Pattern Recognition 128 (2022) 108659

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Molecular substructure graph attention network for molecular property identification in drug discovery

Xian-bin Ye[a,1], Quanlong Guan[a,b,1], Weiqi Luo[a,b], Liangda Fang[a], Zhao-Rong Lai[c,*], Jun Wang[d]

[a] Department of Computer Science, College of Information Science and Technology, Jinan University, Guangzhou 510632, China
[b] Guangdong Institute of Smart Education, Jinan University, Guangzhou 510632, China
[c] Department of Mathematics, College of Information Science and Technology, Jinan University, Guangzhou 510632, China
[d] Ping An Healthcare Technology, Chaoyang, Beijing 100027, China

ARTICLE INFO

Article history:
Received 7 July 2021
Revised 27 January 2022
Accepted 16 March 2022
Available online 18 March 2022

Keywords:
Molecular substructure
Graph attention
Molecular property identification

ABSTRACT

Molecular machine learning based on graph neural network has a broad prospect in molecular property identification in drug discovery. Molecules contain many types of substructures that may affect their properties. However, conventional methods based on graph neural networks only consider the interaction information between nodes, which may lead to the oversmoothing problem in the multi-hop operations. These methods may not efficiently express the interacting information between molecular substructures. Hence, We develop a Molecular SubStructure Graph ATtention (MSSGAT) network to capture the interacting substructural information, which constructs a composite molecular representation with multi-substructural feature extraction and processes such features effectively with a nested convolution plus readout scheme. We evaluate the performance of our model on 13 benchmark data sets, in which 9 data sets are from the ChEMBL data base and 4 are the SIDER, BBBP, BACE, and HIV data sets. Extensive experimental results show that MSSGAT achieves the best results on most of the data sets compared with other state-of-the-art methods.

1. Introduction

Drug discovery is time-consuming, labor intensive, and expensive. It usually starts with experimental discoveries of molecules and targets (i.e., de novo drug design) and the validations with in vitro experiments on cell lines and animals before moving to clinical tests [1]. The entire process from the discovery to the regulatory approval of a new drug can take as long as 12 years and cost upwards of US 2.8 billion. Furthermore, each drug developing stage has a very low success rate of about 1/5000.

Drug discovery is equipped with statistical learning since the rise of computational chemistry. In order to increase the speed of drug screening and reduce costs, researchers in cheminformatics have been building quantitative structure activity relationships (QSAR) via machine learning methods [2,3]. In recent years, with increasing biochemistry data volumes and advanced computing machines (e.g., Graphics Processing Unit, GPU), a large number of deep learning methods are applied to drug discovery because of their powerful capability of feature extraction and flexibility of model structures compared with conventional machine learning methods [4,5]. Due to the particularity of compound structures and the limitations of early-era feature engineering (e.g., molecular fingerprints, descriptors, and Simplified Molecular-Input Line-Entry System strings, SMILES [6]), it is difficult for conventional neural networks to extract compound substructural information from raw molecules.

The emergence of graph convolutional networks (GCN) brings in a new breakthrough in drug-related tasks [7]. Niepert et al. [8] propose a general method to extract local information from graph data and apply it to the activity prediction of compound molecules. Structural representation of compound molecules is encoded as a molecular fingerprint, a high dimensional vector of binary digits. Duvenaud et al. [9] use a GCN to obtain molecular fingerprints and apply it to molecular property prediction. Kearnes et al. [10] develop a GCN called “weave module”, which can aggregate the atom and bond information as node features, and apply it to activity prediction. Zhang et al. [11] propose a graph neural network based on the graph structure GSCN, which balances between the impor-

[*] Corresponding author.

E-mail addresses: yexianbin@stu2019.jnu.edu.cn (X.-b. Ye), gql@jnu.edu.cn (Q. Guan), lwq@jnu.edu.cn (W. Luo), fangld@jnu.edu.cn (L. Fang), laizhr@jnu.edu.cn (Z.-R. Lai), junwang.deeplearning@gmail.com (J. Wang).

[1] These two authors contribute equally to this article.

https://doi.org/10.1016/j.patcog.2022.108659
0031-3203/© 2022 Elsevier Ltd. All rights reserved.

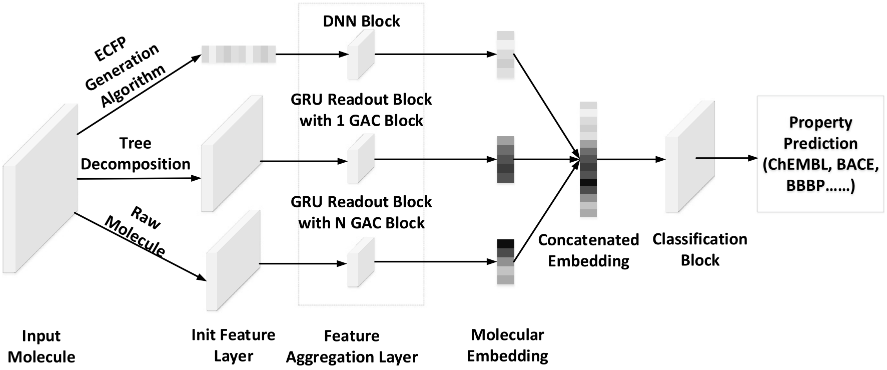
ECFP Generation Algorithm
Tree Decomposition
Raw Molecule
DNN Block
GRU Readout Block with 1 GAC Block
GRU Readout Block with N GAC Block
Concatenated Embedding
Classification Block
Property Prediction (ChEMBL, BACE, BBBP......)
Input Molecule
Init Feature Layer
Feature Aggregation Layer
Molecular Embedding

Fig. 1. Entire structure of MSSGAT.

tance of graph structural information and the node neighboring information. Since molecular property prediction is on the level of the entire compound structure, Herr et al. [12] propose the entire graph-level representation learning, which is shown to be effective by the experiments. Ding et al. [13] learn the graph-level representation by combining the depth-first-search algorithm with their node selection strategy on the features of local structures. Fang et al. [14] introduce a structured multi-head self-attention mechanism to obtain the graph-level representation of the fused graph structural information.

