JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

Cite This: J. Chem. Inf. Model. 2019, 59, 3817−3828

pubs.acs.org/jcim

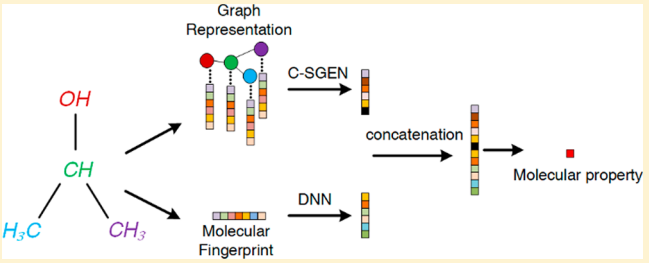
Molecule Property Prediction Based on Spatial Graph Embedding

Xiaofeng Wang, Zhen Li,* Mingjian Jiang, Shuang Wang, Shugang Zhang, and Zhiqiang Wei

College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China

Supporting Information

ABSTRACT: Accurate prediction of molecular properties is important for new compound design, which is a crucial step in drug discovery. In this paper, molecular graph data is utilized for property prediction based on graph convolution neural networks. In addition, a convolution spatial graph embedding layer (C-SGEL) is introduced to retain the spatial connection information on molecules. And, multiple C-SGELs are stacked to construct a convolution spatial graph embedding network (C-SGEN) for end-to-end representation learning. In order to enhance the robustness of the network, molecular fingerprints are also combined with C-SGEN to build a composite model for predicting molecular properties. Our comparative experiments have shown that our method is accurate and achieves the best results on some open benchmark datasets.



1. INTRODUCTION

The prediction of molecular properties is a crucial step in drug discovery. Traditionally, it requires a lot of complicated biochemical time-consuming experiments to obtain the properties of a molecule. Moreover, with the continuous emergence of new molecules, the work of molecule property prediction is heavy and endless. With the rise of computer-aided drug design, using computers to predict molecule properties has become a new trend in the field of bioinformatics. Although the processing speed of the computer is faster, the accuracy of predictions still needs to be improved. Therefore, an accurate algorithm is essential to predict the intrinsic chemistry properties of molecules, which is helpful to save experimental costs and shorten the time of drug discovery.

There are many methods with which to research molecular property prediction. At early stages, molecular descriptors are extracted directly through molecular structures and the properties of molecules can be predicted by a reliable quantitative structure−activity relationship (QSAR) model.[1,2] The application of machine learning on QSAR models has shown good performance in chemical property prediction,[3−6] and the commonly used machine learning models include support vector machines (SVM),[7] random forest (RF),[8] and artificial neural networks (ANN).[9]

In the past few years, deep learning has become a popular method and been applied in molecular fields. Mayr et al. developed the deep neural network (DNN) model to process predefined molecular descriptors, and it outperformed previous machine learning methods in prediction of molecular properties.[10,11] Moreover, the superiority of deep learning over other methods is summed by Mamoshina et al.,[12] deep learning can extract potential features from original features, which owns a broader prospect than machine learning for biomarker development and drug discovery. However, similar to machine learning methods, existing deep learning methods rely excessively on hand-craft feature design of molecule. Since input features are predefined, it may limit the model's search space of potential representations, which will then seriously reduce the performance of the model if the predefined molecular descriptors are not selected correctly.

In order to avoid the drawbacks of predefined features, it may be a good choice to use convolutional neural networks (CNNs) to extract molecule features. Recently, unique advantages in computer vision (CV),[13] natural language processing (NLP),[14] and computational chemistry have been demonstrated.[15] A 3D-convolutional neural network is proposed to predict molecular properties, and regular fixed-size grids are established to represent molecules,[16] which is similar to the pixels in the image, each voxel in the grid calculates atomic features. Imrie[17] discretized the protein−ligand complex into a three-dimensional grid and constructed a scoring function by using a CNN and transfer learning method for virtual screening. This research shows that CNN could automatically obtain data-driven features from regular grid data directly and avoid hand-crafted feature extraction.[18] However, molecular data is a type of graph data, in which each atom represents the node in the graph, so grid data cannot depict the connection between each atom.

Duvenaud et al. referred to the circular fingerprint method, and they designed a convolution neural network on the molecular graph directly to obtain the differentiable neural graph fingerprints.[19] As an extension of CNN, the graph convolution neural network (GCN) is designed to process such non-Euclidean structural data and has attracted great attention, especially in drug discovery. Several research works have applied GCN directly on the molecular graph structure and achieved state-of-the-art performance on some molecular

Received: May 15, 2019
Published: August 22, 2019

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

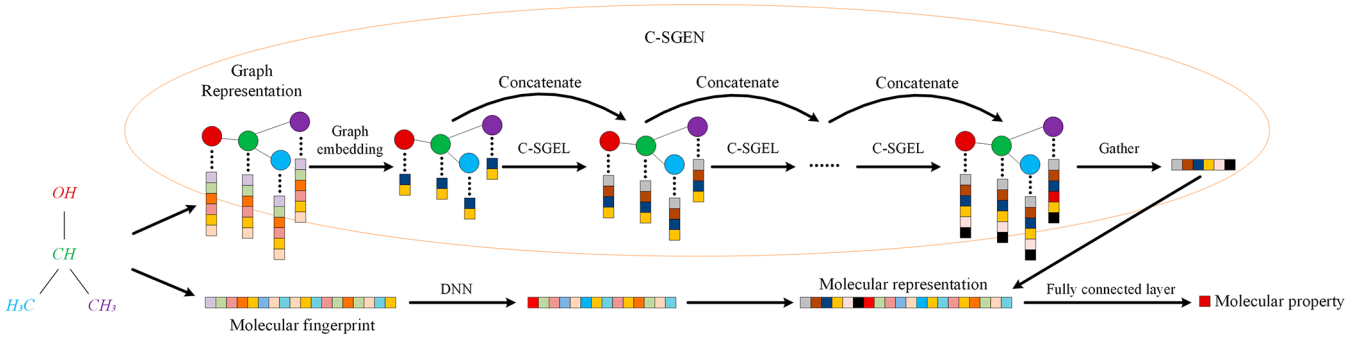


Figure 1. Composited model diagram: schematic diagram of the network structure for predicting molecular properties.

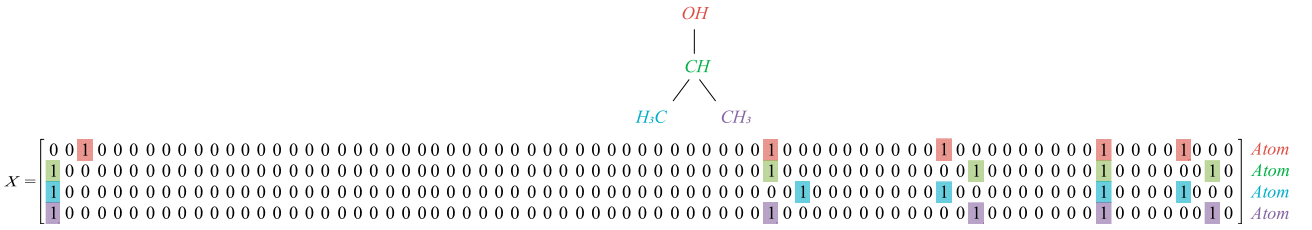


Figure 2. Initial feature matrix of molecules. Extraction of simple atom attributes for isopropanol. Atoms are color coded to indicate the feature vector corresponding to atoms.

property prediction tasks. Kearnes et al. designed a special graph convolution model to exchange and update the information between atomic and pairwise information in each module, and the invariant-preserving descriptor can be obtained.[20] Tsubaki et al. combined GCN for molecular graphs and CNN for protein sequences to design an end-to-end representation learning model.[21] Li et al. reported an attention-based graph convolution network that dynamically learns graph structural features.[22]

However, there are still some problems that need to be solved for GCN in molecular property prediction. First, although there are many graph-based convolution methods used for property prediction, the accuracy of them still needs to be improved. So designing the convolution methods to improve the performance of the molecular property prediction is the first problem. Second, the molecular fingerprint could provide useful information for prediction. So combining this kind of information in the graph model is another problem that needs to be solved, which could improve the performance of the prediction.

