(0.023)}$	$0.944_{(0.021)}$	$0.894_{(0.028)}$	$0.940_{(0.019)}$	$1.544_{(0.397)}$	$0.831_{(0.120)}$	$0.560_{(0.035)}$
MPG	$\mathbf{0.837}_{(0.019)}$	$\mathbf{0.748}_{(0.005)}$	$\mathbf{0.661}_{(0.007)}$	$0.963_{(0.028)}$	$\mathbf{0.920}_{(0.013)}$	0.922(0.012)	$\mathbf{1.269}_{(0.192)}$	$\mathbf{0.741}_{(0.017)}$	$\mathbf{0.556}_{(0.017)}$

The methods in shading cells are pre-trained methods. We followed the same experimental setting as GROVER [46]. Each dataset was split into train/validation/test set by the random scaffold split with a ratio of 8:1:1.

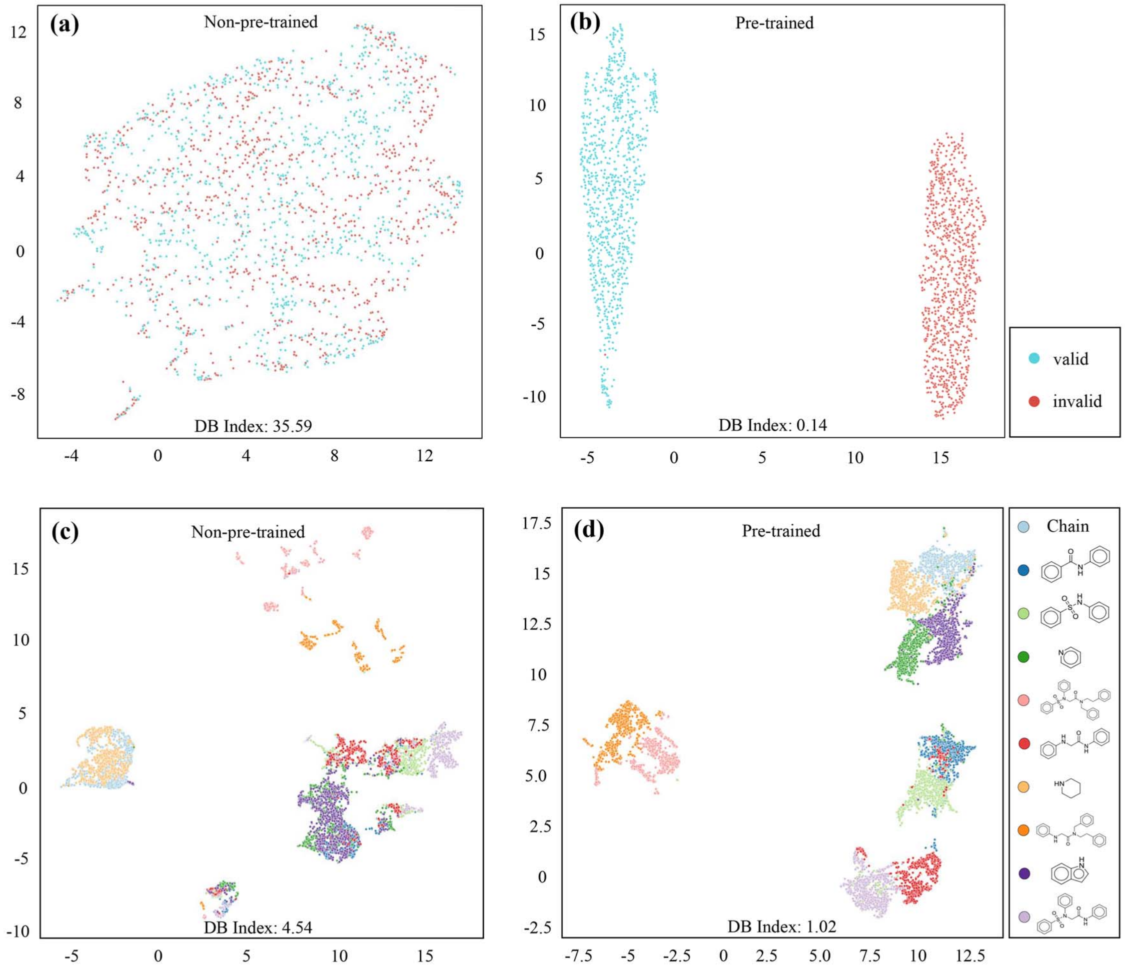


Figure 4. Visualization of the molecular representation by UMAP. The molecular representation is the collection node's embedding extracted from the last layer of non-pre-trained or pre-trained MolGNet. In (**A**) and (**B**), pre-trained MolGNet is capable of distinguishing valid and invalid molecules. In (**C**) and (**D**), the different colors represent different scaffolds the molecules belong to. DB index is Davies Bouldin index [10], a lower DB index means that the clustering has a more appropriate separation.

