IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 14, 2020, accepted January 17, 2020, date of publication January 21, 2020, date of current version January 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968535

Molecular Property Prediction Based on a Multichannel Substructure Graph

SHUANG WANG, ZHEN LI, SHUGANG ZHANG, MINGJIAN JIANG, XIAOFENG WANG, AND ZHIQIANG WEI

Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China

Corresponding author: Zhen Li (lizhen0130@gmail.com)

This work was supported by the Marine S&T Fund of Shandong Province for Pilot National Laboratory for Marine Science and Technology (Qingdao) under Grant 2018SDKJ0402.

ABSTRACT Molecular property prediction is important to drug design. With the development of artificial intelligence, deep learning methods are effective for extracting molecular features. In this paper, we propose a multichannel substructure-graph gated recurrent unit (GRU) architecture, which is a novel GRU-based neural network with attention mechanisms applied to molecular substructures to learn and predict properties. In the architecture, molecular features are extracted at the node level and molecule level for capturing fine-grained and coarse-grained information. In addition, three bidirectional GRUs are adopted to extract the features on three channels to generate the molecular representations. Different attention weights are assigned to the entities in the molecule to evaluate their contributions. Experiments are implemented to compare our model with benchmark models in molecular property prediction for both regression and classification tasks, and the results show that our model has strong robustness and generalizability.

INDEX TERMS Molecular graph, molecular property prediction, substructure-graph.

I. INTRODUCTION

The prediction of molecular properties plays a vital role in drug discovery [1], [2]. Traditional methods such as biochemical experiments are always time-consuming and expensive. Molecules are special graph-structured data carrying specified chemical properties. The computer-aided prediction of molecular properties based on molecular structures could accelerate the drug discovery process. The development of artificial intelligence provides an effective method for learning molecular features and predicting properties, and it has been applied to predict drug-disease associations [3].

There are many types of molecular representations used in deep learning [4]–[7]. The first type is SMILES (simplified molecular-input line-entry system), which encodes a molecule into a meaningful sequence following a specified grammar [8], but rings might be broken when the SMILES format is used. It is possible that one benzene ring may correspond to diverse SMILES considering different broken positions. On the other hand, a fingerprint represents a molecule via a sparse binary vector that records the appearance of molecular substructures [9]. However, for a large molecule dataset, too many substructures could result in the sparse representation problem. In addition, the duplication substructure removal process [10] of a fingerprint could also cause molecular information loss.

Compared with fingerprints and SMILES, the graph representation could retain the molecular structure and topological information [5]. Graphs have been applied in many areas and achieved good results [11], [12]. Through the graph representation, one molecule is interpreted as an undirected graph in which an atom is regarded as a node and a bond is regarded as an edge. With the help of graph structures, a number of methods have applied deep learning on molecules and achieved good results such as graph neural networks [13]. Duvenaud *et al.* [14] extracted molecular fingerprints from molecular graphs via a convolutional neural network, which is different from predefined fixed mode. In addition, the feature vectors of atoms become differentiable, which could predict the molecular properties more precisely. Inspired by Duvenaud, Youjun proposed a molecular graph encoding convolutional neural network (MGE-CNN) architecture

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

for acute oral toxicity prediction [15], and Coley utilized a graph-based convolutional neural network for molecular embedding to predict physical properties [16]. The descriptors obtained by a graph neural network perform better than traditional predefined or handcrafted descriptors.

In the above algorithms, the connection relationships provided by molecule edges are very important in the molecular convolution process because information could be passed and updated when using them. The message passing neural network (MPNN) [17] merges related algorithms [14], [16], [18]–[20] into a single framework to predict quantum chemical properties. These models learn the features of atoms from a message passing algorithm and aggregate them to represent the input molecular graph. Here, the atom features in local chemical environments are included, and also the edge features could be learned in the training process. The weave module [18] uses an edge convolutional neural network combined with a node convolution neural network to learn the properties of molecular structures. Moreover, a general framework for unifying and extending existing methods as blocks has been put forward [21]. The graph defined in the general framework contains node attributes, edge attributes and global attributes. Node, edge, and graph features all update simultaneously. MEGNet [22] uses a defined graph to extend the general framework for predicting state-dependent properties, i.e., the free energy.

In traditional graph convolutional networks, the information of a graph structure is represented by the adjacency matrix and the matrix size is fixed during training; however, the number of atoms in different molecules is diverse. Therefore, how to choose a suitable matrix size is a hard nut to crack. A recurrent neural network such as the RNN, long short-term memory (LSTM) and gated recurrent unit (GRU) has an outstanding ability to handle different sized sequences. These networks are good at handling temporal sequencing problems such as natural language processing [23], [24] and even spatiotemporal graphs [25]. Since general graphs such as social networks and chemical compounds are nonsequential structures, some studies tried to convert graphs into sequences, which could avoid the matrix size problem. Zayats and Ostendorf [26] proposed a Graph-Structured LSTM in which each node in the tree was the input of each single LSTM unit. Jin and JaJa [27] used the random walk approach to sample graph node sequences and extended the RNN models to graph representations. For graphs, neighbors play vital roles in the features of the current node. Teney *et al.* [28] used GRU to update the node state, and neighbors' information could be passed to the center node through multiple iteration. Furthermore, the recurrent neural network has been widely used in molecule generation. Segler et al. trained an RNN as a generative model, and the generated atoms could be added one by one according to the target properties. In addition, GraphRNN [29] divided a graph into a sequence and learned to generate a new graph based on a sequence. These studies verify that recurrent neural networks have the ability to handle graphs. However, in the original graphs, there is no sequential information. Therefore, defining the node ordering is important to ensure the accuracy and robustness of the graph-based RNN model.

Although graph-based deep learning methods have achieved good results for predicting molecular properties, there are still some problems that need to be solved.

First, the atoms in a molecule are not independent of each other. In the molecular graph structure, atoms are regarded as nodes and bonds are regarded as edges. Different from the graph of a social network [30]–[32], where each node is independent, there are common pairs of electrons between the atoms in a molecule, which are chemical bonds, causing the connections between atoms to be constrained by the valence. Directly treating atoms as basic units of graphs is not conducive to maintaining molecular chemical information since some atoms compose the substructure with chemical characteristics, i.e., the benzene ring. If the atoms of the benzene ring are decomposed into several independent atoms, the connections between atoms in the benzene ring are hard to obtain, which will also destroy the properties of benzene rings.

Second, the chemical environments of each substructure determine the connections with other substructures, which are also important factors of molecular properties. Although JunctionTree [33] transformed a molecule into a joint tree to retain the substructure information, it did not discover the internal information between substructures in the tree structure, which limited the chemical property prediction performance.

Third, a molecular graph is structured data in which different substructures may have various contributions to the properties. Summation and average operations could not capture this characteristic. An attention mechanism [34] addressed concerns regarding weighing different parts of the input to make decisions. Although there are several studies focusing on the contributions of different nodes to the graph [35], [36] for molecules, we are more concerned with the role of the different functional groups on the overall molecular properties.

To address above problems, we propose a multichannel substructure-graph gated recurrent unit architecture with attention mechanisms to learn and predict molecular properties. The three main contributions of this paper are highlighted as follows.

1) A molecule is transformed into a substructure-based graph called the S-Graph, which is inspired by JunctionTree [33]. The basic unit is a molecular substructure, which contains several linked atoms and bonds. Each substructure is regarded as a node in the graph, which could preserve the chemical properties. The fine-grained and coarse-grained features of molecules at the node level and molecule level are learned from the S-Graph, which generates a more comprehensive description of the molecule.

2) The joint feature is introduced to capture the detailed connectivity between substructures, which utilizes the precise linking position to obtain not only the inside topology of

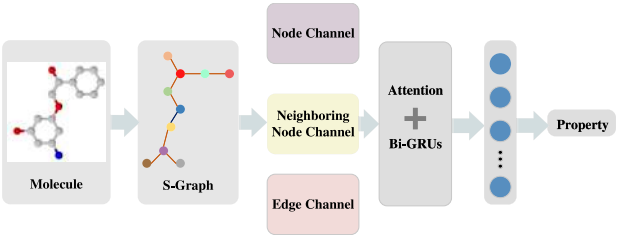

FIGURE 1. The architecture of MSGG.

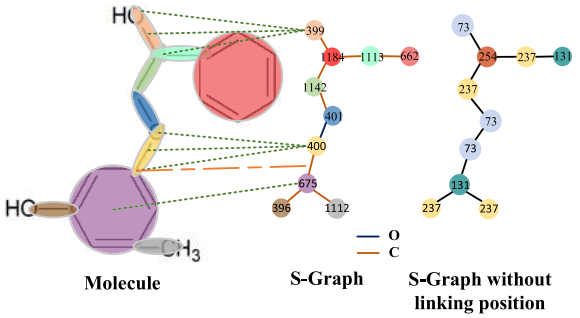

FIGURE 2. The composition of the nodes in the S-Graph and the corresponding substructures in molecules.

the substructure but also the information of the topological connections between substructures.