The contributions of this paper are listed as follows:

A convolution spatial graph embedding layer (C-SGEL) based on learnable graph convolutional networks[23] is proposed, which enables the convolution operation and can be applied on the molecular graph data. Referring to the traditional natural language processing (NLP) tasks, we assume that a molecule is represented by multiple atomic sentences. For each atom, the 1-D CNN is applied to process the spatial graph matrix, where the neighbor atoms are taken into account. Each atom is represented by a vector. And, the spatial graph matrix is generated according to the connection between atoms. Section 2.C of Methods introduces the detail of the spatial graph matrix.

In addition, multiple C-SGELs are stacked to construct the convolution spatial graph embedding network (C-SGEN), which is applied to learn features from molecular graphs.

Finally, the molecular fingerprint is introduced to enhance the generalization performance of the feature, and a composite model combing both graph features and fingerprints is designed for predicting molecular properties.

2. METHODS

2.A. Model Architecture. Figure 1 provides a detailed schematic of the proposed method in this paper. In our model, each molecule is represented by undirected graph and molecular fingerprints, respectively. Specifically, four processes are implemented including graph representation layer, graph embedding layer, the C-SGEL, and the graph gathering layer. Moreover, the C-SGEN is proposed combining these four processes to apply CNNs on molecular graph data. In addition, Ryu et al.[24] and Gao et al.[23] use the method of skipping connection in each hidden layer to solve the vanishing gradient problem caused by deeply stacked graph convolution layers, after each C-SGEL, skip connections are introduced to concatenate the output of each C-SGEL, which could make full use of molecular information and uncover the relationships between atoms of molecules. Meanwhile, molecular fingerprints are introduced to obtain robust molecular information. Finally, the molecular graph and fingerprint representations are concatenated together, and the molecular properties are predicted through the fully connected layer. In this section, we will introduce each layer of the model architecture in detail.

2.B. Graph Representation of Molecules. Convolutional neural networks cannot be directly applied to data with irregular spatial structures such as graphs. However, GCN can effectively extract spatial features from the topological relation for learning, which can be specially used to process the chemical structure data of molecules. First, molecules need to be specifically transformed and embedded into regular input in order to be suitable for neural networks. The graph of each molecule G can be expressed as an adjacency matrix and an initial node matrix. The adjacency matrix represents the connection information between atoms, which can be

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

represented by a matrix of A, $A \in R^{n\times n}$, where n denotes the number of atoms in a molecule. If there is an edge connection between atoms i and j, $A_{ij} = 1$; otherwise, $A_{ij} = 0$. Here the adjacency matrix takes into account the self-connection of atomic pairs, and we do not consider the features of atomic bonds, since previous work[25] has used an attention mechanism[26] to demonstrate that the atomic bond features have little influence on the atomic properties.

The initial node matrix X represents sets of n atoms considered, $X = \{x_1, x_2, ..., x_n\}$, and each atom $x_i = (x_{i1}, x_{i2}, ..., x_{im})$, $x_i \in \chi$ is a m-dimensional vector, where x_{ij} is the value of x_i on the jth feature and m is the feature dimension of the atom. All features of one molecule are extracted through the DeepChem[27] package and encoded into a one-hot vector. A simple example of isopropanol is shown in Figure 2. The features include atomic type, number of neighbors, number of valencies, charges, number of radical electrons, hybridization, aromaticity indicator, and number of hydrogen neighbors, which are described in Table 1. In order to apply the regular convolution operation to the molecular graph data and realize end-to-end learning, the size of the adjacency and feature matrix of molecules should be fixed. In this paper, the adjacency matrix is initialized with zero, and the number of atoms is set as the average of the total atoms in the dataset.

Table 1. Description of Atom Features

feature	description	size
atom type	C, N, O, S, F, Si, P, Cl, Br, Mg, Na, Ca, Fe, As, Al, I, B, V, K, Tl, Yb, Sb, Sn, Ag, Pd, Co, Se, Ti, Zn, H, Li, Ge, Cu, Au, Ni, Cd, In, Mn, Zr, Cr, Pt, Hg, Pb, or "Unknown" (one-hot)	44
degree	number of directly bonded neighbors (one-hot)	11
valence	number of implicit valence (one-hot)	7
formal charge	integer electronic charge	1
radical electrons	number of radical electrons	1
hybridization	sp, sp^2, sp^3, sp^3d, or sp^3d^2 (one-hot)	5
aromaticity	whether the atom is part of an aromatic systems	1
hydrogens	number of hydrogens neighbors (one-hot)	5
		75

The order of the atom in the matrix is crucial to the performance of the model. The degree of the atoms is selected as the criteria for atom order. Since molecular graphs belong to undirected graphs, the degree of each atom in a molecule is the number of its neighbor atoms. In this paper, the initial feature matrix in a molecule is sorted by the order of atom decreased degree. We also tested other sorting methods, and the details are discussed in section 2.C.

2.C. Graph Embedding Layer. After obtaining the graph representation of molecules, the embedding layer is introduced to obtain a spatial graph matrix for each atom.

First, the dimension reduction method is applied to process the atomic feature matrix X. Because matrix X uses one-hot encoding and data distribution is too sparse as shown in Figure 2, it is possible that this type of data is not conducive to the training process of the model, which may lead the curse of dimensionality. Therefore, a simple linear transformation is utilized to transform the original high-dimensional atomic feature vectors into low-dimensional vectors as follows.

$$X^F = X^0 W + b \tag{1}$$

where X^0, $X^0 \in R^{n\times d_0}$ is the high-dimensional initial feature matrix obtained from the graph representation layer. $W \in R^{d_0\times d_1}$ is the learnable weights matrix responsible for transforming X^0 to low-dimensional representations X^F, $b \in R^{n\times d_1}$ is the learnable bias. Therefore, the transformed atomic feature vector is $X^F \in R^{n\times d_1}$, d_0 and d_1 are the dimension sizes of X^0 and X^F, and $d_1 < d_0$.

Subsequently, in order to represent the space connection of each atom, the X^S, $X^S \in R^{n\times n\times d}$ is calculated as follows.

$$X^S_{i,j} = \begin{cases} X^F_j & A_{i,j} = 1 \\ 0 & A_{i,j} = 0 \end{cases} \tag{2}$$

where X^F_i denotes the ith row of the matrix X^F. A simple example of this molecular representation is shown in Figure 3.

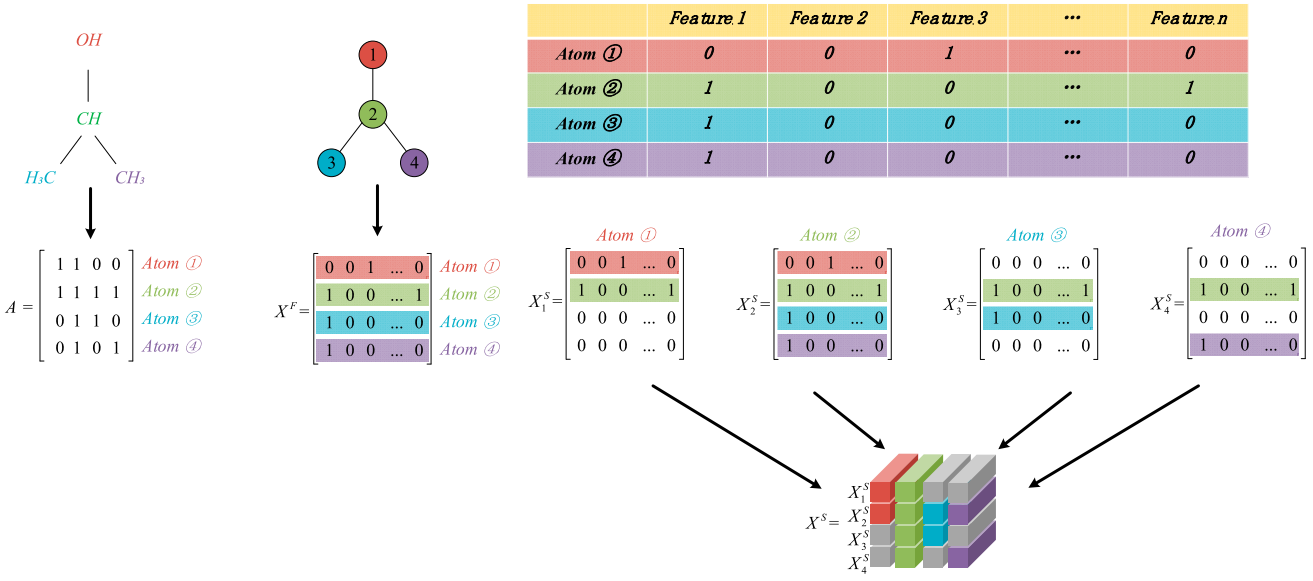


Figure 3. Initial spatial graph matrix of each atom in the molecule. Example representation of isopropanol. Among them, X^F is the molecular feature matrix after dimension reduction, and X^S_i, $i \in \{1, 2, 3, 4\}$ is the spatial graph matrix of each atom.