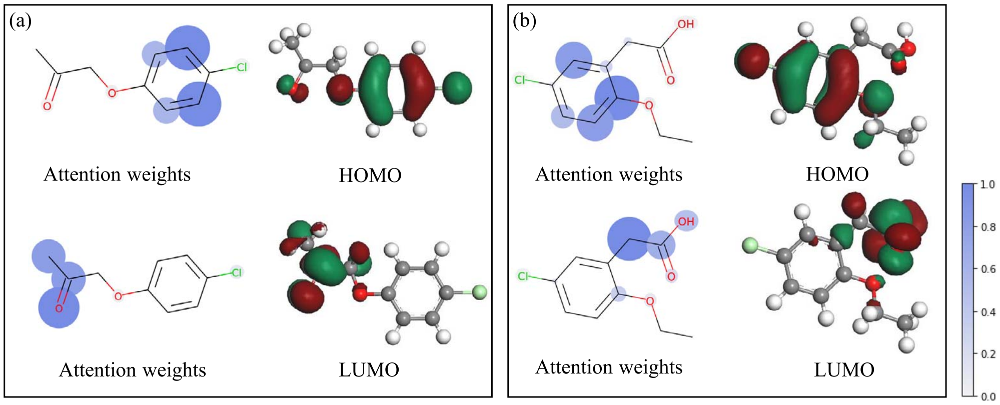


Figure 5. Two example molecules with attention weights coincided with HOMO and LUMO. The attention weights represent the importance of atoms to a molecule's global characteristics, extracted from the last layer of MolGNet and normalized. The larger shading size and deeper color both denote larger attention weight. HOMO/LUMO orbitals are calculated by Materials Studio DMol3.

predicting antibacterial properties of molecules on *Pseudomonas aeruginosa* datasets, for the treatment of secondary infections in patients with COVID-19. Our MPG currently ranks the first with PRC-AUC of 0.725 on this benchmark, outperforming the runner-up with an improvement of 3.3% (Figure 6). It is an inspiring real-world application of MPG, making it possible to find promising drugs for fighting COVID-19 and other emerging pathogens. It can decrease the healthcare burden of secondary

Table 2. MPG provides more accurate DDI prediction than other strong baselines on BIOSNAP dataset

Model	AUC-ROC	PR-AUC	F1
LR	$0.802_{(0.001)}$	$0.779_{(0.001)}$	$0.741_{(0.002)}$
Nat.Prot [58]	$0.853_{(0.001)}$	$0.848_{(0.001)}$	$0.714_{(0.001)}$
Mol2Vec [25]	$0.879_{(0.006)}$	$0.861_{(0.005)}$	$0.798_{(0.007)}$
MolVAE [15]	$0.892_{(0.009)}$	$0.877_{(0.009)}$	$0.788_{(0.033)}$
DeepDDI [47]	$0.886_{(0.007)}$	$0.871_{(0.007)}$	$0.817_{(0.007)}$
CASTER [24]	$0.910_{(0.005)}$	$0.887_{(0.008)}$	$0.843_{(0.005)}$
MPG	$\mathbf{0.966}_{(0.004)}$	$\mathbf{0.960}_{(0.004)}$	$\mathbf{0.905}_{(0.008)}$

The dataset was divided into training/validation/testing sets in a 7:1:2 ratio. The mean and standard deviation of performances run with three random seed are reported. The baselines' performances are taken from CASTER [24].