3) The bidirectional gated recurrent units (Bi-GRUs) and an attention mechanism are combined on three channels to extract the features of different aspects of molecules in order to mine the deep information for molecular property prediction. Three GRUs are adopted to extract the features of the node channel, neighboring node channel and edge channel to generate molecular representations. Different attention weights are assigned to the entities in the sequence to evaluate their contributions.

II. METHODS

In this paper, we propose a multichannel substructure-graph GRU (MSGG) algorithm, which is a novel GRU-based neural network with attention mechanisms applied to molecular substructures to learn and predict properties. In this architecture, substructures are generated to capture the fine-grained molecular information containing several linked atoms and bonds, and the node and edge features within the substructure are learned through neural networks, which are called node-level features. Each substructure is regarded as a node and the shared atoms are regarded as edges in the S-Graph, and the coarse-grained information including the nodes, neighboring nodes and edge channels of molecules is captured at the molecule level. Three Bi-GRUs are adopted to extract the features on three channels to generate the molecular representations. Different attention weights are assigned to entities to evaluate their contributions. The outputs of the three channels are concatenated as the final molecular descriptors, and a simple multilayer perceptron (MLP) is applied to learn the molecular properties. The main architecture is shown as Fig. 1.

A. S-GRAPH

At first, each molecule is divided into multiple substructures as basic nodes of the molecular graph the same as JunctionTree [33] considering their chemical properties. In a molecule, shared electron pairs are expressed as a bond between connected atoms, and the bond is regarded as a fundamental entity in the substructure. In the original molecule, each bond and the atoms linked to the bond are assembled into a substructure. If the atom owns more than one bond, it belongs to more than one substructure. Fig. 2 shows the corresponding relationship between the substructures in the molecule and the nodes in the S-Graph with the same color. The shared atom becomes an edge that connects two substructures in a new substructure-graph. Moreover, if there are three or more nodes connected to the same atom, an extra node (1184 in Fig. 2) is added to represent the atom to avoid the dead loop problem of message passing.

It is important to assign a label to the new substructure. If two substructures are identical, including the internal types of atoms and bonds, topology, and the connection position with the adjacent substructure, these two substructures are regarded as one type and will be assigned same label in new graph. The original molecule is represented as a graph $S_G = (S, E, C)$ in which the nodes correspond to substructures $S = \{s_1, s_2 \ldots s_n\}$, edges $E = \{e_1, e_2, \ldots e_m\}$ correspond to the shared atoms between two nodes, and n and m are the numbers of substructures and edges, respectively. Assigning a type to an edge could preserve the potential connectivity of two adjacent nodes. C is a collection containing all nodes appearing in the molecule dataset and each node owns an index in C.

As for the molecular decomposition, the condition for classifying the substructures in JunctionTree [33] is that the entities (atoms and bonds) belonging to the two substructures are identical, which cannot sufficiently distinguish the substructures. In this paper, in order to capture the detailed distinction between substructures, the linking position of the substructure is introduced. Even if entities within two structures are the same, it is possible that they belong to different categories because of the different linking positions. For instance, the nodes labeled as 662 and 675 both correspond to the benzene ring in the molecule shown in Fig.2. The entities including atoms and bonds belonging to these two benzene rings are identical, which will be regarded as the same node 131 without a linking position. However, the linking positions indicating the atomic unsaturated valence of them are different. In addition, the linking positions represent the potential positions to connect with other substructures. Therefore, these two benzene rings are labeled by different indexes in the S-Graph.

B. NODE-LEVEL FEATURES

After obtaining the S-Graph, it is important to extract the features of each node for further processing, which is called

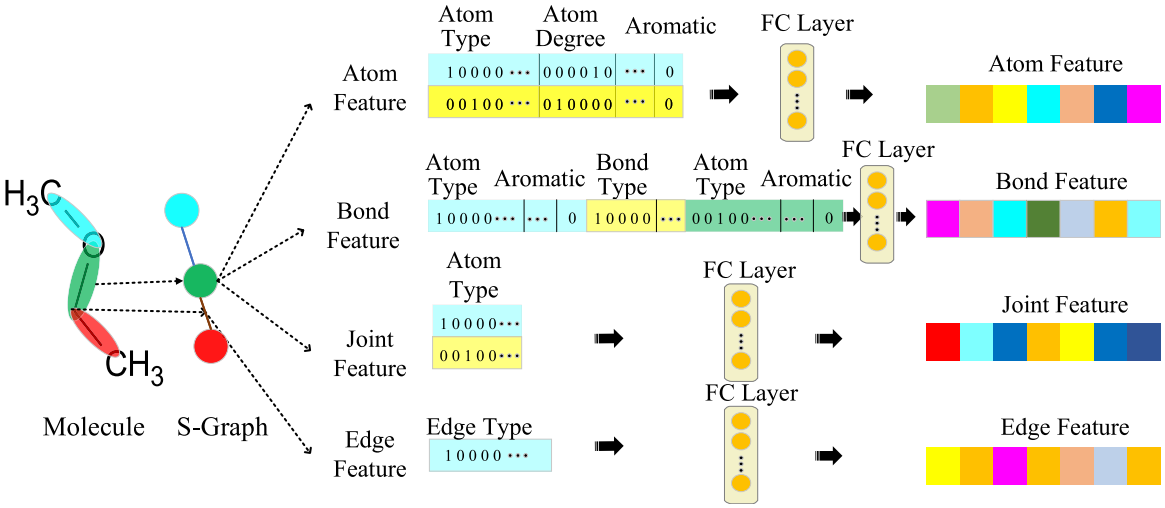

FIGURE 3. Node-level features.

the node-level features. There are three kinds of features: the atom feature, the bond feature and the joint feature. The node-level feature records the internal features of substructures in order to capture the fine-grained features for molecular substructures. Moreover, the edge feature is also introduced to discover the relationship with each node. Fig. 3 describes the four types of feature extraction methods at the node level.

1) ATOM FEATURE

At first, the one-hot encodings of the properties of the atomic element of the substructure are concatenated including the atom type, the atom degree, the number of attached hydrogen atoms, and the chiral and aromatic characteristics. In addition, each feature is transformed into a vector through a fully connected layer, which is shown in Fig. 3. Because there is usually more than one atom in a node, all node features are combined through the sum operation using (1) to obtain the final atom feature.

$$X_a(s) = \sum_{a \in s} \sigma(W_A \times f_a(a) + b_a) \tag{1}$$

where a is the atom belonging to substructure s and $f_a(a)$ is the one-hot encoding of the properties of atom a. W_A is the learned weights for the atomic neural network, σ is the activation function, and b_a is the bias.

2) BOND FEATURE

Similar to the atom feature, the one-hot encodings of the properties of bond d including the bond type, stereo descriptor, and aromatic and ring characteristics are concatenated as feature $f_d(d)$. In addition, the features of two connected atoms $f_a(a_1^d)$ and $f_a(a_2^d)$ are also concatenated, and a linear transformation is used to obtain the final bond representation X_d through (2).

$$X_d(s) = \sum_{d \in s} \sigma(W_D \times [f_a(a_1^d), f_d(d), f_a(a_2^d)] + b_d) \tag{2}$$

where [,] is the concatenation operation and d is the bond belonging to s. W_D is the learnable weights for the bond neural network, and b_d is the bias.

3) JOINT FEATURE

The joint feature indicates the link position of one node to others, where one node is a substructure composed of several atoms and bonds. Even if the types of atoms and bonds of two nodes are the same, there may be potential differences in the properties due to different linking positions of nodes. The type of atom a that links the atoms belonging to other nodes are recorded as one-hot encoding $f_j(a)$. A neural network is applied to obtain the feature vector X_j that captures the linking features of node through (3).

$$X_j(s) = \sum_{a \in N(s)} \sigma(W_J \times f_j(a) + b_j) \tag{3}$$

where $N(s)$ is set of all linking atoms of node s, W_J is the learnable weights, and b_j is the bias.