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

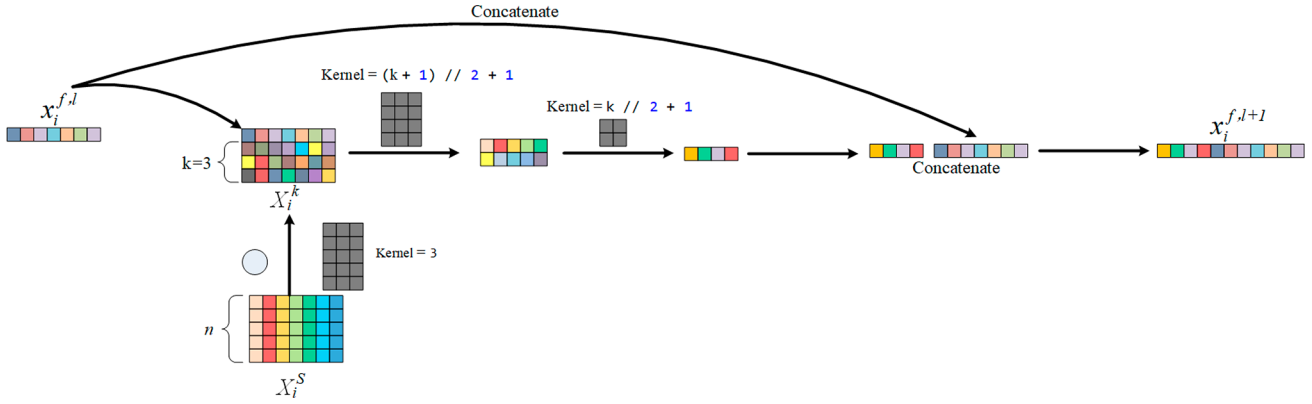


Figure 4. Convolution processing spatial graph embedding layer (C-SGEL) schematic for updating atomic representations. Assuming the number of channels produced by the first convolution k is 3, where x_i^f is the feature vectors of the atom i and X_i^S is spatial graph matrix of the atom i. First, a 1D CNN is applied to process X_i^S, and x_i^f is concatenated with the output matrix X_i^f. In particular, in order to maintain the atomic feature dimension, the first convolutional layer has a kernel, stride, and padding size of 3, 1, and 1, respectively. Subsequent two 1D convolutional layers are applied to obtain the final atomic feature vector, the convolution kernel size is $(k + 1)//2 + 1$ and $k//2 + 1$, the number of output channels, is 2 and 1, respectively, and without padding. The updated atomic feature vector is concatenated with the initial atomic feature vector and input to the next layer.

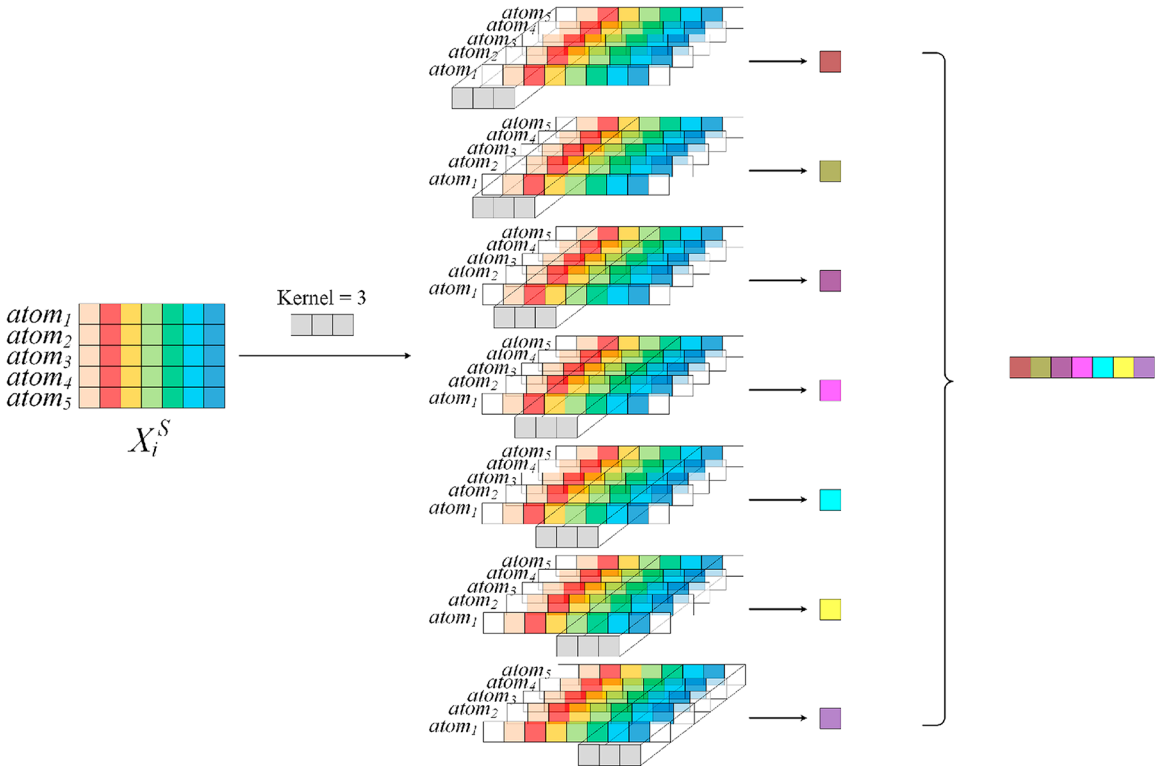


Figure 5. 1D CNN schematic.

In general, the matrix X^S represents the feature of each atom in the molecule taking into account the spatial connection with other atoms. Each atom is represented by a spatial graph matrix $X_i^S \in R^{n\times d}$, $i \in n$, which contains the connection information between the atom and its neighbor atoms (similar to the sentence vector). Both X^F and X^S are as input of C-SGEL.

2.D. Convolution Spatial Graph Embedding Layer (C-SGEL). After obtaining the input of the model, defining the convolution method which is suitable for the graph data is an important issue. Inspired by learnable graph convolutional networks,[23] we introduce the convolution spatial graph embedding layer (C-SGEL) in this section, which mainly process the spatial graph matrix $X_i^S \in R^{n\times d}$, $i \in n$, and the feature vector $x_i^f = X_i^F$, $i \in n$ for each atom.

According to the introduction in section 2.C, for molecules with different atomic numbers, the size of X_i^S is fixed. Therefore, the 1D CNN process is applied to each atom directly.

The convolution process is shown in Figure 4. First, for each atom, the atomic spatial graph matrix X_i^S is convoluted to obtain $X_i^k \in R^{k\times d}$, k is the number of channels produced by the first 1D convolution and also determines the kernel size of the subsequent 1D convolution. It is an important hyper parameter

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

in the network, and the convolution process is shown in Figure 5. Then the feature vector x_i^f is concatenated with X_i^k as input to the next a two-layer convolution. Therefore, the updating of each atomic representation takes into account the information on the atom's neighbor and the information on the atom itself. That is to learn the information on neighbor atoms, and focusing on their own information.

Subsequently, multiple C-SGELs are stacked to obtain the C-SGEN as shown in Figure 1. Each C-SGEL can extract information from direct neighboring atoms, C-SGEN will extract multiorder neighborhood information on atoms, and the updated atomic representation is obtained by repeated C-SGEL calculation. Similar to k, the number of C-SGELs is the important hyperparameter, which will be determined in the experiments.