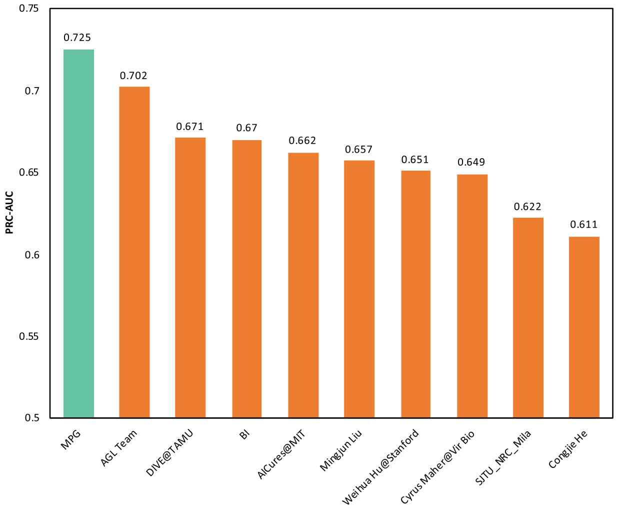


Figure 6. MPG outperforms other methods in the antibacterial properties prediction open task hosted by MIT J-Clinic.

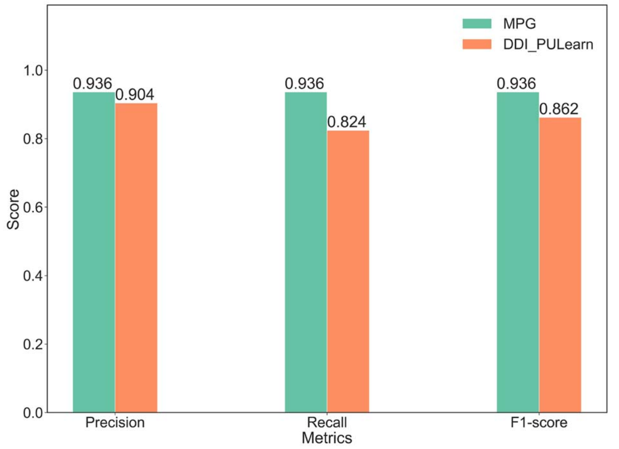


Figure 7. 5-fold cross-validation classification performance on TWOSIDES dataset.

infections and increase the likelihood of survival of critically ill patients with COVID-19.

MPG predicts the drug–drug interaction accurately and rationally

In MPG, we assigned a segmentation embedding to every node and every edge indicating which half-graph it belongs to (details are referred to the Input representation section). This deliberate design endows the model with the capability of taking simultaneously two graphs inputs. In this way, our MPG can be conveniently applied in some tasks with graph pair input, such as commonly used DDI. DDI describes the interactions that one drug may affect others' activities when multiple drugs are administered simultaneously [44]. As the interaction among drugs could trigger an unexpected negative or positive impact on the therapeutic outcomes, characterizing DDI is extremely important for improving drug consumption safety and efficacy. To demonstrate the effectiveness of MPG on DDI prediction, we compared our framework against the recently proposed algorithms on two real-world datasets—BIOSNAP[37] and TWOSIDES[53] (Details about both datasets are referred to SI). To ensure a fair comparison, we followed the identical experimental procedure of two best approaches—CASTER [24] and DDI-PULearn [69], on above two datasets, respectively. The DDI prediction tasks are formalized as a binary classification problem that aims to identify an interaction between two drugs. The classification results are reported in Table 2 and Figure 7.

Table 2 and Figure 7 show that MPG significantly outperforms the previous best methods (CASTER and DDI-PULearn) on both two datasets by a large margin (7% and 9% improvements in terms of F1 score, respectively). CASTER takes SMILES [60] substrings as inputs to represent molecular sub-structures. Compared to SMILES, a hydrogen-depleted molecular graph is more suitable and effective to represent molecules' structural information [64]. DDI-PULearn [69] collected various drug properties to calculate the drugdrug similarities as input representation, including drug chemical substructures, drug targets, side-effects and drug indications. In contrast, our MPG only takes the molecular structure as inputs, and we observed that MPG still yielded significantly better performance than DDI-PULearn. These results demonstrate the prediction superior performance of MPG on DDI prediction.