Finally, all these three features including atom feature, bond feature and joint feature are combined into the final feature of node.

$$X_o(s) = [X_a(s), X_d(s), X_j(s)] \tag{4}$$

4) EDGE FEATURE

After extracting the features of each node, the molecule can be regarded as a graph structure whose basic unit is a node. Edge information is also important in the graph, and the different types of edges e in the S-Graph are distinguished by their shared atomic types, which are encoded as

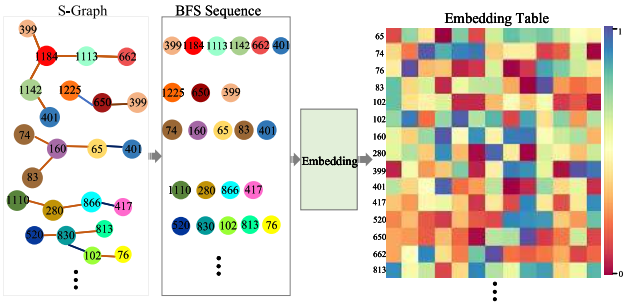

FIGURE 4. Node embedding features.

one-hot vector $f_e(e)$, and the edge feature $X_e(e)$ is described as follows:

$$X_e(e) = \sigma(W_E \times f_e(e) + b_e) \tag{5}$$

where W_E is learnable weights, and b_e is the bias.

5) EMBEDDING FEATURE

To capture the information on the similarity between each node, the embedding feature is introduced. Each node owns an index in vocabulary C, which could be regarded as a word. Inspired by the word2embedding method [37], each node could obtain an embedding vector that represents the attributes of the node in a molecule. Therefore, a node-embedding layer is implemented, which maps a node vocabulary into an embedding matrix $W_G \in R^{d_n \times d_c}$, where d_n is the dimension of the embedding feature and d_c is the size of vocabulary C. The embedding matrix is trained based on the connectivity of the nodes in the S-Graph. As shown in Fig. 4, all the molecules in the dataset are transformed into an S-Graph with the nodes index. After processing the embedding, each node obtains a vector indicating its attributes. If several nodes link to one same node in different molecules, the value of the specified attribute will be close. The node embedding that is to be trained as a part of model focuses on the similarity of the nodes' attributes in molecules. After obtaining the embedding matrix, given the index of a node, its embedding is calculated through (6), where id(s) returns the ID number of node s.

$$X_g(s) = W_G(\mathrm{id}(s)) \tag{6}$$

C. MOLECULE-LEVEL FEATURES

We proposed a two-level structure to explore the comprehensive representation of a molecule. After obtaining each feature of each substructure, the molecule-level feature is extracted. Most algorithms focus on extracting the features from the whole structure of the molecule [8], [10], [12]. However, a molecule is a special graph in which different substructures interact with each other. In addition, different molecules have diverse sizes, which is inconvenient for the traditional graph convolutional network. To provide a size-free method, the bidirectional GRU that could learn the relationships between substructures is applied in this paper.

The molecule-level feature of the S-Graph is extracted from three channels: the node channel, the neighboring node channel and the edge channel, as shown in Fig. 5. All nodal information is processed through the node channel, which includes the basic structure of the molecule graph. The neighboring node channel gathers information on the direct neighbors of the centered nodes, which is used to cover more comprehensive property features. The edge channel collects all edge information. Our model can fit to the message passing block and independent recurrent block in [21]. The molecules in the dataset are stored in SMILES form and the beginning character of SMILES is fixed for each molecule [38]. When the SMILES is converted to a graph structure, the node containing the beginning atom in SMILES is set as the root, which ensures that the root node is fixed in the S-Graph. BFS is applied to traverse the S-Graph starting from the root node to generate the node sequence. In the S-Graph, each node is labeled with the index. When BFS searches from the current depth level to the next depth level, all nodes at the next depth level are sorted by their index in ascending order. The traversing order of nodes is the final node order in the sequence.

1) NODE CHANNEL

The feature vectors of all nodes constitute the sequence $M^{(1)}$ through (7).

$$M^{(1)} = X_o(s_1), X_o(s_2), X_o(s_3), \ldots, X_o(s_n) \tag{7}$$

The node order in the sequence is determined through the BFS method and the sequence length is equal to the number of nodes in a molecule.

2) NEIGHBORING NODE CHANNEL

In this channel, not only the node itself is included but also the direct neighbors of the node are included to introduce wider and comprehensive node features. Connected nodes share a common atom, which is described as edge e in the S-Graph. For a node in the S-Graph, all its direct neighbors and their linked edge information are aggregated in the neighboring node channel through (8). The length of the neighboring node sequence is equal to number of nodes in a molecule.

$$M^{(2)} = H(s_1), H(s_2), \ldots, H(s_n) \tag{8}$$

$$H(s) = [X_o(s), \sum_{s' \in h(s)} [X_e(e^{s,s'}), X_o(s')]] \tag{9}$$

where $M^{(2)}$ is the neighboring node channel of the whole molecule, $h(s)$ is set of direct neighbors of node s, and $e^{s,s'}$ is the edge between s and s'.

3) EDGE CHANNEL

The edge channel contains the edge information, and it also covers the two node embedding features linked to this edge. The edge channel pays more attention to the global molecule architecture. We want to extract information about the overall properties of the molecule in terms of the edges. Therefore,

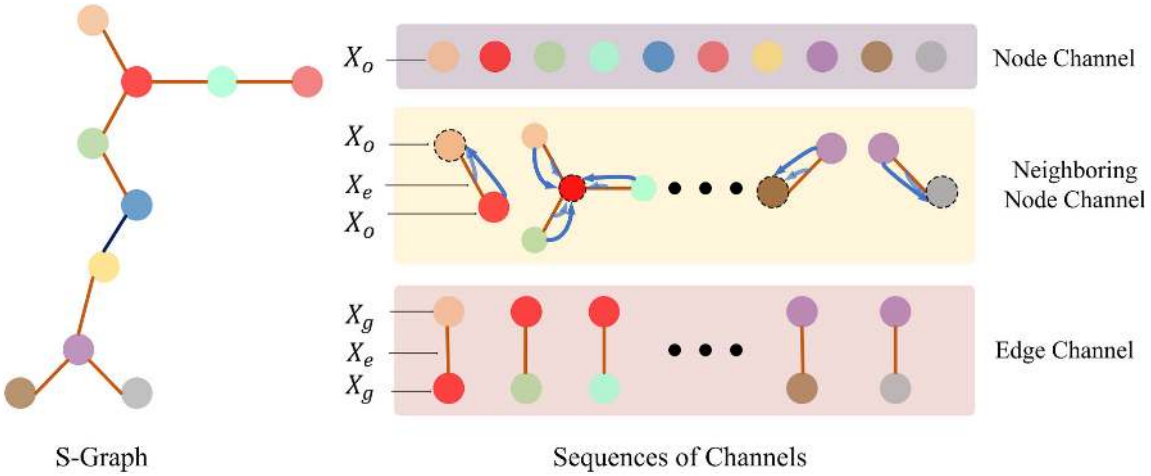

FIGURE 5. Three channels of the molecule-level feature.

the node embedding vector that records the node attributes in molecules is used in the edge channel. The item in the edge sequence is depicted as $E(e)$ in (11). All the edge items in a molecule constitute $M^{(3)}$ in the node order. The length of $M^{(3)}$ is equal to the number of edges in a molecule.

$$M^{(3)} = E(e_1), E(e_2), \ldots, E(e_n) \tag{10}$$

$$E(e) = [X_g(s_1^e), X_e(e), X_g(s_2^e)] \tag{11}$$

where s_1^e and s_2^e are the two nodes linked by the edge e.

D. MODEL STRUCTURE

Different molecules vary in their numbers of atoms and edges, which results in different numbers of nodes in graphs. Therefore, the multilayer bidirectional gated recurrent units (Bi-GRUs) [39] is adopted for three channels, which could be adapted to different numbers of nodes. Considering the connections between nodes, a graph could be transformed into a node sequence following BFS. The input vectors of the three channels are in order of their node sequence. A standard two-layer property prediction is conducted because the model architecture is bidirectional. One item corresponds to two outputs in the GRU model structure, which could capture the bidirectional information for molecules. If the input sequence has n items, there will be $2n$ outputs. The calculation of each single direction layer GRU is as follows:

$$r_t = \sigma(W_r x_t + b_r + W_r' h_{(t-1)}) + b_r') \tag{12}$$

$$z_t = \sigma(W_z x_t + b_z + W_z' h_{(t-1)}) + b_z') \tag{13}$$

$$p_t = \tanh(W_p x_t + b_p + r_t(W_p' h_{(t-1)} + b_p') \tag{14}$$

$$h_t = (1 - z_t)p_t + z_t h_{(t-1)} \tag{15}$$

where x_t is the item input at position t of the sequence and $h_{(t-1)}$ is the hidden state of the layer at position $t-1$. r_t, z_t, and p_t are the reset, update, and new gates, respectively. W_r, W_z, and W_p are the parameters for input x_t. b_r, b_z, b_p are the bias for input x_t. W_r', W_z', and W_p' are the parameters for input $h_{(t-1)}$. b_r', b_z', and b_p' are the bias for input $h_{(t-1)}$. σ is the sigmoid function. h_t is the hidden state at position t.