2.E. Graph-Gathering Layer. After C-SGEN, each atom obtains the optimal vector representation x_i^{output}, $i \in n$. However, in predicting molecular properties, a readout operation is needed to aggregate atomic information and obtain the descriptor of the molecule graph. With help of it, features of the same molecule with different atom sequential orders could be same. In order to enhance the ability of the atom to integrate with surrounding information and be insensitive to the variance of feature ordering, we choose the same method as to calculate the sum of all atomic feature x_i^{output} as follows:

$$x_{\text{graph}} = \sum_i^n x_i^{\text{output}} \tag{3}$$

The molecular representation x_{graph} has the same dimension as the atomic representation x_i^{output}.

2.F. Deep Neural Network for Molecular Fingerprint. As shown in Figure 1, in order to ensure the generalization of molecular feature, the molecular fingerprint is introduced as another input of the network. We also tested the results of fingerprint processing by different methods. Finally, we chose to use DNNs (deep neural networks) to process molecular fingerprints to obtain updated neural fingerprint representations.[10,28,29] Detailed information was discussed in section 2.D. Molecular fingerprints are extracted by ChemoPy[30] (see the Supporting Information), which is a powerful Python software package used to calculate a large number of molecular descriptors in the field of chemical informatics. The fingerprint of a molecule is expressed as $x_{\text{finger}} \in R^{1\times d}$. Equation 4 is used to calculate the hidden layer of the lth layer of the convolutional neural network is as follows

$$x^l = \sigma(w^l x^{l-1} + b^l) \tag{4}$$

where σ is a nonlinear activation function, we choose to use ReLU,[31] w^l and b^l are the learnable weight matrix and bias for the lth convolution layer, respectively, and $x^0 = x_{\text{finger}}$, The output x^l of the last convolution is expressed as $x_{\text{finger\_optimal}}$.

2.G. Prediction of Molecular Properties by Fully Connected Layers. We concatenate $x_{\text{graph}} \in R^{1\times f_1}$ and $x_{\text{finger\_optimal}} \in R^{1\times f_2}$ obtained from graph aggregation layer and molecular convolution layer into the final molecular representation x_{molecule}, that is

$$x_{\text{molecule}} = [x_{\text{graph}}, x_{\text{finger\_optimal}}] \tag{5}$$

Finally, the obtained molecule is expressed as x_{molecule}, and the predictive properties of the molecule are obtained through the fully connected layer, as

$$y_{\text{predicted}} = w_{\text{molecule}} x_{\text{molecule}} + b_{\text{molecule}} \tag{6}$$

where $w_{\text{molecule}} \in R^{1\times(f_1+f_2)}$ is the weight matrix and $b_{\text{molecule}} \in R^1$ is the bias vector.

3. IMPLEMENTATION AND ACCESSIBILITY

Our model was implemented using DeepChem and PyTorch Geometric (PyG) for all model and tensor-related computations in the Python programming language. All molecular manipulations were handled using rdkit and ChemoPy. Modeling experiments were carried out using a machine with an Intel(R) Xeon(R) E5-2620 v4 at 2.10 GHz CPU, 64 GiB of RAM, and an NVIDIA TITAN Xp graphics card.

4. RESULTS AND DISCUSSION

The prediction performance of model is validated in the experiment. In order to compare with the benchmarks in MoleculeNet,[32] ESOL, FreeSolv, and Lipophilicity are randomly split, and PDBbind is time split. All datasets are split into three sets, training set (80%), validation set (10%), and testing set (10%), and then fixed different numerical seeds on the model to evaluate the performance. In addition, for different datasets, we also experimentally determine the hyperparameters of the model.

4.A. Datasets. In order to train and test the performance of the model in the Physical Chemistry and Biophysics tasks, five publicly available benchmark datasets are obtained from DeepChem package[27] which democratizes the use of deep-learning in drug discovery, materials science, quantum chemistry, and biology.[27]

- ESOL: ESOL is a widely used public benchmark dataset for aqueous solubility prediction, it consists of 1128 kinds of aqueous solubility data calculated directly from the structure of compounds.[33] These compounds represent their structures by SMILES formats,[34] so these structures do not include information about the spatial arrangement of atoms in the molecule.
- FreeSolv: The Free Solvation Database is a selective database of 642 neutral small molecule experiments and calculations of hydration free energies.[35] The structural information is contained in SMILES strings; the experimental values are used for model training. The SAMPL4 challenge selected 47 small molecules from the dataset to blindly predict the solvation free energy.[36]
- Lipophilicity: This repository archives a lipophilicity dataset that includes the chemical structure (SMILES) of 4200 organic compounds and their n-octanol/buffer solution distribution coefficients at pH 7.4 (logD7.4). As a determinant of several ADME properties, lipophilicity (logD7.4) is a key physical property in the development of small molecule oral drugs. This dataset, curated from ChEMBL database, can be applied for method benchmarking in regression modeling, cheminformatics, and chemometrics research.
- PDBbind-full: The full subsets of PDBbind provide a comprehensive collection of experimental binding affinity data for biomolecular complexes.[37,38] The primary reference of each complex was examined to collect experimentally determined binding affinity data (K_d, K_i, and IC50) of the given complex. This type of information is the much-needed basis for various computational and statistical studies on molecular

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

recognition. Binding data for roughly 15 000 complexes. This dataset only provides the structure of ligands.

- PDBbind-refined: PDBbind is a comprehensive database of experimental binding affinity ($-\log K_d/K_i$) data for biomolecular complexes. PDBbind-refined contains about 4000 complete data; it is the protein−ligand complexes dataset with better quality out of the full data.

4.B. Metrics. In order to compare with the baseline models in the paper, root mean square error (RMSE) is applied to evaluate the performance of the model on the regression tasks.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2} \tag{7}$$

where f_i and y_i represent the predicted and experimental ith molecular property. RMSE indicates the deviation between the predicted and experimental values; the smaller the value of RMSE, the better the performance of the predicted model.

4.C. Comparison of the Different Atomic Order. In order to determine the effect of different ranking methods of atoms on the experimental results, the atoms are sorted by atom degree in increasing, decreasing and random order, respectively.

The results shown in Figure 6 that, when the atoms are sorted in decreasing degree order, the generated node matrix

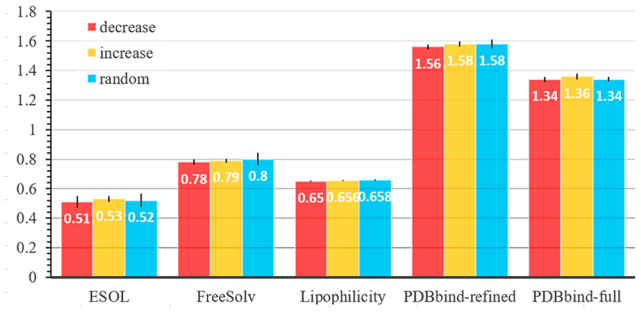


Figure 6. Comparisons of atomic sorting results. The experimental results of atoms with different ranking under the optimal parameters.

consists of those atoms with more adjacent edges; this also indicates that more connection information and structures are considered. On the other hand, if all atoms sorted in increasing degree order, it is possible that atoms included in the model are with fewer connections with each other, which could affect the result of prediction.

4.D. Comparison of Fingerprint Processing in Different Methods. For each dataset, we tested the model using DNN and CNN to process molecular fingerprints respectively. Meanwhile, we also added controlled experiment which utilized the fingerprint directly without any processing. The specific results are shown in the Table 2. It is demonstrated that the effect of using DNN to process molecular fingerprints was better than that of CNN and None.