Furthermore, MPG can generate an interpretable prediction. Given an input drug pair, MPG assigns an attention weight to each atom in molecules, indicating the importance of the interaction. We chose the interaction between Sildenafil and other nitrate-based drugs as a case study. Sildenafil, a PDE5 inhibitor, is developed as an effective treatment for pulmonary hypertension and erectile dysfunction. Because nitrate-based drugs and Sildenafil increase cGMP (nitrates increase cGMP formation and Sildenafil decrease cGMP breakdown), it could lead to sudden drops in blood pressure and even heart attack when used in combination. Thus, we tested if our MPG can pay more attention to the nitrate group when it predicts the interaction

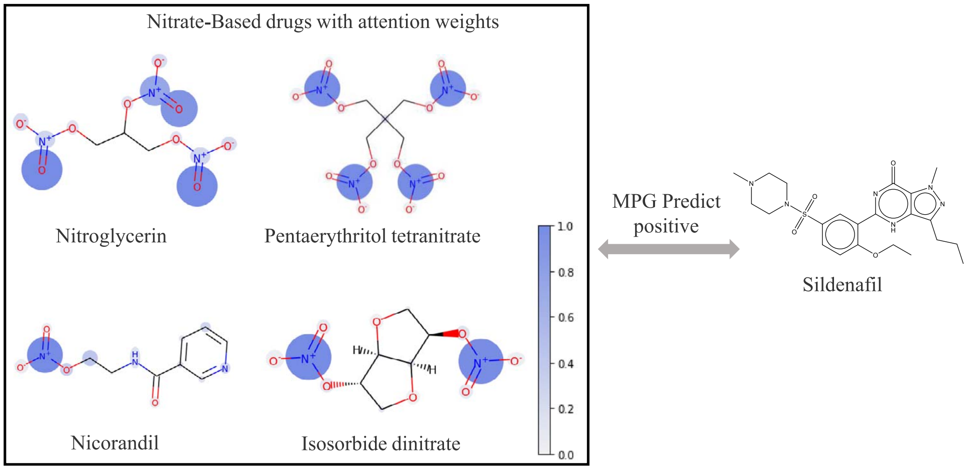


Figure 8. MPG provides explainability for DDI prediction. The left part of the figure illustrates nitrate-based drugs with each atom colored by the attention weight. Note that our MPG always pays more attention to the nitrate group.

Table 3. MPG boosts classification performances for DTI prediction on Human (a) and *Caenorhabditis elegans* (b) datasets

Datasets	Methods	Precision	Recall	AUC
Human	Tsubaki *et al.* [54]	92.3	91.8	97.0
	MPG	**95.2**	**94.0**	**98.5**
	(Improvement)	(3.1%)	(2.4%)	(1.6%)
C. elegans	Tsubaki *et al.* [54]	93.8	92.9	97.8
	MPG	**95.4**	**95.9**	**98.6**
	(Improvement)	(1.7%)	(3.2%)	(0.8%)

between Sildenafil and other nitrate-based drugs. Specifically, we extracted and normalized the atom's attention weights to the collection nodes from the last layer of MolGNet. After visualizing the attention weights, we observed that there always existed high attention weights on the nitrate group (Figure 8). This suggests that MPG could leverage sparse and reasonable information of molecules to generate DDI prediction.

MPG boosts the performance of drug-target interaction prediction

As experiments above show that MPG achieves impressive performance on ligand-based CADD tasks, we further explored MPG's capacity on structure-based CADD. Structure-based CADD aims to identify the interaction between the compound and target protein for drug discovery. Various deep learning methods have been developed and achieved excellent performance for DTI prediction [6, 40, 61]. Generally, the deep learning algorithms for DTI prediction comprise of a compound encoder and a protein encoder. Recently, Tsubaki *et al.* [54] proposed a new DTI prediction framework by combining a GNN for compounds and a CNN for proteins, which significantly outperformed existing methods. Here, we adapted their framework and replaced their compound encoder with our pre-trained MolGNet (as shown in Figure S3) to evaluate its effectiveness on DTI prediction. We followed the same experimental procedure as Tsubaki *et al.* [54] to ensure a fair comparison on two widely used datasets—Human and *C.elegans* datasets. Table 3 shows that replacing the compound encoder with our MPG significantly improves the performance on both datasets, re-confirming MPG's powerful capacity for modeling molecules.