Although there are a large number of GNNs and GCNs handling molecular structures, these conventional methods only consider the interaction information between nodes, which may suffer from the oversmoothing problem of multi-hop operations. They seldom take molecular substructures into consideration, but the interacting information between substructures is crucial to molecular properties. Consequently, the molecular substructural information is not fully utilized, especially for biomacromolecules containing polycyclic structures. To fill this gap, we propose a Molecular SubStructure Graph ATtention (MSSGAT) network, whose entire structure is shown in Fig. 1. The main contributions can be summarized as follows: **1.** We propose to use a structural feature extraction scheme including 3 types of features (raw + tree decomposition + ECFP): raw molecular graphs, molecular structural features via tree decomposition [15], and Extended-Connectivity FingerPrints (ECFP) [16]. **2.** We design a framework including several graph attention convolutional (GAC) blocks and deep neural network (DNN) blocks to process the above structural features. We also improve the GAC blocks to relieve the gradient vanishing or exploding problem. **3.** We design a readout block based on gated recurrent units (GRU) [17]. The readout blocks collaborate with the GAC blocks in a nested architecture to obtain molecular embeddings. **4.** We visualize the molecules and mark the most important atoms with the attention scores produced by MSSGAT, which can be a good reference for subsequent researches by medicinal chemists. We evaluate the performance of MSSGAT on 13 benchmark data sets, in which 9 are from the ChEMBL data base [18] and 4 are the SIDER [3], BBBP [3], BACE [3], and HIV [3] data sets. Extensive experimental results show that MSSGAT achieves the best results on most of the data sets compared with other state-of-the-art methods.

2. Related works

Due to the establishment of drug data bases, methods based on deep learning have caught more attention in the pharmaceutical industry. First, DNN has been widely used in the quantitative structure activity relationship (QSAR). Ma et al. [19] use experiments to verify that QSAR models based on DNNs are better than some traditional machine learning models (the random forest and the support vector machine). You et al. [20] show that DNNs are effective in predicting drug-target pairs and can be used for drug repurposing. Li et al. [21] use a multi-task DNNs model to predict human cytochrome P450 inhibitors, and the results show that the multi-task model has a better predictive effect than several traditional machine learning models (SVM, KNN, the decision tree, and the logistic regression).

Accelerating the speed of virtual screening and accurately capturing compounds that interact with the targets have been hot spots in drug research in recent years. The emergence of Generative Adversarial Networks (GAN) [22] provides new ideas for speeding up the research of virtual drug screening. Kadurin et al. [23] adopt the anti-autoencoding (AAE) network structure, and use NCI-60 cell line assay data for 6252 compounds to train the network. The output of the network is used to search the pubchem data base and screen out candidates with anticancer activities. AAEs can be used to generate new molecular fingerprints that have specific molecular characteristics.

Recently, some GCNs have been applied to the property prediction for small molecules. They mainly consider the interacting information between nodes, which is indicated by the adjacency matrix of the molecular graph. However, traditional GCNs may neglect the fact that chemical bonds (edges) in different molecules can be similar if the interatomic distances are similar. To address this problem, Shang et al. [24] develop an Edge–Aware multi-view spectral GCN (EAGCN) approach to enhance the property prediction for small molecules.

Nevertheless, existing graph-based models may neglect the interacting information between molecular substructures, which also influences the molecular property based on the knowledge of chemistry. Zhang et al. [25] develop a fragment-oriented GAT (FraGAT) to boost the interaction between fragments of molecular graphs, which may retain functional groups. FraGAT also aggregates the atom-level features to represent the molecular graph. However, if most rings are partitioned into the same fragment, FraGAT may deteriorate in macromolecules like polycyclic molecules (i.e., molecules containing no less than 5 rings), because the topological information between rings is not fully utilized. Similarly, the RNN-based MSGG model [26] transforms a molecule into a substructure-based graph. Then this graph is expanded into three-channel sequences for the input of a Bi-GRU model. However, MSGG pays less attention to the interacting and topological information between atoms in the original molecular graph. ECFP represents many molecular substructures via sparse binary vectors but neglects the topological information compared with the graph-based method. Thus ECFP may not catch the interacting information of the atoms. To better exploit the useful fine-grained fragments of ECFP, we adopt it as one of the features in the proposed MSSGAT.

Table 1
Summary of 13 benchmark data sets.