Our model is composed of two stacked Bi-GRUs. Each layer is calculated as follows:

$$h_t^{(l)} = f(W_h^{(l)} h_{(t-1)}^{(l)} + W_i^{(l)} h_t^{(l-1)} + b^{(l)}) \tag{16}$$

where $h_t^{(l)}$ is the hidden state of layer l at position t. $h_{(t-1)}^{(l)}$ is the hidden state of layer l at position $t-1$. $W_h^{(l)}$ and $W_i^{(l)}$ are the parameters for layer l. $h_t^{(l-1)}$ is the hidden state of layer $l-1$ at position t.

The final output is represented as follows:

$$Y_h = h_1^{(2)} h_2^{(2)} \ldots h_n^{(2)} h'^{(2)}_1 h'^{(2)}_2 \ldots h'^{(2)}_n \tag{17}$$

where $h_n^{(2)}$ and $h'^{(2)}_n$ indicate the bidirectional outputs at position n of the second layer. Y_h with size $2n$ is the output of the two-layer bidirectional GRU.

The features of all items are aggregated with different attention weights to obtain the final molecular graph feature. The molecule is assigned an initial value with the average values $\bar{Y}$ of all features. The assignments of the attention weights are calculated by the feature similarity between each item and $\bar{Y}$, which is calculated as (19).

$$\bar{Y} = \frac{\sum_{i=1}^{n} Y_i}{n} \tag{18}$$

$$\alpha_i = \frac{\exp(\text{LeakyReLU}(W_t \times [W(\bar{Y}), W(Y_i)]))}{\sum_{i=1}^{n} \exp(\text{LeakyReLU}(W_t \times [W(\bar{Y}), W(Y_i)]))} \tag{19}$$

where $Y_i \in R^F$ is the feature of the i^{th} sequence item. $\bar{Y} \in R^F$ is the average feature of all nodes in the S-Graph, and $W(Y_i)$ is the linear transformation of Y_i. $W_t \in R^{2F}$ is the parameter of a simple feedforward neural network. LeakyReLu is the activation function. α_i is the attention weight for the i^{th} item.

The final graph output is as follows:

$$G_f(Y) = \sigma(\sum_{i=1}^{n} \alpha_i \times W(Y_i)) \tag{20}$$

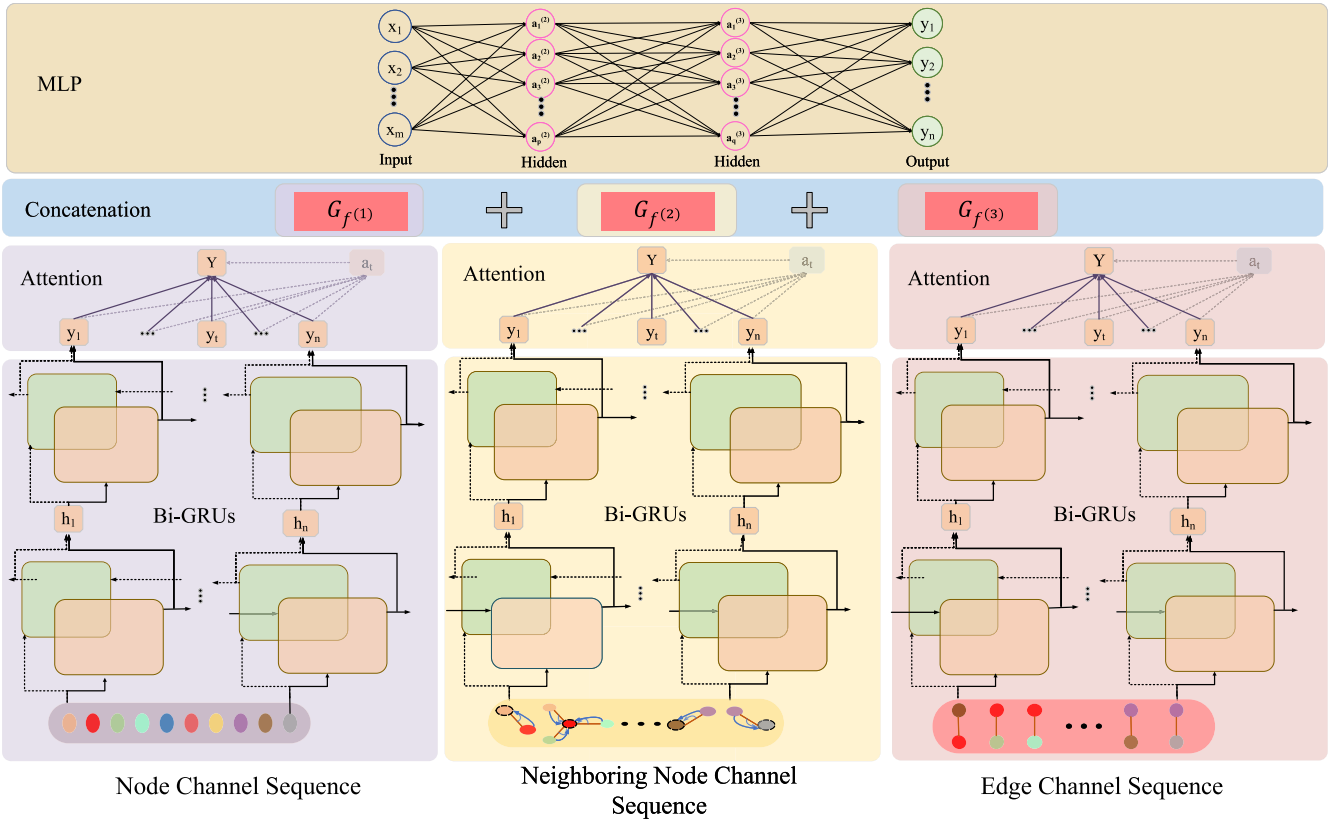

FIGURE 6. Bi-GRUs for extracting molecular feature.

where $G_f(Y)$ is the final feature of input sequence Y in a single channel. $W(Y_i)$ is the linear transformation of the i^{th} item's feature Y_i and α_i is the corresponding attention weight.

Three input sequences of channels are fed into the three layer bidirectional GRU neural network and attention model separately. The whole molecular feature gathering information of the three channels is represented as follows:

$$G_u = [G_{f(1)}, G_{f(2)}, G_{f(3)}] \tag{21}$$

where $G_{f(1)}$ is the node channel feature. $G_{f(2)}$ is the neighboring node channel feature. $G_{f(3)}$ is the edge channel feature and G_u is the whole molecular feature.

To learn the molecular properties, a multilayer perceptron is applied on the learned molecular descriptors for the prediction and classification. The architecture is shown in Fig. 6.

III. DATASET

To objectively prove the advantages of the proposed model, multiple datasets with different properties are adopted for regression and classification tasks.

A. REGRESSION TASK

Four datasets are prepared for the regression task with root-mean-square error (RMSE) metric. The Free Solvation Database is a dataset of experimental and calculated hydration free energies for small neutral molecules in water [40] and has a size of 642. ESOL is a dataset containing the water solubility data for 1128 compounds [41]. Lipophilicity provides the experimental results of the octanol/water distribution coefficient of 4200 molecules screened from the ChEMBL database [42]. These three datasets are randomly split into training/validation/test sets as in MoleculeNet. PDBbind is a dataset that collected the experimentally measured binding affinities for protein-ligand complexes [43], [44], and has a size of 9880 (since 1982). The complexes in PDBbind are updated over time, and are split into training/validation/test sets by time.

B. CLASSIFICATION TASK

The BACE dataset and Blood-Brain Barrier Penetration (BBBP) dataset are used to evaluate the performance of the proposed model. The BACE dataset measures the binding results for the inhibitors of human β-secretase 1 (BACE-1) [45]. MoleculeNet [46] provided a collection of 1552 compounds with binary labels for the classification task from the BACE dataset. The BBBP dataset collected data from studies on the modeling and prediction of the barrier permeability [47], and it contains 2053 compounds with binary labels for the permeability properties. We adopted the same splitting method as MoleculeNet to compare the results with other state of the art methods. The BACE and BBBP datasets were split using the scaffold splitting method relying on the

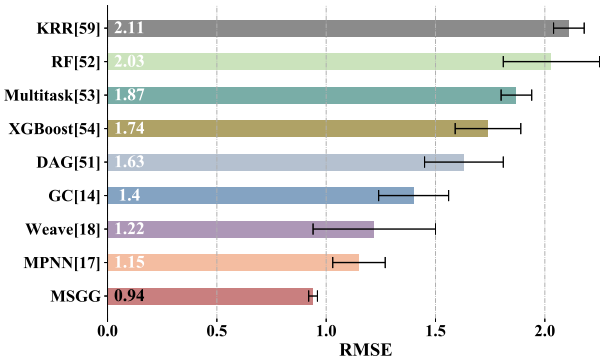

FIGURE 7. Performance of the different algorithms on FreeSolv.