4.E. Model Training and Hyperparameter Optimization. The model is trained on the training set using different hyperparameters, based on the result of verification for hyperparameter tuning, and the final model performance is assessed on the test set. Table 3 lists the hyperparameters that need to be optimized and the possible values. On all datasets, we chose the adaptive moment (Adam)[39] as the optimizer, with default parameters for momentum scheduling $\beta 1 = 0.99$,

Table 2. Comparison of Fingerprint Processing in Different Methods

	ESOL		FreeSolv		lipophilicity		PDBbind (refined)		PDBbind (full)	
	valid	test	valid	test	valid	test	valid	test	valid	test
DNN	0.62 ± 0.02	0.51 ± 0.04	1.19 ± 0.04	0.78 ± 0.02	0.763 ± 0.007	0.650 ± 0.002	1.44 ± 0.04	1.56 ± 0.02	1.40 ± 0.02	1.34 ± 0.02
none	0.95 ± 0.06	0.92 ± 0.09	1.45 ± 0.05	1.30 ± 0.07	0.914 ± 0.006	0.862 ± 0.004	1.55 ± 0.01	1.78 ± 0.05	1.50 ± 0.03	1.47 ± 0.06
CNN	1.36 ± 0.06	1.39 ± 0.07	2.50 ± 0.59	1.97 ± 0.52	1.078 ± 0.117	1.032 ± 0.133	1.68 ± 0.03	1.75 ± 0.02	1.62 ± 0.02	1.41 ± 0.01

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

Table 3. Hyperparameter Considerations for the Model

hyperparameters	selection optimization
C-SGELs layer	1, 2, 3, 4, 5, 6
learning rate	5×10^{-3}, 1×10^{-3}, 5×10^{-4}, 1×10^{-4}, 5×10^{-5}, 1×10^{-5}
k	2, 4, 8, 16, 32
graph embedding size	4, 8, 16, 32, 64
mini-batch	2, 4, 8, 16, 32, 64, 96, 128
dropout	0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
initialization function	He-normal, He-uniform normal, uniform, Glorot-normal

$\beta 2 = 0.999$. We use Glorot weights to initialize our network and rectified linear unit (ReLU) as activation function. Meanwhile, in order to improve the generalization ability of the model and avoid overfitting, the penalty term L2 weight decay with $\lambda = 0.0005$ is applied. Table 4 lists the concrete set of hyperparameters and the training time for each epoch on each dataset.

In practice, the number of epochs was set to 75 at most, and the early stopping strategy was employed based on the training and validation loss. As shown in Figure 7, the network was trained iteratively until it reached a convergence state. The number of training epoch more than a certain number led to overfitting of the network, indicated by the inconsistent trends in training and validation sets.

In the Physical Chemistry datasets, the smiles format data in ESOL, FreeSolv, and Lipophilicity are input into the model. First, the characteristic dimension of atoms in the molecule is reduced through the graph embedding layer. Here, the output size of the graph embedding layer is set to 48, 16, and 4, respectively. Subsequently, the embedded molecular representation is processed with 6, 2, and 2 C-SGELs and same number of k is set to 4, respectively. Figure 8 shows the effects of different C-SGELs and k values on the performance of the model.

For PDBbind-full and PDBbind-refined datasets, the output size of the graph embedding layer is set to 64 and 16. As shown in Figure 8, we stack 5 and 1 C-SGELs with $k = 16$; when the learning rate is 1×10^{-5} and the dropout of the processing molecular feature matrix and the adjacency matrix is 0.5 in each layer, the model achieves the best performance. The specific model configuration for each dataset is described in Table 4.

For different datasets, different numbers of C-SGELs are set to achieve optimal performance. In our opinion, this is related to the complexity of molecular structure in datasets and the quality of datasets. The molecular structure of ESOL datasets is relatively simple, so the model needs more C-SGELs and stronger learning ability. The model does not work well on PDBbind-full datasets, which may be due to the fact that the data in PDBbind-full datasets are not clean enough to cause serious overfitting.[32] Therefore, this may lead to not too many C-SGELs of our model.

4.F. Property Prediction Performance Comparison. The RMSE results of the proposed composite model and various baseline models in the paper[32] are listed in Tables 5 and 6. On a machine learning (ML) basis, the model includes random forest (RF),[40] which is a classifier that uses multiple decision trees to train and predict samples. It can also be used for regression problems. Kernel ridge regression (KRR) is a linear function which combines kernel techniques and ridge regression. Gradient boosting (XGBoost)[41] is a boosting method. The main idea is to build a model in the direction where the loss gradient of the previous model decreases.

In terms of graph-based methods, the baseline includes graph convolutional models (GC),[19] which refer to circular fingerprints to make convolution process molecular graphs directly and extract effective molecular representations. Directed acyclic graph (DAG)[42] applies recurrent neural networks to the processing of undirected cyclic graphs of molecules. The weave module[20] uses the attributes of atoms and bonds in small molecules to calculate the feature vector for each pair of atoms. The message passing neural network (MPNN)[43] designs a single common framework on graphs. Multitask networks[44] are also used for comparison; these are networks that can share weights and process multiple tasks.

Besides, four recently proposed graph-based models are selected for comparison. Graph attention networks (GATs)[26] give different nodes different importance in the convolution process by stacking the masked self-attention layer, and different sizes of domains can be processed at the same time. Attention-based graph neural network (AGNN)[45] removes all the completely connected layers in the graph neural network and replaces the propagation layer of the network with the attention mechanism based on graph structure to obtain a dynamic and adaptive local summary of the field. ARMA[46] is a graph convolutional layer based on autoregressive moving average filters, in which the filters have a recursive and distributed formulation. SGC[47] reduces the unnecessary complexity and redundancy of GCN by eliminating nonlinearities and collapsing weight matrices between consecutive layers. The additional graph-based models are implemented by PyTorch Geometric,[48] which consist of various methods for deep learning on graphs and other irregular structures from a variety of published papers. Compared with these benchmark models, the proposed model has comparable performance on physical chemistry and biophysics tasks.

Table 4. Model Configuration for Each Dataset

hyperparameters	ESOL	FreeSolv	lipophilicity	PDBbind (refined)	PDBbind (full)
C-SGELs layer	6	2	2	5	1
learning rate	5×10^{-5}	5×10^{-4}	1×10^{-4}	1×10^{-5}	1×10^{-5}
k	4	4	4	16	16
graph embedding size	48	16	4	64	16
mini-batch	32	8	4	4	96
dropout	0.5	0.5	0.5	0.5	0.5
epochs	43	33	51	57	11
training time (GPU/epoch)	6 s	2 s	18 s	22 s	76 s