Data set	Name	Data type	Number of compounds
CHEMBL203	Epidermal growth factor receptor erbB1	SMILES	1794
CHEMBL267	Tyrosine-protein kinase SRC	SMILES	1251
CHEMBL279	Vascular endothelial growth factor receptor 2	SMILES	3266
CHEMBL325	Histone deacetylase 1	SMILES	517
CHEMBL340	Cytochrome P450 3A4	SMILES	3542
CHEMBL333	Matrix metalloproteinase-2	SMILES	321
CHEMBL2971	Tyrosine-protein kinase JAK2	SMILES	1582
CHEMBL2842	Serine/threonine-protein kinase mTOR	SMILES	2455
CHEMBL4005	PI3-kinase p110-alpha subunit	SMILES	2232
HIV	Human immuno-deficiency virus	SMILES	$41,913$
BBBP	Blood-brain barrier penetration	SMILES	2053
BACE	Human β-secretase 1	SMILES	1522
SIDER	Side Effect Resource	SMILES	1427

3. Dataset preparation

3.1. Anti-cancer data sets from ChEMBL

The anti-cancer active molecules are collected from the ChEMBL data base [18], which includes some common variables like the IC50 value, the EC50 value, Inhibition, and the Ki value. The data base uses pChEMBL values to record the relative activity of the compounds, which allows for a number of measurements (i.e., half-maximal response concentration/epotency/affinity) to be compared in a negative logarithmic scale. According to Lenselink et al. [27], pChEMBL $= 6.5$ (approximately $300nM$) is chosen as the decision boundary. It indicates that a compound with pChEMBL $\geqslant 6.5$ is an inhibitor, otherwise it is a non-inhibitor. In addition, some compounds have multiple legal activity test records, so we average all the legal pChEMBL values for the same compound as a relatively reasonable result. To demonstrate the superiority of our model for biomacromolecules containing polycyclic structures, we retain molecules containing no less than 5 ring structures in the data set.

3.2. Other benchmark data sets

3.2.1. HIV

The HIV data set is introduced by the Drug Therapeutics Program (DTP) AIDS Antiviral Screen, which tests the abilities of 41,913 compounds to inhibit HIV replication. Original results are divided into three categories: inactive, active, and moderately active. Wu et al. [3] combine the latter two classes, making it a binary classification task and propose a scaffold splitting for this data set to discover new structures of HIV inhibitors.

3.2.2. BACE

The BACE data set contains the experimental values collected from the scientific literature over the past decade. It provides binding results (binary labels) for the set of inhibitors of BACE-1 [3].

3.2.3. BBBP

The Blood-brain barrier penetration (BBBP) data set is collected from the study of modeling and predicting the barrier permeability.

3.2.4. SIDER

The Side Effect Resource (SIDER) is a data set collected from marketed drugs with adverse drug reactions. This data set includes 12 binary-classification tasks.

All the above data sets are summarized in Table 1. We use a scaffold split [3] to divide a data set into three parts: a training set, a validation set and a test set (the ratio is $8:1:1$). The scaffold split attempts to discriminate between different molecular structures in the train/validation/test sets, which offers a greater challenge and demands a higher level of generalization ability for deep learning models than the random split. In addition, ROC-AUC is used for model evaluation. Anti-cancer data sets of 9 targets are mentioned with the following “ChEMBL” IDs.

4. MSSGAT

4.1. Structural feature extraction for anti-cancer inhibitors

Traditional machine learning methods usually use molecular descriptors (e.g., molecular weight and Alogp) as inputs, but pharmacologists usually analyze molecular structures instead of molecular descriptors. Besides, molecular descriptors may easily neglect the local structural information of molecules. Hence molecular descriptors may not provide sufficient classification information. On the other hand, a molecular fingerprint is high-dimensional and sparse. The valid substructure bits in the fingerprint vector are sparse, and it is difficult to obtain the effective correlation information between the substructures. In recent years, although many GNN models come out, their input features are just local information of molecular graphs.

In order to extract structural features for anti-cancer inhibitors, we propose a composite feature scheme “raw + tree decomposition + ECFP” as follows.

Raw molecular graph and its descriptors. The raw molecular graph is a basic structure of atomic relationships, where each node represents an atom. Each atom has 9 atomic features, which are summarized in Table 2. The number of charges and the number of free radicals are encoded as integers, while other features are encoded as one-hot vectors. Such raw features are acquired by the open-source chemical information calculation library RDkit [28]. The distributions of atom numbers and pChEMBL values of the CHEMBL340 data set are shown in Fig. 2.

Structural features via tree decomposition. In order to extract global structural features, we adopt the tree decomposition algorithm for molecular graphs and generate multiple effective substructures [15,29,30]. Such a tree-like structure could represent the substructural components and the connections between these components, then we could use the connection trees formed by these substructures to represent the molecules. Substructures are regarded as nodes and their connections are regarded as edges. All substructures corresponding to SMILES (namely, token) form the vocabulary, and the substructures mapping dictionaries are defined for each data set. The tree decomposition process is shown in Fig. 3. Word embeddings are initialized by summing up the atom embedding vectors in each substructure of the raw molecular graph from the “raw” branch. The substructural embeddings are represented by concatenating word embeddings and one-hot

Table 2
Atomic descriptors for raw molecular graph: initialization of atomic representations of molecules.