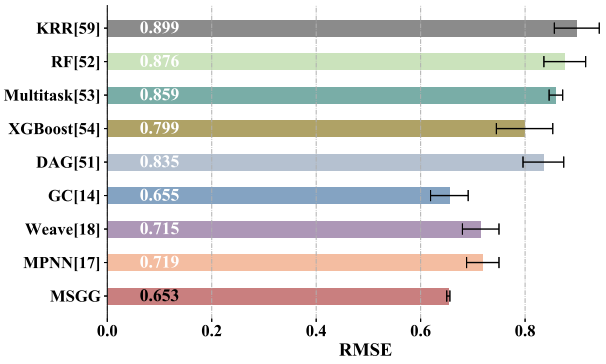

FIGURE 8. Performance of the different algorithms on ESOL.

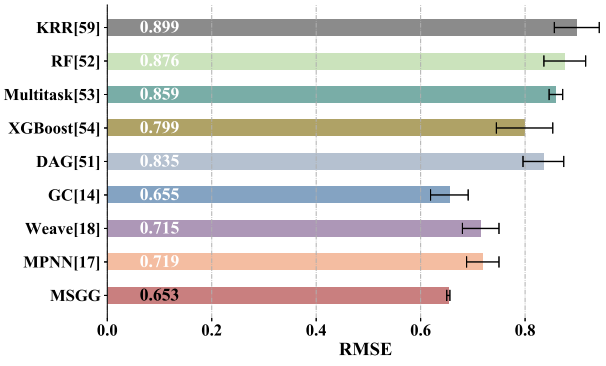

FIGURE 9. Performance of the different algorithms on Lipophilicity.

2D structure information [48] through RDKit [49]. Compared with random splitting, scaffold splitting increases the difference between the training, validation and tests dataset, which could better verify the generalizability of the models.

IV. EXPERIMENTS

All models were trained using the stochastic gradient descent (SGD) algorithm with the ADAM optimizer [50]. The initial learning rate was randomly chosen from $5e^{-4}$ to $5e^{-3}$. Different seeds were selected for all models to verify the robustness of the models and grid search was utilized for hyperparameter screening. In the experiments, we compared our model with models mentioned in MoleculeNet on these six datasets. The models not only include state of art graph-based deep learning methods such as graph convolutions (GC) [14], directed acyclic graph (DAG) models [51], weave models (Weave) [18], and message passing neural networks (MPNN) [17], but also include classic conventional algorithms such as random forests (RF) [52], multitask networks (Multitask) [53], gradient boosting (XGBoost) [54], logistic regression models (Logreg) [55], support vector machines (KernalSVM) [56], influence relevance voting (IRV) [57], bypass networks (Bypass) [58] and kernel ridge regression (KRR) [59]. All trained MSGG models are available at https://github.com/ShuangWangCN/MSGG. The results of the comparison of our algorithm and these models are given as follows, and the comparisons of the different algorithm designs are also described.

A. PERFORMANCE ON THE REGRESSION TASK

For the regression task, the performance metric is the root-mean-square error (RMSE), and a lower value represents better performance. Our model was applied to three datasets (FreeSolv, ESOL, Lipophilicity) in which the number of molecules is smaller than 5000 and the results are shown in Fig. 7- Fig. 9. It is demonstrated that our algorithm is superior to all models on the FreeSolv (0.94), ESOL (0.55) and Lipophilicity (0.653) datasets, which proves that the generalizability of our model is better than that of other methods.

To evaluate the performance of our model on a large dataset in which the number of molecules is close to 10000, an experiment was carried out on the PDBbind dataset and the results are shown in Fig. 10. Different from the other datasets, PDBbind, which records the binding affinities for protein-ligand complexes, contains the molecules (ligands) and the corresponding proteins. There are several methods [60]–[62] focusing on the comprehensive three-dimensional representation of a protein-ligand interaction to investigate the binding affinities, which have shown good performance. Potentialnet gathered the bonded ligand information and spatial proximity to protein atoms in one graph [60]. In [61], a CNN-based model is applied to the 3D grid representation of the protein-ligand structures. However, the 3D information is not considered in our molecular processing model. Instead, the ligands are processed with MSGG. The corresponding protein sequences are processed by embedding and the 1D CNN (convolutional neural network), which is same as GraphDTA [63]. These two representation vectors are then concatenated, and a multilayer perceptron is utilized to predict the binding affinities. Our model obtained the best performance among all methods on PDBbind. In general, our model performed the best on all four datasets on the regression task, which proves that the generalizability of our model is better than that of other methods.

B. PERFORMANCE ON CLASSIFICATION TASK

As for the classification task on the BBBP and BACE datasets in Fig. 11 and Fig. 12, the AUC-ROC metric was adopted, for which a higher value represents better performance.

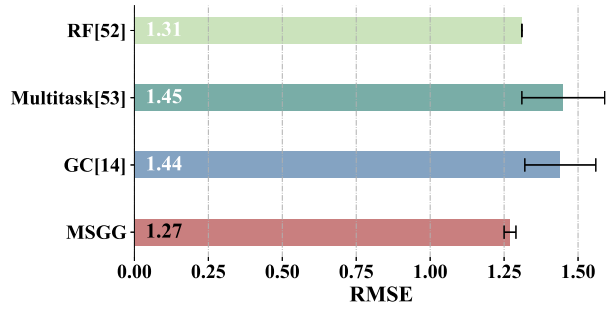

FIGURE 10. Performance of different algorithms on PDBbind.

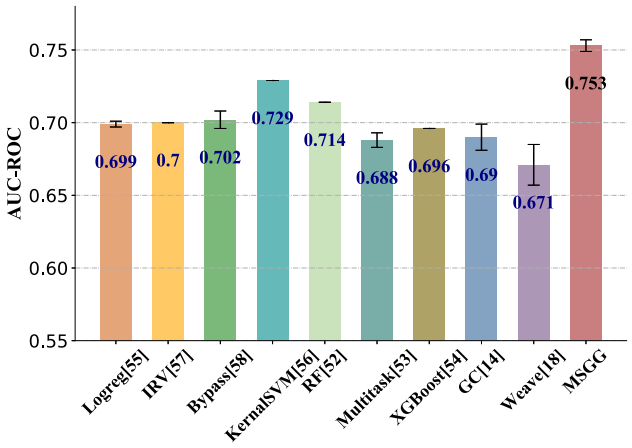

FIGURE 11. Performance of different algorithms on BBBP.

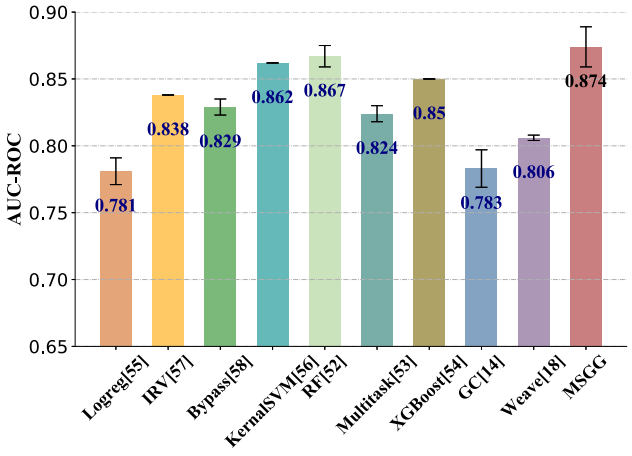

FIGURE 12. Performance of different algorithms on BACE.

BBBP and BACE are two imbalanced classification datasets in which the positive/negative weights are 3.25 and 0.84, respectively. In the training process, we assigned different weights to the positive and negative samples, which is the same as MoleculeNet. Our MSGG obtained the best performance with 0.753 on the BBBP dataset and 0.874 on the BACE dataset. On the two classification tasks, our model achieved the best results among both the graph-based deep learning methods and the conventional machine learning algorithms.

TABLE 1. Training time/epoch for the six datasets.