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

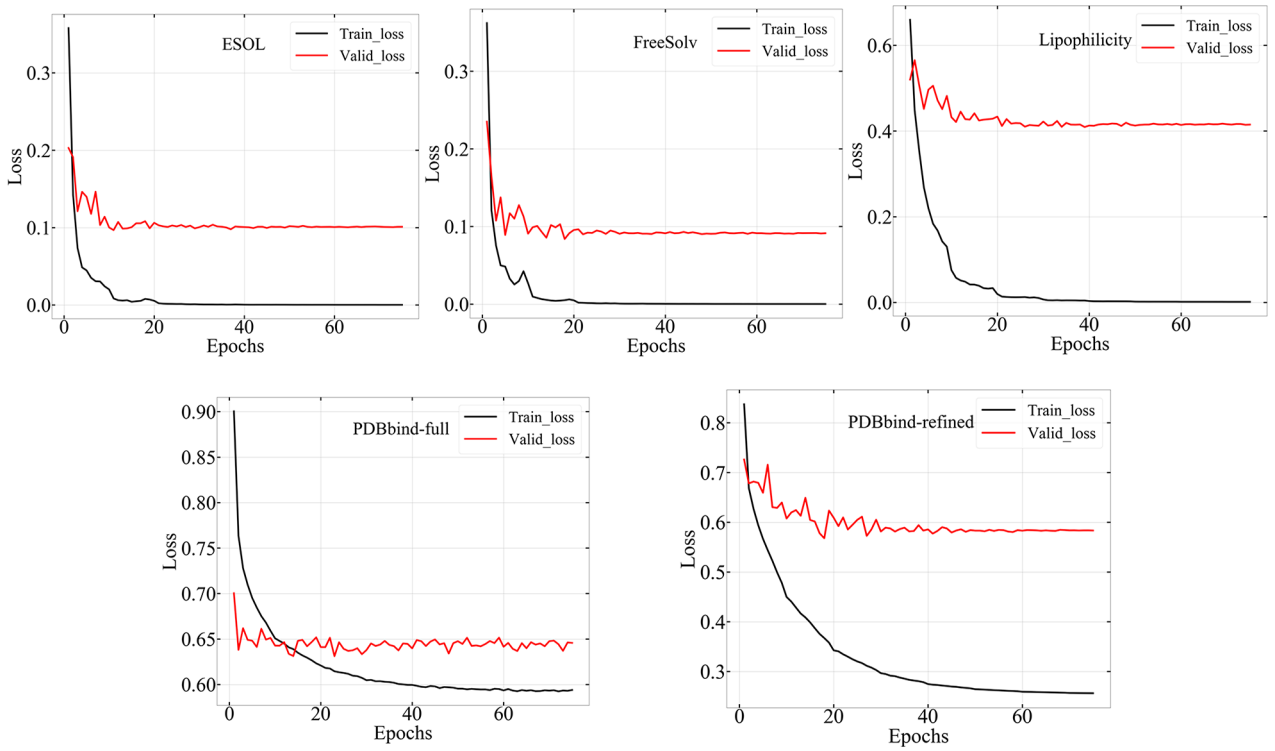


Figure 7. Loss curve of verification set and test set. The black curve represents the training curve of the model on the training set, and the red curve represents the loss curve of the model on the verification set. After training for a certain number of epochs, the curve tends to be smooth.

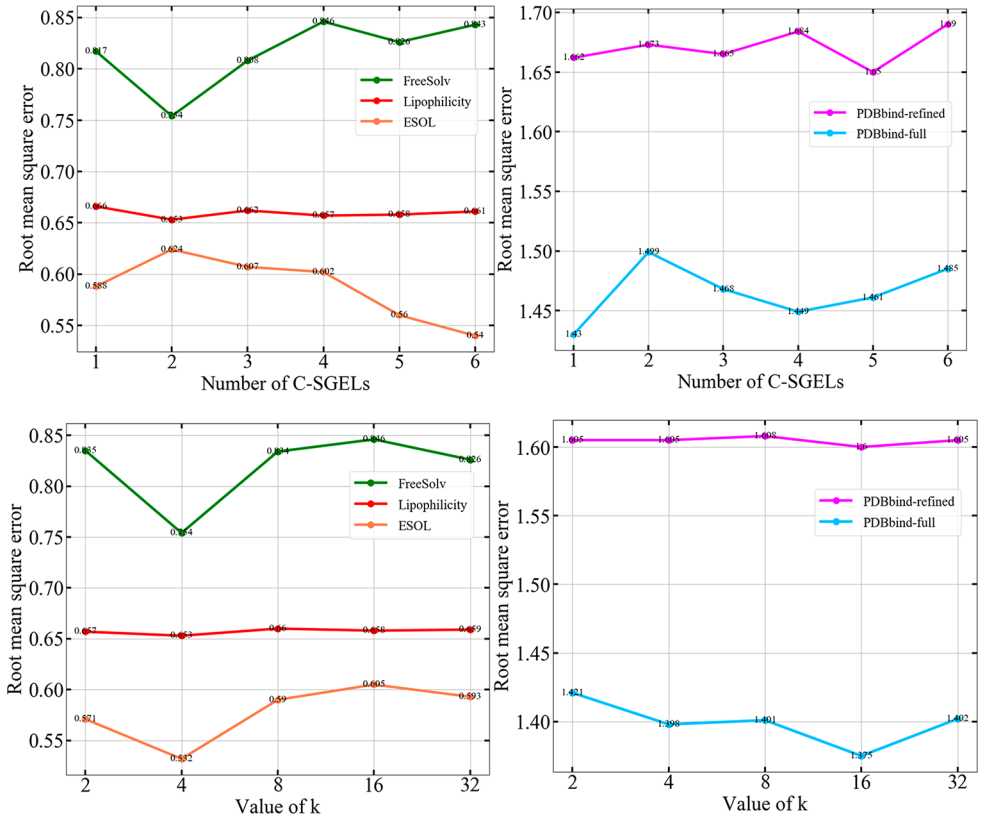


Figure 8. Result trend charts of different C-SGELs and *k* values on the ESOL, FreeSolv, Lipophilicity, PDBbind-refined, and PDBbind-full datasets. The figures show that, when the remaining hyperparameters remain invariant, the layers are set to 6, 2, 2, 1, and 5, and *k* is set to 4, 4, 4, 16, and 16, respectively, the model achieves the best performance.

In short, our methods obtained the best results among physical chemistry test datasets. In the ESOL and Lipophilicity benchmark dataset, message passing neural network (MPNN)[43] and GC[19] achieve optimal performance on the

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

Table 5. Prediction Results of Physical Chemistry Tasks

task	Physical Chemistry (RMSE)					
dataset	ESOL		FreeSolv		Lipophilicity	
model	validation	testing	validation	testing	validation	testing
RF	1.16 ± 0.15	1.07 ± 0.19	2.12 ± 0.68	2.03 ± 0.22	0.835 ± 0.036	0.876 ± 0.040
multitask	1.17 ± 0.13	1.12 ± 0.15	1.95 ± 0.41	1.87 ± 0.07	0.852 ± 0.048	0.859 ± 0.013
XGBoost	1.05 ± 0.10	0.99 ± 0.14	1.76 ± 0.21	1.74 ± 0.15	0.783 ± 0.021	0.799 ± 0.054
KRR	1.65 ± 0.19	1.53 ± 0.06	2.10 ± 0.12	2.11 ± 0.07	0.889 ± 0.009	0.899 ± 0.043
DAG	0.74 ± 0.04	0.82 ± 0.08	1.48 ± 0.15	1.63 ± 0.18	0.857 ± 0.050	0.835 ± 0.039
GC	1.05 ± 0.15	0.97 ± 0.01	1.35 ± 0.15	1.40 ± 0.16	**0.678 ± 0.040**	0.655 ± 0.036
weave	0.57 ± 0.04	0.61 ± 0.07	1.19 ± 0.08	1.22 ± 0.28	0.734 ± 0.011	0.715 ± 0.035
MPNN	**0.55 ± 0.02**	0.58 ± 0.03	1.20 ± 0.02	1.15 ± 0.12	0.757 ± 0.030	0.719 ± 0.031
SGC	1.03 ± 0.00	0.87 ± 0.00	2.39 ± 0.01	1.98 ± 0.04	1.046 ± 0.000	0.998 ± 0.001
AGNN	0.95 ± 0.02	0.76 ± 0.01	1.72 ± 0.06	1.24 ± 0.03	1.001 ± 0.005	0.963 ± 0.002
GAT	0.84 ± 0.01	0.71 ± 0.02	2.03 ± 0.14	1.37 ± 0.17	0.950 ± 0.006	0.889 ± 0.006
ARMA	0.82 ± 0.02	0.72 ± 0.01	1.48 ± 0.05	1.08 ± 0.06	0.953 ± 0.010	0.894 ± 0.006
C-SGEN+ fingerprint	0.62 ± 0.02	**0.51 ± 0.04**	**1.19 ± 0.04**	**0.78 ± 0.02**	0.763 ± 0.007	**0.650 ± 0.002**

Table 6. Prediction Results of Biophysics Tasks

task	Biophysics (RMSE)			
dataset	PDBbind-refined		PDBbind-full	
model	validation	testing	validation	testing
RF (grid)	**1.37 ± 0.00**	**1.38 ± 0.00**	**1.35 ± 0.00**	**1.25 ± 0.00**
multitask	1.53 ± 0.03	1.66 ± 0.05	1.42 ± 0.05	1.45 ± 0.14
multitask (grid)	1.41 ± 0.02	1.46 ± 0.05	1.40 ± 0.03	1.28 ± 0.02
GC	1.55 ± 0.05	1.65 ± 0.03	1.57 ± 0.20	1.44 ± 0.12
SGC	1.64 ± 0.02	1.81 ± 0.03	1.43 ± 0.01	1.35 ± 0.00
AGNN	1.61 ± 0.00	1.77 ± 0.00	1.42 ± 0.00	1.35 ± 0.00
GAT	1.60 ± 0.01	1.78 ± 0.01	1.42 ± 0.00	1.35 ± 0.01
ARMA	1.62 ± 0.01	1.76 ± 0.00	1.39 ± 0.00	1.36 ± 0.00
C-SGEN + fingerprint	1.44 ± 0.04	1.56 ± 0.02	1.40 ± 0.02	1.34 ± 0.02

validation dataset, respectively. At the same time, our model outperforms these two models and achieves 12.1% and 7.6% performance improvement in the test dataset, respectively.