Atom feature	Feature size	Description
Atom	16	[B, C, N, O, F, Si, P, S, Cl, As, Se, Br, Te, I, At, metal]
Degree	11	Number of covalent bonds [0,1,2,3,4,5,6,7,8,9,10]
Formal charge	1	Electrical charge (integer)
Radical electrons	1	Number of radical electrons (integer)
Hybridization	6	[sp, sp2, sp3, sp3d, sp3d2, other]
Aromaticity	1	Aromatic system (0/1)
Hydrogens	5	Number of connected hydrogens [0,1,2,3,4]
Chirality	1	Chiral center (0/1)
Chirality type	2	R/S

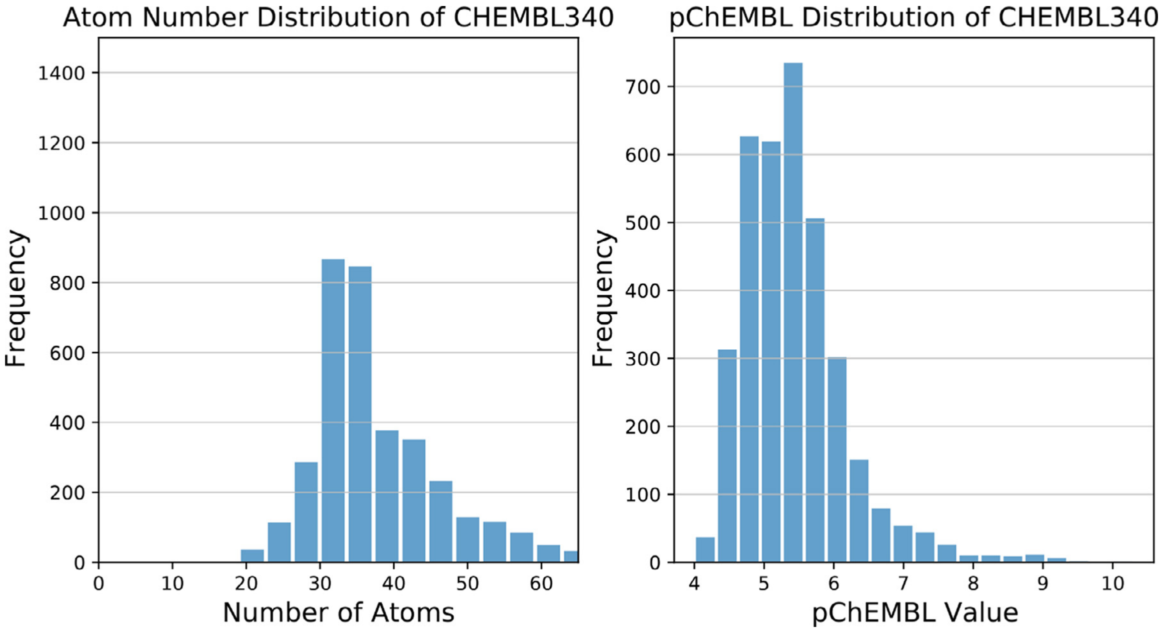

Fig. 2. Distributions of atom numbers and pChEMBL values of CHEMBL340 data set.

embeddings, and the entire connection tree is formed by a matrix of these substructural embeddings.

Extended-connectivity FingerPrints (ECFP). It is better to represent chemical molecules by structural descriptors (e.g., atom-pair [31] and topological torsion [32]) besides global descriptors (e.g., molecular weight, polar surface area, and logP). Molecular fingerprints provide structural molecular characteristics and improve stability and generalization of MSSGAT. We use the Extended-Connectivity FingerPrints (ECFP) [16] for MSSGAT, and design some particular network blocks to process these features (Section 4.2.3). ECFP splits the molecule into structural identifiers by the traversal substructures within a distance from each atom. Then the identifiers are hashed to a vector with a fixed size (See Fig. 4). The RDkit can be used to calculate ECFPs, and the effective diameter and the length of the representation vectors are set as 2 and 512 according to Rogers and Hahn [16], respectively.

4.2. Key modules for MSSGAT

Now we have 3 types of features: “raw + tree decomposition + ECFP”, denoted by $\{\mathbf{a}^{[l]}\}_{l=1}^{L}$, $\{\mathbf{b}^{[l]}\}_{l=1}^{L}$ and $\{\mathbf{c}^{[l]}\}_{l=1}^{L}$ (L is the number of samples in a batch), respectively. Then they will be processed by MSSGAT to make classifications. MSSGAT mainly consists of four modules: several GAC blocks for $\{\mathbf{a}^{[l]}\}_{l=1}^{L}$ and $\{\mathbf{b}^{[l]}\}_{l=1}^{L}$, a DNN block for $\{\mathbf{c}^{[l]}\}_{l=1}^{L}$, a readout block based on GRU, and a classifier based on a multilayer perceptron. After receiving and processing the above features, the GAC, DNN and readout blocks output graph embedding vectors, which are further concatenated as the final embedding vector. This final vector is fed into the classifier to get the classification result. The whole framework of the entire MSSGAT is shown in Fig. 1.

4.2.1. Graph attention convolutional block

The existing GCN [33] assigns the same weight to all the neighboring nodes of the central node, which is not suitable for representing molecular structures, because the contributions of different atoms or clusters to the central atom are different. For example, the benzene ring has a different effect from the hydroxyl group on the atom C of the carboxyl group in the benzoic acid. Inspired by [34], we propose a kind of GAC block to address such different effects of different molecular parts. It consists of 3 steps:

- Calculate the attention coefficient $\alpha_{ij}^{[l]}$.
- Compute the weighted feature summation $\mathbf{h}_{i,(K)}^{[l]\prime}$.
- Implement several post-processing operations to obtain the updated hidden states.