Dataset	FreeSolv	ESOL	lipophilicity
Time	19s	51s	339s
Dataset	PDBbind	BBBP	BACE
Time	1316s	142s	170s

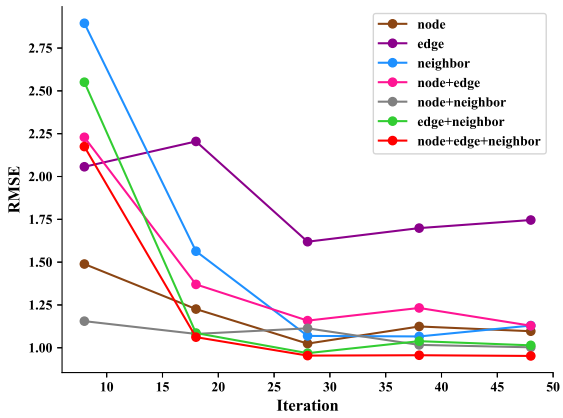

FIGURE 13. The performance of different combinations of channels.

In summary, these results indicate that our model performs well in both regression and classification tasks on many kinds of datasets, which suggests that the proposed model is robust enough to handle diverse predictions of molecular properties. To take it a step further, our model could extract the deep information of molecules. Biologically meaningful substructures are the basic units in our model, which guarantees the rationality of the feature extraction. The combination of different channels that capture the node, neighboring node and edge features, respectively, could cover richer information compared with other graph-based models, which enhances the feature discrimination. Therefore, the proposed method could perform well on complex tasks.

Our algorithm based on PyTorch was performed on a workstation with 64 GiB of RAM, an Intel(R) Xeon(R) CPU E5-2603 v4 CPU, and a TITAN Xp Graphics card. The training time of one epoch for the six datasets is shown in Table 1. The training time is highly related to the size of the dataset. It took approximately 22 minutes for one epoch on the PDBbind dataset because it is a large dataset containing 7904 molecules for training.

C. COMPARISON FOR DIFFERENT COMBINATIONS OF CHANNELS

Here, a more detailed experiment to illustrate the necessity of combining the three channels was performed. Fig. 13 illustrates the performance of different combinations of the three channels on the FreeSolv test dataset with different numbers of training iterations (8, 18, 28, 38, and 48), including three single channels, combinations of two channels and a combination of three channels. Seven models were trained with the same hyperparameters and architecture. From the figure, we could see that the RMSE gradually decreases as

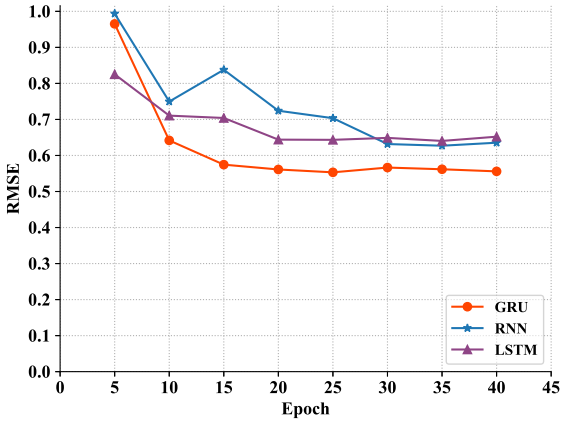

FIGURE 14. Comparison of the different recurrent neural networks on the ESOL dataset.

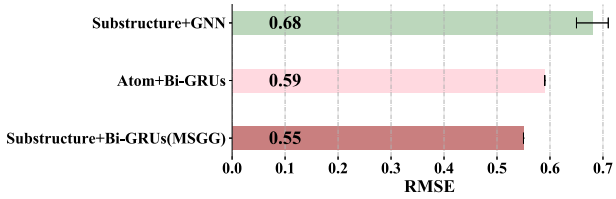

FIGURE 15. The comparison of the substructure + GNN, atom + Bi-GRUs and the proposed model.

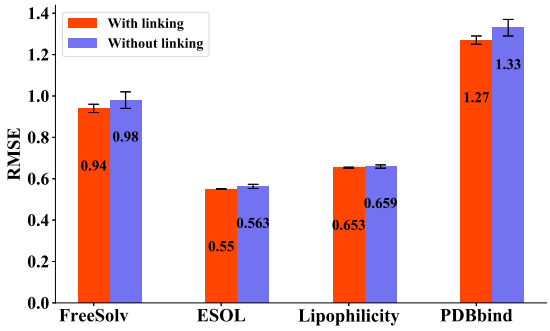

FIGURE 16. The comparison of the substructures with and without linking information on the regression task.

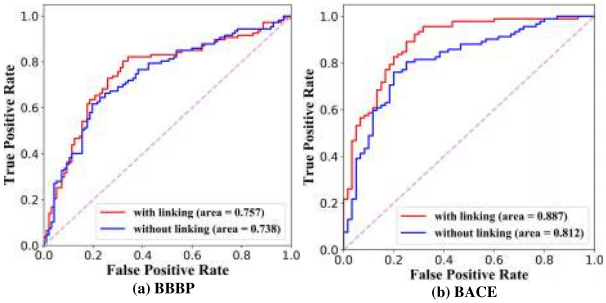

FIGURE 17. The comparison of the substructures with and without linking information on the classification task on (a) BBBP dataset and (b) BACE dataset.

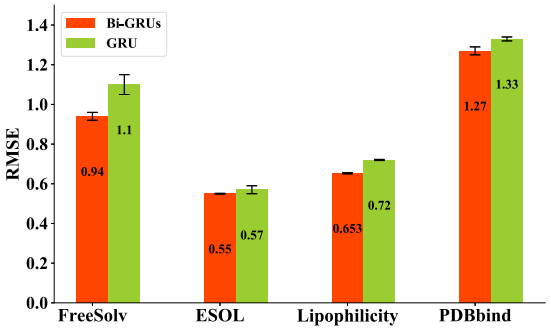

FIGURE 18. The comparison of the GRU and Bi-GRUs on the regression task.

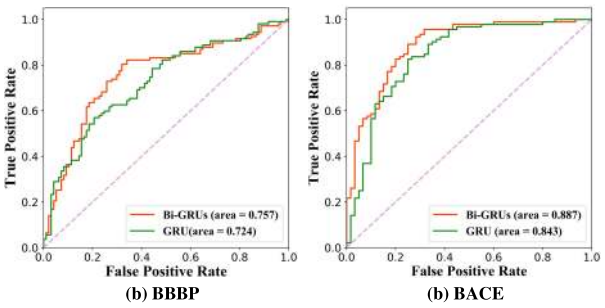

FIGURE 19. The comparison of the GRU and Bi-GRUs on the classification task on (a) BBBP dataset and (b) BACE dataset.

the training iteration increases. The convergence speed of the combination of three channels is the fastest and obtains the lowest RMSE.

D. COMPARISON WITH DIFFERENT RECURRENT NEURAL NETWORKS

As for the main model architecture, the GRU network is selected due to its ability to filter the sequence information. To verify its effectiveness, an experiment comparing it with other two widely used recurrent neural networks (the standard RNN and LSTM) was carried out. The three models were applied to the ESOL test dataset. The results are shown in Fig. 14. We trained all models with different numbers of epochs (5, 10, 15, 20, 25, 30, 35, and 40). From Fig. 14, it can be seen that GRU model performs better than the other two models as the number of epochs increases.

E. COMPARISON OF DIFFERENT COMBINATIONS OF SUBSTRUCTURES AND BI-GRUS

The molecules in our model are decomposed into substructures. These substructures consist of a new S-Graph that describes the original molecule. Node-level features are utilized to describe the basic nodes and edges in the S-Graph. The molecule-level features obtained by the Bi-GRUs are applied to describe the whole molecular properties. In this section, an experiment was performed to evaluate the combinations of substructures and Bi-GRUs. Two other models are introduced for comparison purposes. The first one is the combination of substructures with node-level features and a higher-order graph neural network [64], which is an advanced graph-based deep learning method. The second model is the combination of an atom-based molecular graph and Bi-GRUs. The results are shown in Fig. 15. From the