For the FreeSolv dataset, our model gets the same performance as weave[20] and achieves optimal performance on the validation dataset. Meanwhile, compared with ARMA,[46] we improve the best performance by 29.7% and achieve the state-of-the-art results on test datasets. This demonstrates that our model has better generalization ability, and the predicted value of the model is closer to the real value.

For the PDBbind dataset, our model achieves the best performance in the graph-based model, but it does not exceed the grid-based feature extraction method. It is possible that for the data of ligand-protein binding, the best choice is to build a grid to extract voxels.[15−17,49] For the compound after docking, more information should be given such as the docked pose, energy, and chemical bonds of the docking of complex, etc. The grid method can better describe the 3D structure of the complex.

These results fully prove the validity of transforming molecular graph data into regular grid structure data and employing regular one-dimensional convolution to process the spatial graph matrix of the molecule retaining spatial information. Compared with other graph-based methods, regular convolution calculation can extract more potentially effective features from the input molecular spatial information. Owing to the fact that convolution uses regular convolution kernels, the features extracted by convolution pay more attention to local information, and then, the local information will be synthesized at a higher level to get global information. For molecular data, the use of convolution focuses more on the functional group structure of molecules, which determines the chemical properties of molecules.

4.G. Comparison Effect of Fingerprint. In order to improve the prediction and generalized performance of the model, the molecular fingerprint is introduced. In this section, the fingerprint model, SGEN model, and the composite model were measured on Physical Chemistry and Biophysics datasets with the typical hyperparameters, respectively. Figure 9 displays the results. On all datasets, models using fingerprint input alone perform poorly. This may be due to the insufficient

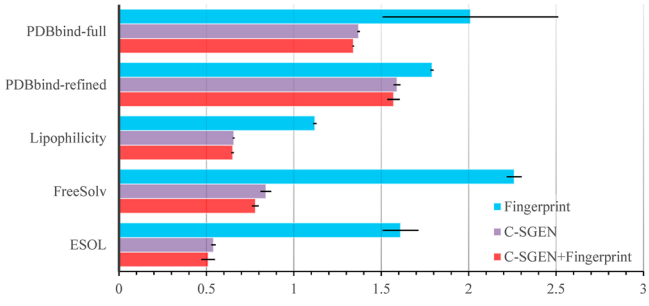


Figure 9. Performance of the C-SGEN model, fingerprint model, and the composite model on each dataset. The *x*-axis reports RMSE for test subsets under different random seeds. Error bars represent standard deviation.

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

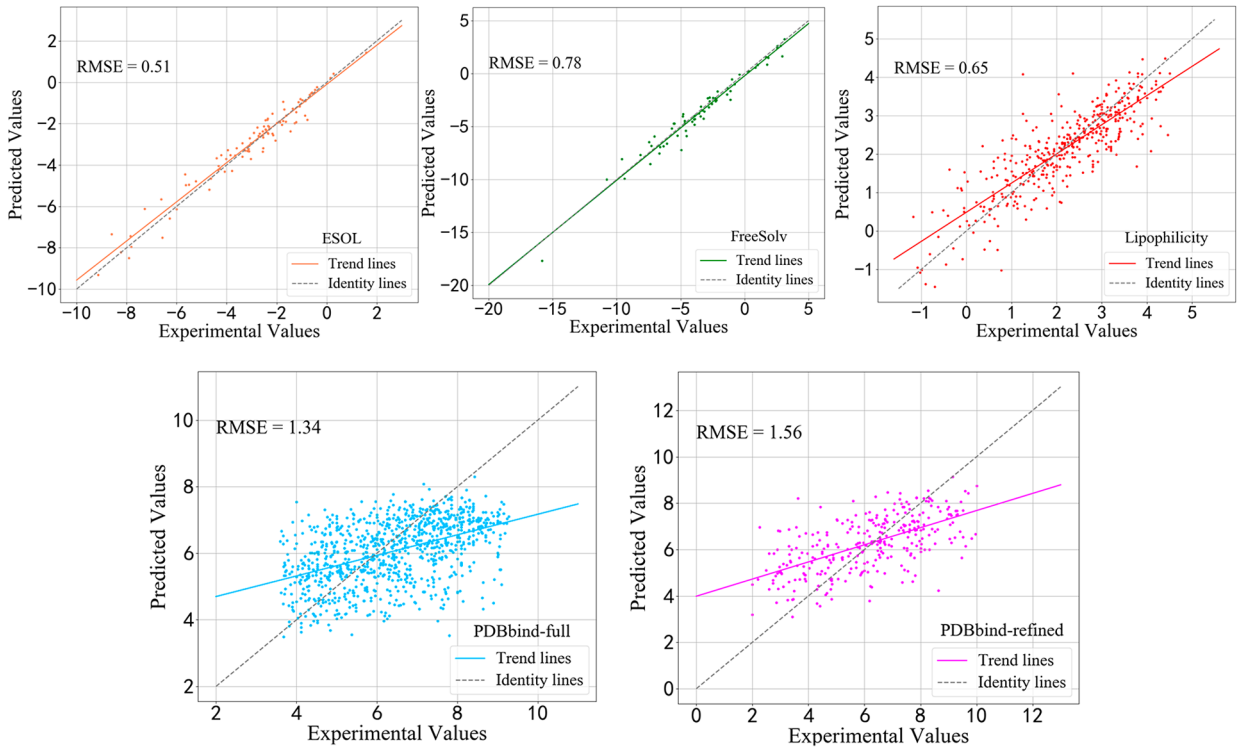


Figure 10. Confusion diagrams of experimental and predicted values on test subsets for all datasets. The red solid line is the fitting line, and the gray dashed line is the baseline. The values above and below the baseline are overpredicted and underpredicted values.

information on predefined molecular features and the low correlation between features and molecular properties, which makes the model unable to extract potentially effective information. At the same time, the results of C-SGEN on the test set are comparable with those of the optimal baseline model, and there are no large gaps between the results under different random seeds. This indicates the powerful performance of the C-SGEN model for processing atomic graph matrices, which can extract useful information from the neighborhood information on atoms and learn the spatial graph matrix of each atom with correct potential representation. Specifically, the performance of models on ESOL and FreeSolv datasets has been significantly improved by 5.6% and 7.1%, respectively, while the performance of models on Lipophilicity, PDBbind-refined, and PDBbind-full datasets has been improved by 0.9%, 1.3%, and 2.2%, respectively. These results also indicate the validity of molecular fingerprint information in predicting molecular properties. At the same time, for different datasets, the contribution of fingerprints to C-SGEN is different, which may be due to the relationship between the extracted features and molecular properties. In future work, the performance of the model may be further improved by selecting appropriate fingerprint features for predicting the different properties of molecules.

4.H. Distribution of Experimental and Predicted Value. Figure 10 shows the distribution of experimental and predicted values on test subsets for all datasets. It can be seen from the figure that the fitting curve of the predicted value is close to the datum curve and has a linear relationship. This shows that the predicted value of the model is close to the experimental value, and the model has good performance. In addition, it should be noted that PDBbind splits datasets based on time, which probable due to the uneven distribution of datasets, resulting in a large RMSE on the test set.

5. CONCLUSION

In this work, graph structure is introduced to represent molecular data, which is fed into a convolution network to discover the relationship between each atom. Furthermore, we designed a convolution spatial graph embedding layer (C-SGEL), which used one-dimensional convolution to process the spatial graph matrix of each atom in the molecule. Moreover, multiple C-SGELs are stacked to construct the convolution spatial graph embedding network (C-SGEN). Based on C-SGEN, a composite network is built to combine molecular graphs and fingerprints to predict molecular properties. The experimental results show that our composite model performs better than state-of-the-art model in five different datasets.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.9b00410.

Molecular fingerprints that were extracted by ChemoPy to describe molecules in the Physical Chemistry and Biophysics tasks (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: lizhen@ouc.edu.cn.