Given the initial input of the lth sample $\mathbf{z}^{(0),[l]}$ containing vertices $\{\ldots,\mathbf{h}_i^{[l]},\ldots,\mathbf{h}_j^{[l]},\ldots\}$, the attention coefficient is calculated with a concatenation operator and a single-layer feedforward map:

$$e_{ij}^{[l]} = f\left(\left[\mathbf{W}\mathbf{h}_i^{[l]} \| \mathbf{W}\mathbf{h}_j^{[l]}\right]\right), \quad j \in \mathcal{N}_i^{[l]}, \tag{1}$$

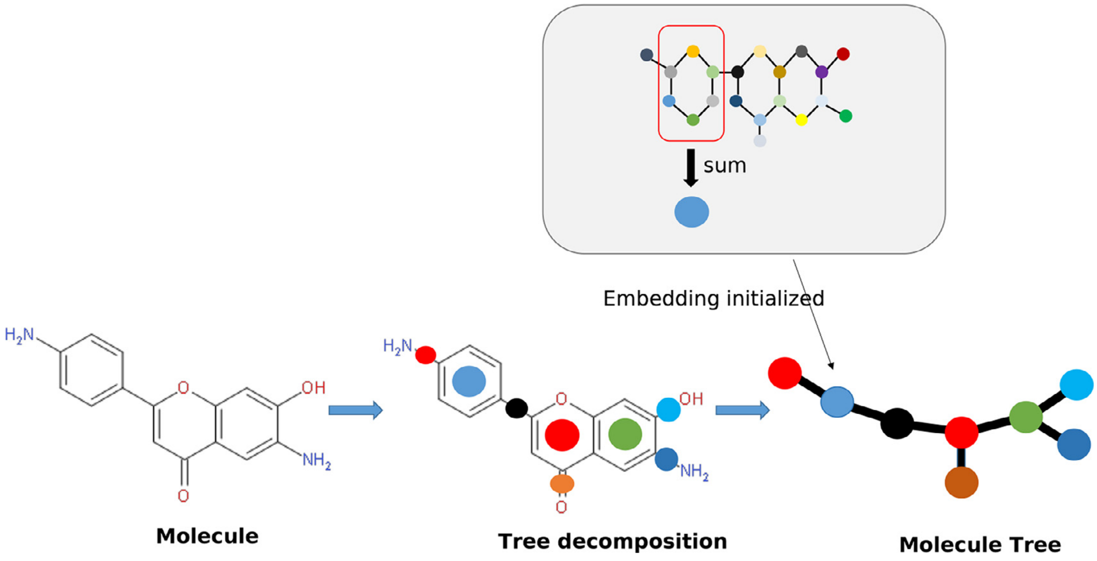

Fig. 3. Tree decomposition process for molecules. The upper black box indicates the initialization of substructural embeddings.

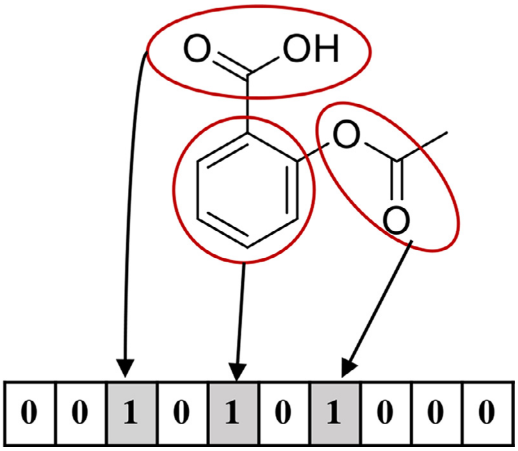

Fig. 4. Taking the aspirin as an example, if the pre-defined substructures exist, the corresponding positions of the ECFP vector are set as 1.

$$\alpha_{ij}^{[l]} = \frac{\exp\left(\text{LeakyReLU}\left(e_{ij}^{[l]}\right)\right)}{\sum_{m\in\mathcal{N}_i^{[l]}} \exp\left(\text{LeakyReLU}\left(e_{im}^{[l]}\right)\right)}, \tag{2}$$

where $\mathcal{N}_i^{[l]}$ is the neighboring node set of vertex $\mathbf{h}_i^{[l]}$, $\mathbf{W}$ is a shared parameter of the linear map f that adjusts the features of the vertices $\{\ldots,\mathbf{h}_i^{[l]},\ldots,\mathbf{h}_j^{[l]},\ldots\}$, and $\|$ is the concatenation operator that concatenates the dominated terms. Eq. (1) calculates a kind of correlation between vertices $\mathbf{h}_i^{[l]}$ and $\mathbf{h}_j^{[l]}$, and (2) normalizes this correlation with the softmax function. Next, the attention coefficient $\alpha_{ij}^{[l]}$ is used to adjust the importance of the neighboring node. Moreover, we use a multi-head attention mechanism that includes K convolution kernels to calculate K new features $\mathbf{h}_{i,(k)}^{[l]\prime}$

$$\mathbf{h}_{i,(k)}^{[l]\prime} = \sum_{j\in\mathcal{N}_i^{[l]}} \alpha_{ij,(k)}^{[l]} \mathbf{W}_{(k)} \mathbf{h}_j^{[l]}. \tag{3}$$

The eigenvalues produced by the convolution operation easily deviate from the normal distribution, thus they have an adverse effect on the convergence of the network (gradient disappearance or gradient explosion) and worsen the model performance. Therefore, we add a ReLU layer and a Batch Normalization (BN) layer after each convolution kernel in each GAC block

$$\mathbf{z}_{i,(k)}^{[l]} = ReLU(\mathbf{h}_{i,(k)}^{[l]\prime}), \tag{4}$$

$$\mathbf{z}_{i,(k)}^{[l]\prime} = BN(\mathbf{z}_{i,(k)}^{[l]}, \{\mathbf{z}_{i,(k)}^{[l]}\}_{l=1}^{L}). \tag{5}$$