Ablation studies

The effect of pre-training

To verify the necessity of pre-training in MPG, we compared the performances of pre-trained and non-pre-trained MolGNet on molecular properties prediction tasks, both of which have the identical hyper-parameter setting. Table 4 shows that, compared with the pre-trained MolGNet, the MolGNet without pre-training demonstrates significant decreases in classification AUC-ROC score, and increases in RMSE of regression tasks, which confirmed that our self-supervised strategies could provide a favorable initialization for the model and improve the performance of downstream tasks. Notably, compared to previous methods without pre-training in Table 1, our MolGNet achieved best performances on ToxCast, BACE, FreeSolv and Lipo datasets and comparable performance on the rest. Besides, Table 4 indicates that the smaller datasets, including BBBP, SIDER, ClinTox, BACE and FreeSolv, had a greater performance gain through pre-training, demonstrating the effectiveness and generalizability of the self-supervised pre-training for tasks with insufficient labeled molecules.

The effect of PHD strategy

In the pre-training process of MPG, we employed AttrMasking and PHD strategies to jointly pre-train MolGNet. To investigate the contributions of these two strategies, we pre-trained our model with AttrMasking or with PHD separately to compare their performances on downstream classification tasks. These self-supervised strategies follow the same hyper-parameter setting. Table 4 shows that both strategies can improve the average AUC-ROC score compared with no pre-training. In the meantime, our PHD strategy outperforms AttrMasking on 8 out of 9 data sets,

Table 4. Effect of pre-training strategies

Datasets			Tox21	ToxCast	SIDER	ClinTox	BACE	BBBP	FreeSolv	ESOL	Lipo
Molecules			7831	8575	1427	1478	1513	2039	642	1128	4200
	AttrMasking	PHD	Classification (AUC-ROC %)						Regression (RMSE)		
MPG (no pre-train)	-	-	80.1(1.2)	69.9(1.6)	58.5(1.5)	92.4 (3.4)	86.8(1.4)	89.2(0.8)	1.967(0.556)	0.896(0.145)	0.628(0.062)
MPG (node-level)	✓	-	81.9(0.8)	72.6(0.7)	61.1(0.8)	93.5(2.3)	87.7(1.4)	90.2(1.5)	1.829(0.172)	0.835(0.192)	0.710(0.049)
MPG (graph-level)	-	✓	83.4(1.0)	72.2(1.0)	62.2(0.7)	95.1 (1.5)	88.4(0.8)	91.1(0.8)	1.464(0.196)	0.814(0.067)	0.608(0.021)
MPG	✓	✓	**83.7(1.9)**	**74.8(0.5)**	**65.8(1.2)**	**96.3(2.8)**	**92.0(1.3)**	**92.2(1.2)**	**1.269(0.192)**	**0.802(0.043)**	**0.576(0.029)**

which indicates the importance and superiority of graph-level self-supervised learning for molecular properties prediction. It should be noted that combining these two strategies for pre-training yields a greater improvement than pre-training through either of the two strategies.

Discussion

Molecular representations. Molecular representations can be generally categorized into handcrafted representations and learned representations. Fingerprint [18] and SMILES [60] are two widely used handcrafted representations. The most common type of fingerprint is a series of binary digits (bits) representing the presence or absence of particular substructures in the molecule. Although molecular fingerprint features in its flexibility and ease of computation for reaction prediction [49], it also gives rise to several issues, including bit collisions and vector sparsity. Besides, molecules can be encoded as SMILES [60] in the format of single-line text. Nevertheless, a key weakness in representing molecules using text sequences is its fragility of the representation, since small changes in the text sequence can lead to a large change in the molecular structure. Compared to the handcrafted representations, the learned molecular representation by deep learning has better generalization and higher expressive power, but it usually lacks explainability. That is, we have no idea about how the representation is generated and what the representation stands for. This study makes an attempt to investigate the explainability of molecular representation, and found that our MPG can capture some chemical knowledge. Further theoretical and empirical analysis are needed to better understand when/why/how pre-training for GNNs can work.

Self-supervised strategies. Self-supervised strategies have crucial impact on performance of pre-trained model. Current self-supervised strategies for pre-training GNNs suffer from either high computational complexity or falling into node-level learning, which are time-consuming and ineffective when applied in large-scale molecule pre-training. Here, we applied three main principles for designing an appropriate self-supervised strategy to pre-train on molecule—computation friendliness, architecture-free, learning on both node and graph-level. First, our strategy is simple which enables the model to pre-train on large scale data to encode more information. Second, the strategy is independent of the model, as we may evaluate different models to select the optimal architecture. Last, our strategy can pre-train the model at both node-level and graph-level to encode more information on structural characteristic. This work serves as an important first step towards the graph-level self-supervised learning on large scale molecule data. Although we focused on molecular representation for drug discovery, the approach presented in this work is more general, and can be adapted to graph representation learning for other areas, such as social networks and knowledge graph.