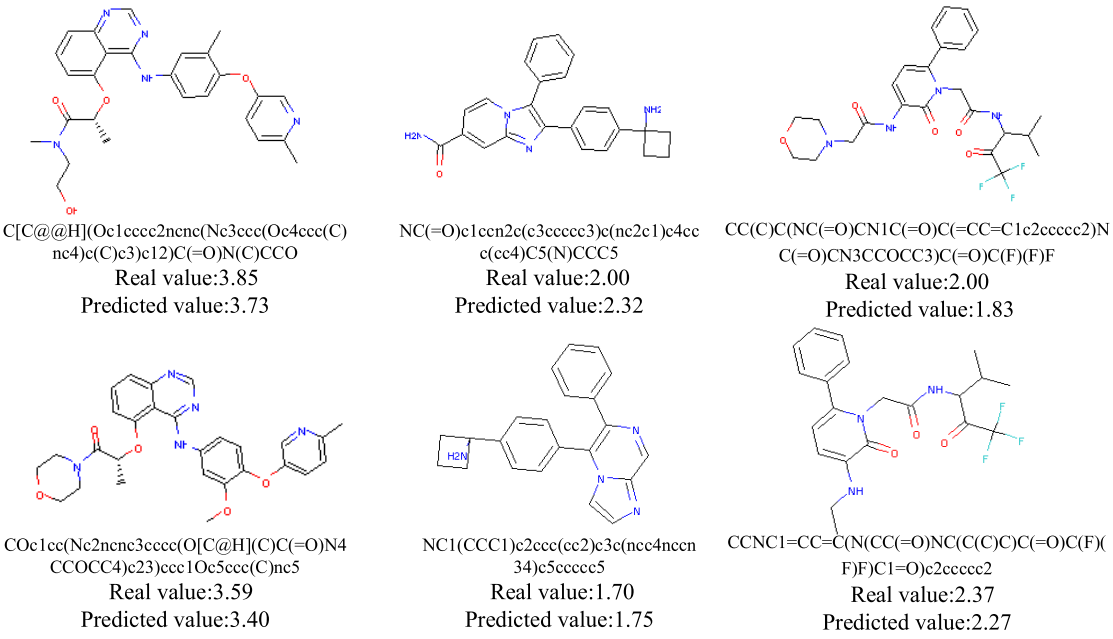

FIGURE 20. The comparison of the molecules that are similar in structure but distinct in SMILES.

comparison of Fig. 15 and Fig. 8, it is obvious that these two methods are superior to the GC and DAG which are graph-based deep learning methods, but the proposed method that combined both substructure encoding and Bi-GRUs performed better than these two.

F. COMPARISON OF SUBSTRUCTURES WITH LINKING AND WITHOUT LINKING

To improve the generalizability, the linking information is added to distinguish the different substructures. We designed an experiment to compare the performance of substructures with and without linking information. The experiment was performed on six public datasets, in which only the substructure identification methods were different and the feature extraction procedures were the same. The substructures with linking features have more categories those that without linking features. For example, there are 1254 categories (IDs) of the substructure with linking and 281 categories (IDs) of the substructure without linking in the BBBP dataset. Fig. 16 and Fig. 17 illustrate the comparison of the substructures with linking features and without linking features on the regression task with the RMSE metric and on the classification task with the AUC-ROC metric, respectively. As shown in the two figures, the substructure with linking outperformed the substructure without linking on all datasets for the regression task and on both datasets for the classification task, which proves that substructure splitting including linking information improves the performance of MSGG. The linking feature of nodes represents chemical information in the molecule. Although the atoms and bonds belonging to two nodes are same, they may connect with different nodes if the linking information of the two nodes is different. The potential connectivity possibilities between nodes could be learned in the training process.

G. COMPARISON OF GRU AND BI-GRUS

In the training and prediction process, the bidirectional GRUs was adopted considering that there are dependencies between molecular substructures, and bidirectional connections between substructures are helpful to discover the relationship between the adjacent nodes in a graph. The comparison between the GRU and Bi-GRUs was implemented and the results are shown in Fig. 18 and Fig. 19. The Bi-GRUs model outperformed the GRU model on all four regression datasets and the two classification datasets, which demonstrates that the design of the bidirectional GRU is the best because the connections between the nodes in molecules are bidirectional, and the standard GRU could not capture the complete connectivity information.

H. COMPARISON FOR MOLECULES WITH SIMILAR STRUCTURES BUT DISTINCT SMILES

Three pairs of molecules are selected from the Lipophilicity test dataset, which provides experimental results on the octanol/water distribution coefficient (logD at pH 7.4). For each pair, the two molecules are very similar in structure but very distinct in SMILES. The visualizations of the three-pair molecules with the SMILES and logD values are displayed

in Fig. 20. The real experimental logD of the molecule and the average prediction of our trained models are also labeled. It can be observed that the experimental logD values of the two molecules in each pair have little difference, which suggests that the logD value is affected by the structure. Moreover, the predicted logD values are close to the ground truth, which demonstrates that our model could capture the molecular structures.

V. CONCLUSION

In this paper, we proposed a novel MSGG architecture based on molecular substructures. A new S-Graph that decomposes molecules into substructures was put forward. Features were extracted both at the node level and molecule level, which capture the fine-grained and coarse-grained connectivity information of molecules. Three Bi-GRUs were adopted on three channels for molecules to cover more elaborate information. The experiments on both the regression and classification tasks on different datasets support the generalizability and robustness of our model, which outperformed state of the art algorithms for molecular property predictions. Molecules are special graph-structured data containing much chemical information, and molecular functional groups play important roles in molecular properties. In the future, we will aim to explore the molecular properties based on molecular functional groups utilizing deep learning methods.

REFERENCES

[1] F. Montanari, L. Kuhnke, A. Ter Laak, and D.-A. Clevert, "Modeling physico-chemical ADMET endpoints with multitask graph convolutional networks," *Molecules*, vol. 25, no. 1, p. 44, Dec. 2019.
[2] E. N. Feinberg, R. Sheridan, E. Joshi, V. S. Pande, and A. C. Cheng, "Step change improvement in admet prediction with PotentialNet deep featurization," Mar. 2019, *arXiv:1903.11789*. [Online]. Available: https://arxiv.org/abs/1903.11789
[3] H. Chen and Z. Zhang, "Prediction of drug-disease associations for drug repositioning through drug-miRNA-disease heterogeneous network," *IEEE Access*, vol. 6, pp. 45281–45287, 2018.
[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[5] B. Sanchez-Lengeling and A. Aspuru-Guzik, "Inverse molecular design using machine learning: Generative models for matter engineering," *Science*, vol. 361, no. 6400, pp. 360–365, Jul. 2018.
[6] W. Jeon and D. Kim, "FP2VEC: A new molecular featurizer for learning molecular properties," *Bioinformatics*, vol. 35, no. 23, pp. 4979–4985, Dec. 2019.
[7] M. Heinonen, H. Shen, N. Zamboni, and J. Rousu, "Metabolite identification and molecular fingerprint prediction through machine learning," *Bioinformatics*, vol. 28, no. 18, pp. 2333–2341, Sep. 2012.
[8] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Model.*, vol. 28, no. 1, pp. 31–36, Feb. 1988.
[9] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse, "Reoptimization of MDL keys for use in drug discovery," *J. Chem. Inf. Comput. Sci.*, vol. 42, no. 6, pp. 1273–1280, Nov. 2002.
[10] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *J. Chem. Inf. Model.*, vol. 50, no. 5, pp. 742–754, May 2010.
[11] J. Ma, N. Wang, and B. Xiao, "Semi-supervised classification with graph structure similarity and extended label propagation," *IEEE Access*, vol. 7, pp. 58010–58022, 2019.
[12] D. Li, Y. Cheng, X. Wang, and Q. Yu, "Incremental graph embedding based on spatial-spectral neighbors for hyperspectral image classification," *IEEE Access*, vol. 6, pp. 10996–11006, 2018.
[13] F. Scarselli, M. Gori, A. Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[14] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. NIPS*, Montreal, QC, Canada, 2015, pp. 2224–2232.
[15] Y. Xu, J. Pei, and L. Lai, "Deep learning based regression and multiclass models for acute oral toxicity prediction with automatic chemical feature extraction," *J. Chem. Inf. Model.*, vol. 57, no. 11, pp. 2672–2685, Nov. 2017.
[16] C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola, and K. F. Jensen, "Convolutional embedding of attributed molecular graphs for physical property prediction," *J. Chem. Inf. Model.*, vol. 57, no. 8, pp. 1757–1772, Aug. 2017.
[17] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 1263–1272.
[18] S. Kearnes, K. Mccloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: Moving beyond fingerprints," *J. Comput.-Aided Mol. Des.*, vol. 30, no. 8, pp. 595–608, Aug. 2016.
[19] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," Nov. 2015, *arXiv:1511.05493*. [Online]. Available: https://arxiv.org/abs/1511.05493
[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," Sep. 2016, *arXiv:1609.02907*. [Online]. Available: https://arxiv.org/abs/1609.02907
[21] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," Jun. 2018, *arXiv:1806.01261*. [Online]. Available: https://arxiv.org/abs/1806.01261
[22] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, "Graph networks as a universal machine learning framework for molecules and crystals," *Chem. Mater.*, vol. 31, no. 9, pp. 3564–3572, May 2019.
[23] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, Chiba, Japan, Sep. 2010, pp. 1045–1048.
[24] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, Montreal, QC, Canada, 2014, pp. 3104–3112.
[25] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 5308–5317.
[26] V. Zayats and M. Ostendorf, "Conversation modeling on reddit using a graph-structured LSTM," *Trans. Assoc. for Comput. Linguistics*, vol. 6, pp. 121–132, Dec. 2018.
[27] Y. Jin and J. F. JaJa, "Learning graph-level representations with recurrent neural networks," May 2018, *arXiv:1805.07683*. [Online]. Available: https://arxiv.org/abs/1805.07683
[28] D. Teney, L. Liu, and A. V. D. Hengel, "Graph-structured representations for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3233–3241.
[29] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec, "GraphRNN: Generating realistic graphs with deep auto-regressive models," Feb. 2018, *arXiv:1802.08773*. [Online]. Available: https://arxiv.org/abs/1802.08773
[30] C. Zang, P. Cui, and C. Faloutsos, "Beyond Sigmoids: The NetTide model for social network growth, and its applications," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA, 2016, pp. 2015–2024.
[31] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "DeepInf: Modeling influence locality in large social networks," in *Proc. 24nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., 2018, pp. 2110–2119.
[32] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, London, U.K., 2018, pp. 974–983.
[33] W. Jin, R. Barzilay, and T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," Feb. 2018, *arXiv:1802.04364*. [Online]. Available: https://arxiv.org/abs/1802.04364
[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," Oct. 2017, *arXiv:1710.10903*. [Online]. Available: https://arxiv.org/abs/1710.10903