Then we concatenate these new features and implement a full connection $FC0$ to obtain the updated hidden state $\mathbf{z}_{i,(K)}^{(1),[l]}$:

$$\mathbf{z}_{i,(K)}^{(1),[l]} = FC0(\|_{k=1}^{K}\mathbf{z}_{i,(k)}^{[l]\prime}). \tag{6}$$

The architectures of the Graph Attention kernels (named as GA below) and the GAC blocks are shown in Fig. 5. Furthermore, multiple GAC blocks can be stacked to constitute a deeper network that can process molecular parts with more nodes:

$$\mathbf{z}^{(n+1),[l]} = GAC(\mathbf{z}^{(n),[l]}), \quad n = 0, 1, \ldots, N, \tag{7}$$

where the subscripts i and (K) can be omitted since they do not disturb the operator GAC. Since the raw molecular graph $\mathbf{a}^{[l]}$ usually has much more nodes than its structural features via tree decomposition $\mathbf{b}^{[l]}$, we use N GAC blocks for $\mathbf{a}^{[l]}$ while only 1 GAC block for $\mathbf{b}^{[l]}$. Then the number of stacked blocks is consistent with the number of nodes, which is beneficial to the convolution performance. The deployments are combined with the readout block in the next subsubsection.

4.2.2. Readout block based on gated recurrent units

Readout operation is similar to the global pooling of CNN, which performs an aggregation operation on the features of all nodes to output a global representation of the graph. Inspired by GRU [17] (a variant of an LSTM [35] recurrent network unit), we design a readout block that can synthesize molecular embeddings according to the order of the GAC blocks. Suppose the hidden state of the ith node after the nth GAC block for the lth sample is $\mathbf{z}_{i,(K)}^{(n),[l]}$, then the graph embedding is $\mathbf{g}^{(n),[l]}$:

$$\mathbf{g}^{(n),[l]} = Mean\left(\mathbf{z}_{i,(K)}^{(n),[l]} \mid \forall v_i^{[l]} \in V^{[l]}\right), \tag{8}$$

where $V^{[l]}$ is the vertex set of the lth sample. Denote $\mathbf{G}^{(n+1),[l]}$ as the molecular embedding after the nth GAC block and $GRU^{(n)}$ as the update function at iteration n, then

$$\mathbf{G}^{(0),[l]} = Mean\left(f^{Lin}(\mathbf{z}_{i,(K)}^{(0),[l]}) \mid \forall v_i^{[l]} \in V^{[l]}\right), \tag{9}$$