Key Points

- We proposed a framework, named MPG, where we developed a deep graph neural network—MolGNet and a graph-level self-supervised strategy—PHD for pre-training MolGNet.
- The molecular representation learned by MPG can capture meaningful patterns of molecules, including validity, scaffold and some quantum properties.

- MPG can significantly outperform current state-of-the-art methods on multiple molecular representation learning tasks, including molecular property predictions(quantum properties, bioactivity, physiology), drugdrug interaction and drugtarget interaction identification.

Data availability

The pre-training datasets are available on the ZINC (http://zinc15.docking.org/) and ChemBL (https://www.ebi.ac.uk/chembl/). The molecular properties data that support the findings of this study are available on the website of MoleculeNet: http://moleculenet.ai. The DDI data sets including BIOSNAP and TWOSIDES are available at CASTER repository (https://github.com/kexinhuang12345/CASTER) and DDI-PULearn additional files (https://drive.google.com/drive/folders/1wKnY4L4iAjBdTMcJBewYNqCgUQ15DXmY?usp=sharing). The DTI data sets including Human and *C. elegans* are available at https://github.com/masashitsubaki/CPI_prediction.

Code Availability

The source code is available on GitHub: https://github.com/pyli0628/MPG.git

Author Contribution

P.L. conceived the research project. S.S., J.W. and G.X. supervised the research project. P.L. designed and implemented the MPG framework. P.L., J.W., Y.Q. and H.C. conducted the computational analyses. Y.Y. calculated the HOMO/LUMO orbitals. P.L., J.W., S.S., Y.Q., X.Y. and H.C. wrote the manuscript. All the authors discussed the experimental results and commented on the manuscript.

Acknowledgments

The authors acknowledge the anonymous reviewers for reviewing the manuscript.

Funding

This work was supported in part by funds from Department of Education Key Innovation Research Grant, Beijing innovation center for future chips, Institute Guoqiang at Tsinghua University, the National Natural Science Foundation of China (61836004), and Beijing Brain Science Special Project (No. Z181100001518006).

Competing Interests statement

The authors declare no competing interests.

References

1. Devlin J, Chang M-W, Lee K, *et al.* Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
2. Abbasi K, Razzaghi P, Poso A, *et al.* Deep learning in drug target interaction prediction: Current and future perspective. *Curr Med Chem* 2020.
3. Bemis GW, Murcko MA. The properties of known drugs. 1. *molecular frameworks. J Med Chem* 1996; **39**(15): 2887–93.
4. Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, *et al.* Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.
5. Chan HCS, Shan H, Dahoun T, *et al.* Advancing drug discovery via artificial intelligence. *Trends in pharmacological sciences*. 2019; **40**(8): 592–604.
6. Chen R, Liu X, Jin S, *et al.* Machine learning for drug-target interaction prediction. *Molecules* 2018; **23**(9): 2208.
7. Artem Cherkasov EN, Muratov DF, Alexandre Varnek II, *et al.* Qsar modeling: where have you been? where are you going to? *J Med Chem* 2014; **57**(12): 4977–5010.
8. Seyone Chithrananda, Gabe Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. arXiv preprint arXiv:2010.09885, 2020.
9. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.
10. Davies DL, Bouldin DW. A cluster separation measure. *IEEE Trans Pattern Anal Mach Intell* 1979; 224–7.
11. Sofia D'S, Prema KV, Balaji S. Machine learning models for drug–target interactions: current knowledge and future directions. *Drug Discov Today* 2020; **25**(4): 748–56.
12. Gaulton A, Bellis LJ, Bento AP, *et al.* ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res* 2011; **40**(D1): D1100–7.
13. Ghasemi F, Mehridehnavi A, Perez-Garrido A, *et al.* Neural network and deep-learning algorithms used in qsar studies: merits and drawbacks. *Drug Discov Today* 2018; **23**(10): 1784–90.
14. Gilmer J, Schoenholz SS, Riley P, *et al.* Neural message passing for quantum chemistry. *International Conference on Machine Learning*. Sydney, NSW, Australia: ICML, 2017, 1263–72.
15. Gómez-Bombarelli R, Wei JN, Duvenaud D, *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science* 2018; **4**(2): 268–76.
16. William L, Hamilton RY, Leskovec J. Inductive representation learning on large graphs. arXiv preprint arXiv:1706.02216, 2017.
17. He K, Fan H, Wu Y, *et al.* Momentum contrast for unsupervised visual representation learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle, WA, USA: IEEE, 2020, 9729–38.
18. Heinonen M, Shen H, Zamboni N, *et al.* Metabolite identification and molecular fingerprint prediction through machine learning. *Bioinformatics* 2012; **28**(18): 2333–41.
19. Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.
20. Hill RG. Drug discovery and development-E-book: technology in transition. *Elsevier Health Sciences* 2012.
21. Shion Honda, Shoi Shi, and Hiroki R Ueda. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. arXiv preprint arXiv:1911.04738, 2019.