[36] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," Mar. 2018, *arXiv:1803.03735*. [Online]. Available: https://arxiv.org/abs/1803.03735

[37] Y. Goldberg and O. Levy, "word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," Feb. 2014, *arXiv:1402.3722*. [Online]. Available: https://arxiv.org/abs/1402.3722

[38] D. Weininger, A. Weininger, and J. L. Weininger, "SMILES. 2. Algorithm for generation of unique SMILES notation," *J. Chem. Inf. Model.*, vol. 29, no. 2, pp. 97–101, May 1989.

[39] D. Zhang, L. Tian, M. Hong, F. Han, Y. Ren, and Y. Chen, "Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification," *IEEE Access*, vol. 6, pp. 73750–73759, 2018.

[40] D. L. Mobley and J. P. Guthrie, "FreeSolv: A database of experimental and calculated hydration free energies, with input files," *J. Comput.-Aided Mol. Des.*, vol. 28, no. 7, pp. 711–720, Jul. 2014.

[41] J. S. Delaney, "ESOL: Estimating aqueous solubility directly from molecular structure," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 3, pp. 1000–1005, May 2004.

[42] A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, and E. Cibrián-Uhalte, "The ChEMBL database in 2017," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D945–D954, 2016.

[43] R. Wang, X. Fang, Y. Lu, C.-Y. Yang, and S. Wang, "The PDBbind Database: Methodologies and updates," *J. Med. Chem.*, vol. 48, no. 12, pp. 4111–4119, Jun. 2005.

[44] R. Wang, X. Fang, Y. Lu, and S. Wang, "The PDBbind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures," *J. Med. Chem.*, vol. 47, no. 12, pp. 2977–2980, Jun. 2004.

[45] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny, "Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches," *J. Chem. Inf. Model.*, vol. 56, no. 10, pp. 1936–1949, Oct. 2016.

[46] Z. Wu, B. Ramsundar, E. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: A benchmark for molecular machine learning," *Chem. Sci.*, vol. 9, no. 2, pp. 513–530, Oct. 2017.

[47] I. F. Martins, A. L. Teixeira, L. Pinheiro, and A. O. Falcao, "A Bayesian approach to in silico blood-brain barrier penetration modeling," *J. Chem. Inf. Model.*, vol. 52, no. 6, pp. 1686–1697, Jun. 2012.

[48] G. W. Bemis and M. A. Murcko, "The properties of known drugs. 1. Molecular frameworks," *J. Med. Chem.*, vol. 39, no. 15, pp. 2887–2893, Jan. 1996.

[49] G. Landrum. (2016). *RDKit: Open-Source Cheminformatics Software*. [Online]. Available: http://www.rdkit.org/

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[51] A. Lusci, G. Pollastri, and P. Baldi, "Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules," *J. Chem. Inf. Model.*, vol. 53, no. 7, pp. 1563–1575, Jul. 2013.

[52] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[53] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively multitask networks for drug discovery," Feb. 2015, *arXiv:1502.02072*. [Online]. Available: https://arxiv.org/abs/1502.02072

[54] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[55] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (With discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, Apr. 2000.

[56] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[57] S. J. Swamidass, C.-A. Azencott, T.-W. Lin, H. Gramajo, S.-C. Tsai, and P. Baldi, "Influence relevance voting: An accurate and interpretable virtual high throughput screening method," *J. Chem. Inf. Model.*, vol. 49, no. 4, pp. 756–766, Apr. 2009.

[58] B. Ramsundar, B. Liu, Z. Wu, A. Verras, M. Tudor, R. P. Sheridan, and V. Pande, "Is multitask deep learning practical for Pharma?" *J. Chem. Inf. Model.*, vol. 57, no. 8, pp. 2068–2076, Aug. 2017.

[59] C. Robert, "Machine learning, a probabilistic perspective," *Chance*, vol. 27, no. 2, pp. 62–63, Apr. 2014.

[60] E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar, and V. S. Pande, "PotentialNet for molecular property prediction," *ACS Central Sci.*, vol. 4, no. 11, pp. 1520–1530, Nov. 2018.

[61] M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, and D. R. Koes, "Protein-ligand scoring with convolutional neural networks," *J. Chem. Inf. Model.*, vol. 57, no. 4, pp. 942–957, Apr. 2017.

[62] J. D. Durrant and J. A. Mccammon, "NNScore 2.0: A neural-network receptor-ligand scoring function," *J. Chem. Inf. Model.*, vol. 51, no. 11, pp. 2897–2903, Nov. 2011.

[63] T. Nguyen, H. Le, and S. Venkatesh, "GraphDTA: Prediction of drug-target binding affinity using graph convolutional networks," 2019, *BioRxiv:684662*. [Online]. Available: https://www.biorxiv.org/content/10.1101/684662v1.abstract

[64] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and Leman go neural: Higher-order graph neural networks," in *Proc. AAAI*, Honolulu, HI, USA, vol. 33, Sep. 2019, pp. 4602–4609.

SHUANG WANG received the B.E. degree in computer science and technology from the Ocean University of China, Qingdao, China, in 2014, where she is currently pursuing the Ph.D. degree in computer application technology. Her research interests mainly focus on using deep learning method for computer-aided drug design, which include molecular property prediction, drug-target affinity prediction, and data driven drug design and optimization.

ZHEN LI received the Ph.D. degree in computer science from the Ocean University of China, Qingdao, China, in 2014. He is currently an Associate Professor with the Ocean University of China, Qingdao, China. He has authored over 20 international journal and conference papers. His current research interests include graph convolution model, machine learning and bioinformatics. He is focusing on using deep learning method for computer-aided drug design.

SHUGANG ZHANG received the B.S. and Ph.D. degrees from the Ocean University of China, Qingdao, China, in 2013 and 2019, respectively. He was a Visiting Ph.D. Student with The University of Manchester, U.K., from 2017 to 2018. He is currently a Postdoctoral Researcher with the Department of Computer Science and Technology, Ocean University of China. His research interests mainly focus on the bioinformatics and biophysiology, which include computational modeling of cardiac electrophysiology, cardiotoxicity prediction of common ambient air pollutants, and in-silico drug design using deep learning approaches.

MINGJIAN JIANG received the B.E. degree in computer science and technology from Qingdao University, Qingdao, China, in 2014. He is currently pursuing the Ph.D. degree in computer application technology with the Ocean University of China. His research interests are computer-aided drug design and bioinformatics, which include the protein binding site prediction, protein structure prediction, and drug-target affinity prediction. Recently, he has focused on the application of deep learning in drug design research.

XIAOFENG WANG received the B.S. degree from the Henan University of Technology, in 2016. He is currently pursuing the master's degree in computer application technology with the Ocean University of China, Qingdao, China. His research interests mainly focuses on computer-aided drug design using deep learning methods, which include molecular property prediction, compound-protein interaction prediction, and drug side effects.

ZHIQIANG WEI received the Ph.D. degree from Tsinghua University, China, in 2001. He is currently a Professor with the Ocean University of China. He is also the Director of the High Performance Computing Center, Pilot National Laboratory for Marine Science and Technology (Qingdao). His current research interests are in the fields of intelligent information processing, social media, and big data analytics.