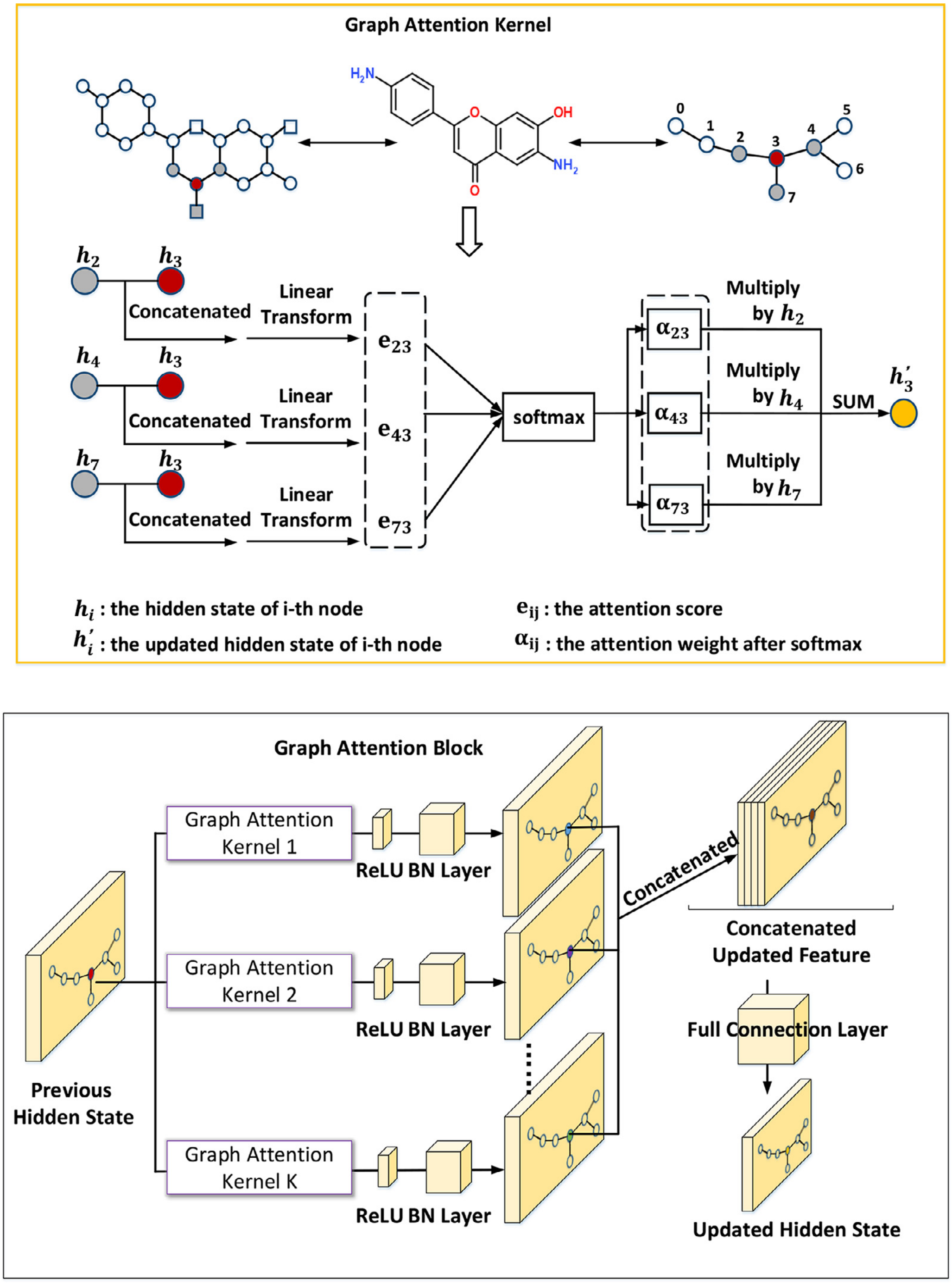

Fig. 5. Architectures of Graph Attention Kernel and Graph Attention Convolutional Block.

$$\mathbf{G}^{(n+1),[l]} = GRU^{(n)}\left(\mathbf{g}^{(n),[l]}, \mathbf{G}^{(n),[l]}\right), n = 0, 1, \ldots, N, \tag{10}$$

where f^{Lin} is a linear transform for initialization.

The architecture of the readout block is shown in Fig. 6. As for the raw molecular graph $\mathbf{a}^{[l]}$ and its structural features via tree decomposition $\mathbf{b}^{[l]}$, the readout deployments are:

$$\mathbf{z}^{(0),[l]} \leftarrow \mathbf{a}^{[l]}, \quad \mathbf{a}^{[l]\prime} \leftarrow \mathbf{G}^{(N+1),[l]}; \tag{11}$$

$$\mathbf{z}^{(0),[l]} \leftarrow \mathbf{b}^{[l]}, \quad \mathbf{b}^{[l]\prime} \leftarrow \mathbf{G}^{(2),[l]}. \tag{12}$$

Note that the notations $\mathbf{G}^{(N+1),[l]}$ and $\mathbf{G}^{(2),[l]}$ here go through different readout progresses because their inputs are different ($\mathbf{a}^{[l]}$ and $\mathbf{b}^{[l]}$).

If we zoom in the GRU operator (10) and omit the superscript $[l]$ without confusion, $\mathbf{g}^{(n)}$ and $\mathbf{G}^{(n)}$ first go through the update gate $\mathbf{u}^{(n)}$ and the reset gate $\mathbf{r}^{(n)}$:

$$\mathbf{u}^{(n)} = \sigma\left(\mathbf{W}_{\mathbf{u}^{(n)}}\mathbf{g}^{(n)} + \mathbf{X}_{\mathbf{u}^{(n)}}\mathbf{G}^{(n)}\right), \tag{13}$$

$$\mathbf{r}^{(n)} = \sigma\left(\mathbf{W}_{\mathbf{r}^{(n)}}\mathbf{g}^{(n)} + \mathbf{X}_{\mathbf{r}^{(n)}}\mathbf{G}^{(n)}\right), \tag{14}$$

where $\mathbf{W}_{\mathbf{u}^{(n)}}$, $\mathbf{X}_{\mathbf{u}^{(n)}}$, $\mathbf{W}_{\mathbf{r}^{(n)}}$ and $\mathbf{X}_{\mathbf{r}^{(n)}}$ are linear transforms to be trained for $\mathbf{u}^{(n)}$ and $\mathbf{r}^{(n)}$, respectively. Then the hidden state $\tilde{\mathbf{G}}^{(n)}$

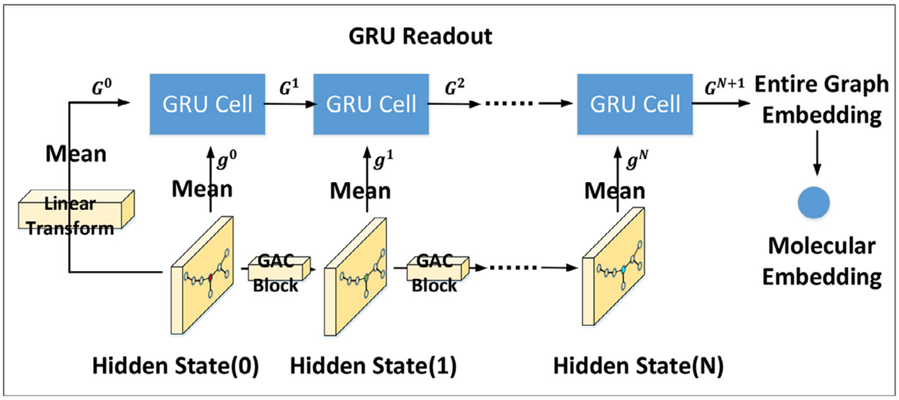

Fig. 6. Readout block based on GRU for MSSGAT.