ORCID

Zhen Li: 0000-0001-5093-4221

Author Contributions

X.W. initiated the project and wrote the manuscript, Z.L. designed the experiment and supervised the project, M.J.

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

conducted the experiments, S.W. explained the experimental results, S.Z. prepared the data, and Z.W. analyzed the data.

Funding

This work is supported by the National Nature Science Foundation of China (No. 61602430, No. 61872326).

Notes

The authors declare no competing financial interest.

The trained C-SGEN model is free to use, and the code is accessible at https://github.com/wxfsd/C-SGEN.

REFERENCES

(1) Tropsha, A. Best practices for QSAR model development, validation, and exploitation. *Mol. Inf.* **2010**, *29*, 476−488.

(2) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; Consonni, V.; Kuz'min, V. E.; Cramer, R.; Benigni, R.; Yang, C.; Rathman, J.; Terfloth, L.; Gasteiger, J.; Richard, A.; Tropsha, A. QSAR Modeling: Where Have You Been? Where Are You Going To? *J. Med. Chem.* **2014**, *57*, 4977−5010.

(3) Tao, L.; Zhang, P.; Qin, C.; Chen, S. Y.; Zhang, C.; Chen, Z.; Zhu, F.; Yang, S. Y.; Wei, Y. Q.; Chen, Y. Z. Recent Progresses in the Exploration of Machine Learning Methods as In-Silico ADME Prediction Tools. *Adv. Drug Delivery Rev.* **2015**, *86*, 83−100.

(4) Mitchell, J. B. O. Machine Learning Methods in Chemoinformatics. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2014**, *4*, 468−481.

(5) Dudek, A. Z.; Arodz, T.; Gálvez, J. Computational Methods in Developing Quantitative Structure-Activity Relationships (QSAR): A Review. *Comb. Chem. High Throughput Screening* **2006**, *9*, 213−228.

(6) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure−Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.

(7) Yao, X. J.; Panaye, A.; Doucet, J.-P.; Zhang, R. S.; Chen, H. F.; Liu, M. C.; Hu, Z. D.; Fan, B. T. Comparative Study of QSAR/QSPR Correlations Using Support Vector Machines, Radial Basis Function Neural Networks, and Multiple Linear Regression. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1257−1266.

(8) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947.

(9) Niculescu, S. P. Artificial Neural Networks and Genetic Algorithms in QSAR. *J. Mol. Struct.: THEOCHEM* **2003**, *622*, 71−83.

(10) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction Using Deep Learning. *Front. Environ. Sci.* **2016**, *3*, 80.

(11) Unterthiner, T.; Mayr, A.; Klambauer, G.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Hochreiter, S. Deep Learning as an Opportunity in Virtual Screening. In *Proceedings of the deep learning workshop at NIPS*; 2014; Vol. *27*, pp 1−9.

(12) Mamoshina, P.; Vieira, A.; Putin, E.; Zhavoronkov, A. Applications of Deep Learning in Biomedicine. *Mol. Pharmaceutics* **2016**, *13*, 1445−1454.

(13) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*; 2012; pp 1097−1105.

(14) Kim, Y. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; 2014; pp 1746−1751.

(15) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. K. DEEP: Protein−Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018**, *58*, 287−296.

(16) Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-Based Drug Discovery. *arXiv.org* **2015**, 1510.02855.

(17) Imrie, F.; Bradley, A. R.; van der Schaar, M.; Deane, C. M. Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need for More Data. *J. Chem. Inf. Model.* **2018**, *58*, 2319−2330.

(18) Wang, T.; Wu, D. J.; Coates, A.; Ng, A. Y. End-to-End Text Recognition with Convolutional Neural Networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*; 2012; pp 3304−3308.

(19) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in neural information processing systems*; 2015; pp 2224−2232.

(20) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving beyond Fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595−608.

(21) 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*, 309−318.

(22) Li, X.; Yan, X.; Gu, Q.; Zhou, H.; Wu, D.; Xu, J. DeepChemStable: Chemical Stability Prediction with an Attention-Based Graph Convolution Network. *J. Chem. Inf. Model.* **2019**, *59*, 1044−1049.

(23) Gao, H.; Wang, Z.; Ji, S. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; 2018; pp1416−1424.

(24) Ryu, S.; Lim, J.; Kim, W. Y. Deeply Learning Molecular Structure-Property Relationships Using Graph Attention Neural Network. *arXiv.org* **2018**, 1805.10988.

(25) Shang, C.; Liu, Q.; Chen, K.-S.; Sun, J.; Lu, J.; Yi, J.; Bi, J. Edge Attention-Based Multi-Relational Graph Convolutional Networks. *arXiv.org* **2018**, 1802.04944.

(26) Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph Attention Networks. *arXiv.org* **2017**, 1710.10903.

(27) Ramsundar, B. deepchem.io. https://github.com/deepchem/deepchem, 2016.

(28) Gonczarek, A.; Tomczak, J. M.; Zareba, S.; Kaczmar, J.; Dabrowski, P.; Walczak, M.Ł J. Interaction Prediction in Structure-Based Virtual Screening Using Deep Learning. *Comput. Biol. Med.* **2018**, *100*, 253−258.

(29) Kato, Y.; Hamada, S.; Goto, H. Molecular Activity Prediction Using Deep Learning Software Library. In *International Conference on Advanced Informatics: Concepts*; 2016, pp 1−6.

(30) Cao, D.-S.; Xu, Q.-S.; Hu, Q.-N.; Liang, Y.-Z. ChemoPy: Freely Available Python Package for Computational Biology and Chemoinformatics. *Bioinformatics* **2013**, *29*, 1092−1094.

(31) Nair, V.; Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*; 2010; pp 807−814.

(32) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9*, 513−530.

(33) Delaney, J. S. ESOL: Estimating Aqueous Solubility Directly from Molecular Structure. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1000−1005.

(34) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.

(35) Mobley, D. L.; Guthrie, J. P. FreeSolv: A Database of Experimental and Calculated Hydration Free Energies, with Input Files. *J. Comput.-Aided Mol. Des.* **2014**, *28*, 711−720.

(36) Mobley, D. L.; Wymer, K. L.; Lim, N. M.; Guthrie, J. P. Blind Prediction of Solvation Free Energies from the SAMPL4 Challenge. *J. Comput.-Aided Mol. Des.* **2014**, *28*, 135−150.

(37) Chang, T. H.; Ke, C.-H.; Lin, J.-H.; Chiang, J.-H. AutoBind: Automatic Extraction of Protein-Ligand-Binding Affinity Data from Biological Literature. *Bioinformatics* **2012**, *28*, 2162−2168.

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828

(38) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind Database: Methodologies and Updates. *J. Med. Chem.* **2005**, *48*, 4111−4119.
(39) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv.org* **2014**, 1412.6980.
(40) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.
(41) Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189−1232.
(42) Lusci, A.; Pollastri, G.; Baldi, P. Deep Architectures and Deep Learning in Chemoinformatics: The Prediction of Aqueous Solubility for Drug-like Molecules. *J. Chem. Inf. Model.* **2013**, *53*, 1563−1575.
(43) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*; 2017; pp 1263−1272.
(44) Ramsundar, B.; Kearnes, S.; Riley, P.; Webster, D.; Konerding, D.; Pande, V. Massively Multitask Networks for Drug Discovery. *arXiv.org* **2015**, 1502.02072.
(45) Thekumparampil, K. K.; Wang, C.; Oh, S.; Li, L.-J. Attention-Based Graph Neural Network for Semi-Supervised Learning. *arXiv.org* **2018**, 1803.03735.
(46) Bianchi, F. M.; Grattarola, D.; Livi, L.; Alippi, C. Graph Neural Networks with Convolutional Arma Filters. *arXiv.org* **2019**, 1901.01343.
(47) Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K. Simplifying Graph Convolutional Networks. In *International Conference on Machine Learning*; 2019; pp 6861−6871.
(48) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv.org* **2019**, 1903.02428.
(49) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Development and Evaluation of a Deep Learning Model for protein−ligand binding affinity prediction. *Bioinformatics* **2018**, *34*, 3666−3674.

DOI: 10.1021/acs.jcim.9b00410
J. Chem. Inf. Model. 2019, 59, 3817−3828