22. Hu W, Liu B, Gomes J, *et al.* Strategies for pre-training graph neural networks. *In International Conference on Learning Representations*. New Orleans, LA, USA: ICLR publisher, 2019.
23. Hu Y, Stumpfe D, Bajorath J. Computational exploration of molecular scaffolds in medicinal chemistry: Miniperspective. *J Med Chem* 2016; **59**(9): 4062–76.
24. Huang K, Xiao C, Hoang T, *et al.* Caster: Predicting drug interactions with chemical substructure representation. *Proceedings of the AAAI Conference on Artificial Intelligence*. New York City, NY, USA: AAAI publisher, 2020, **34**:702–9.
25. Jaeger S, Fulle S, Turk S. Mol2vec: unsupervised machine learning approach with chemical intuition. *J Chem Inf Model* 2018; **58**(1): 27–35.
26. Kapetanovic IM. Computer-aided drug discovery and development (caddd): in silico-chemico-biological approach. *Chem Biol Interact* 2008; **171**(2): 165–76.
27. Kearnes S, McCloskey K, Berndl M, *et al.* Molecular graph convolutions: moving beyond fingerprints. *J Comput Aided Mol Des* 2016; **30**(8): 595–608.
28. Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.
29. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. Nevada, USA: Neural Information Processing Systems Foundation, 2012, **25**:1097–105.
30. Li G, Muller M, Thabet A, *et al.* Deepgcns: Can gcns go as deep as cnns? In: *Proceedings of the IEEE International Conference on Computer Vision*. Seoul, South Korea: IEEE, 2019, 9267–76.
31. Li P, Li Y, Hsieh C-Y, *et al.* Trimnet: learning molecular representation from triplet messages for biomedicine. *Brief Bioinform* 2020.
32. Liu M, Gao H, Ji S. Towards deeper graph neural networks. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. San Diego, CA, USA: ACM, 2020, 338–48.
33. Liu S, Demirel MF, Liang Y. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. In: *Advances in Neural Information Processing Systems*. Vancouver, BC, Canada: Neural Information Processing Systems Foundation, 2019, 8466–78.
34. Shengchao Liu, Mehmet Furkan Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. arXiv preprint arXiv:1806.09206, 2018.
35. Liu X, Zhang F, Hou Z, *et al.* Self-supervised learning. In: *Generative or contrastive*. arXiv preprint arXiv:2006.08218, 2020.
36. Lu C, Liu Q, Wang C, *et al.* Molecular property prediction: A multilevel quantum interactions modeling perspective. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. Honolulu, Hawaii, USA: AAAI Press, 2019, 1051–60.
37. Sagar Maheshwari Marinka Zitnik, Rok Sosič and Jure Leskovec. BioSNAP Datasets: Stanford biomedical network dataset collection. http://snap.stanford.edu/biodata, August 2018.
38. Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction arXiv preprint arXiv:1802.03426, 2018.
39. Mnih A, Kavukcuoglu K. Learning word embeddings efficiently with noise-contrastive estimation. *In Advances in Neural Information Processing Systems*. Lake Tahoe, Nevada, USA: Neural Information Processing Systems Foundation, 2013, 2265–73.
40. Mousavian Z, Masoudi-Nejad A. Drug–target interaction prediction via chemogenomic space: learning-based methods. *Expert Opin Drug Metab Toxicol* 2014; **10**(9): 1273–87.
41. Pesciullesi G, Schwaller P, Laino T, *et al.* Transfer learning enables the molecular transformer to predict regio- and stereoselective reactions on carbohydrates. *Nat Commun* 2020; **11**(1): 1–8.
42. Qiu J, Chen Q, Dong Y, *et al.* Graph contrastive coding for graph neural network pre-training. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. San Diego, CA: ACM, 2020, 1150–60.
43. Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. arXiv preprint arXiv:1502.02072, 2015.
44. David Rodrigues A. *Drug-drug interactions*. CRC Press, 2019.
45. Rogers D, Hahn M. Extended-connectivity fingerprints. *J Chem Inf Model* 2010; **50**(5): 742–54.
46. Yu R, Bian Y, Xu T, *et al.* Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*. Vancouver, BC, Canada: Neural Information Processing Systems Foundation, 2020, 33.
47. Ryu JY, Kim HU, Lee SY. Deep learning improves prediction of drug–drug and drug–food interactions. *Proc Natl Acad Sci* 2018; **115**(18): E4304–11.
48. Schütt K, Kindermans P-J, Felix HES, *et al.* Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In: *Advances in Neural Information Processing Systems*. Long Beach, CA: Neural Information Processing Systems Foundation, 2017, 991–1001.
49. Segler MHS, Waller MP. Modelling chemical reasoning to predict and invent reactions. *Chem* 2017; **23**(25): 6118–28.
50. Sliwoski G, Kothiwale S, Meiler J, *et al.* Computational methods in drug discovery. *Pharmacol Rev* 2014; **66**(1): 334–95.
51. Sterling T, Irwin JJ. Zinc 15-ligand discovery for everyone. *J Chem Inf Model* 2015; **55**(11): 2324–37.
52. Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000, 2019.
53. Nicholas P, Tatonetti P, Patrick Y, *et al.* Data-driven prediction of drug effects and interactions. *Science Translational Medicine* 2012; **4**(125): 125ra31–1.
54. Tsubaki M, Tomii K, Sese J. Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics* 2019; **35**(2): 309–18.
55. Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. In: *Advances in Neural Information Processing Systems*. Long Beach, CA: Neural Information Processing Systems Foundation, 2017, 5998–6008.
56. Veličković P, Cucurull G, Casanova A, *et al.* Graph Attention Networks. In: *International Conference on Learning Representations*. Toulon, France: ICLR publisher, 2018.
57. Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. arXiv preprint arXiv:1809.10341, 2018.
58. Vilar S, Uriarte E, Santana L, *et al.* Similarity-based modeling in large-scale prediction of drug-drug interactions. *Nat Protoc* 2014; **9**(9): 2147.
59. Wang S, Guo Y, Wang Y, *et al.* Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In: *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. Niagara Falls, NY, USA: ACM, 2019, 429–36.

60. Weininger D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 1988; **28**(1): 31–6.
61. Wen M, Zhang Z, Niu S, *et al*. Deep-learning-based drug–target interaction prediction. *J Proteome Res* 2017; **16**(4): 1401–9.
62. Winter R, Montanari F, Noé F, *et al*. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem Sci* 2019; **10**(6): 1692–701.
63. Wu Z, Ramsundar B, Feinberg EN, *et al*. MoleculeNet: A benchmark for molecular machine learning. *Chem Sci* 2018; **9**(2): 513–30.
64. Wu Z, Ramsundar B, Feinberg EN, *et al*. Moleculenet: a benchmark for molecular machine learning. *Chem Sci* 2018; **9**(2): 513–30.
65. Xiong Z, Wang D, Liu X, *et al*. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J Med Chem* 2019; **63**(16): 8749–60.
66. Zheng X, Wang S, Zhu F, *et al*. An unsupervised deep molecular embedding for drug discovery. In: *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*. Boston, MA, USA: ACM, 2017, 285–94.
67. Xue L, Bajorath J. Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. *Comb Chem High Throughput Screen* 2000; **3**(5): 363–72.
68. Yang K, Swanson K, Jin W, *et al*. Analyzing learned molecular representations for property prediction. *J Chem Inf Model* 2019; **59**(8): 3370–88.
69. Zheng Y, Peng H, Zhang X, *et al*. Ddi-pulearn: a positive-unlabeled learning method for large-scale prediction of drug-drug interactions. *BMC bioinformatics* 2019; **20**(19): 1–12.