is computed by:

$$\tilde{\mathbf{G}}^{(n)} = \tanh\left(\mathbf{W}_{\mathbf{G}^{(n)}}\mathbf{g}^{(n)} + \mathbf{X}_{\mathbf{G}^{(n)}}(\mathbf{r}^{(n)} \odot \mathbf{G}^{(n)})\right), \qquad (15)$$

where $\mathbf{W}_{\mathbf{G}^{(n)}}$ and $\mathbf{X}_{\mathbf{G}^{(n)}}$ are linear transforms to be trained, and $\odot$ is the element-wise multiplication. When $\mathbf{r}^{(n)}$ is close to $\mathbf{0}$, the current state $\mathbf{G}^{(n)}$ would be highly forgotten. The 1st recurrent state is an element-wise linear interpolation

$$\mathbf{G}^{(n,1)} = (\mathbf{1} - \mathbf{u}^{(n)}) \odot \mathbf{G}^{(n)} + \mathbf{u}^{(n)} \odot \tilde{\mathbf{G}}^{(n)}, \qquad (16)$$

where the update gate $\mathbf{u}^{(n)}$ controls the update strength from the current embedding $\mathbf{G}^{(n)}$ to the hidden state $\tilde{\mathbf{G}}^{(n)}$.

We let $\mathbf{g}^{(n)}$ and $\mathbf{G}^{(n,1)}$ go through the recurrence (13)–(16) and obtain $\mathbf{G}^{(n,2)}$, then $\mathbf{g}^{(n)}$ and $\mathbf{G}^{(n,2)}$ go through the recurrence... The next molecular embedding is $\mathbf{G}^{(n+1)} \triangleq \mathbf{G}^{(n,M)}$, where M is the number of recurrences. The weights of these two module are updated within one backward pass. We will conduct experiments to compare the readout modules of LSTM and Concat + FC (concatenating the 3 types of features and using a fully connected layer) to prove the superiority of our GRU readout module in Section 5.4.

4.2.3. Deep neural network block for ECFPs

In order to fully consider the structural molecular characteristics and improve stability and generalization of MSSGAT, molecular fingerprints are necessary as input features, because it is more suitable to represent chemical molecules by structural descriptors (e.g., atom-pair [31] and topological torsion [32]) rather than global descriptors (e.g., molecular weight, polar surface area, and logP). The ECFPs are good choices, but they are high-dimensional and sparse vectors where each bit is binary, which will cause the dimensionality disaster. Therefore, we introduce a pyramid-form DNN block where the number of neurons is gradually reduced by one-half per layer from the input layer to the output layer, to further extract lower-dimensional features from the ECFPs.

$$\mathbf{c}^{[l]\prime} = DNN(\mathbf{c}^{[l]}). \qquad (17)$$

4.2.4. Classification block

We propose a three-layer feedforward classification block for MSSGAT. The above GAC and DNN blocks produce concatenated embedding features $\mathbf{d}_l \triangleq [\mathbf{a}^{[l]\prime\top}; \mathbf{b}_l^{[l]\prime\top}; \mathbf{c}_l^{[l]\prime\top}]^\top$, which are then fed into two parallel modules: the fully connected layer 1 (FC1) and the wide fully connected layer (FCwide). FCwide provides a feature extraction channel by one fully connected layer directly connecting with high-level features.

$$\mathbf{d}_l^{BN} = BN(\mathbf{d}_l, \{\mathbf{d}_l\}_{l=1}^{L}), \qquad (18)$$

$$\mathbf{d}_l^{FC1} = FC1(\mathbf{d}_l^{BN}), \qquad (19)$$

$$\mathbf{d}_l^{FCwide} = FCwide(\mathbf{d}_l^{BN}). \qquad (20)$$

The FC1 features would further go through the fully connected layer 2 (FC2) before being concatenated with the FCwide features.

$$\mathbf{d}_l^{FC1,BN} = BN(\mathbf{d}_l^{FC1}, \{\mathbf{d}_l^{FC1}\}_{l=1}^{L}), \qquad (21)$$

$$\mathbf{d}_l^{FC2} = FC2(\mathbf{d}_l^{FC1,BN}), \qquad (22)$$

$$\mathbf{d}_l^{fin} = [\mathbf{d}_l^{FCwide\top}; \mathbf{d}_l^{FC2\top}]^\top. \qquad (23)$$

With more FC layers, MSSGAT could gather low-level features to form higher-level features for classification. The reasons to concatenate FC2 and FCwide features are:

- High-level features could capture global information. Simultaneously using low-level and high-level features could improve the generalization ability of MSSGAT.
- The backpropagation of the error terms could get smoother and the gradient vanishing problem could be relieved to some extent.

We use the ReLU activation and the dropout technique with dropout rate 0.1 in FC1, FC2 and FCwide layers, and use the softmax function for the output layer:

$$p_l = \frac{1}{1 + e^{-\theta^\top \mathbf{d}_l^{fin}}}, \qquad (24)$$

where θ is the regression coefficient vector that would be trained in the network training, and p_l could be seen as the lth sample probability being an anti-cancer inhibitor. Last, MSSGAT can be trained by maximizing the log-likelihood of the observations:

$$l(\mathcal{Y}, \mathcal{P}) = \sum_{l=1}^{L}(y_l \log(p_l) + (1 - y_l)\log(1 - p_l)), \qquad (25)$$

where $\mathcal{Y} = \{y_l\}_{l=1}^{L}$ and $\mathcal{P} = \{p_l\}_{l=1}^{L}$ are the true probabilities and the estimated probabilities by MSSGAT for a batch of observations being inhibitors, respectively. The diagram of the classification block is shown in Fig. 7.

4.3. Model summary for MSSGAT

The model structure and training details for MSSGAT are summarized as follows:

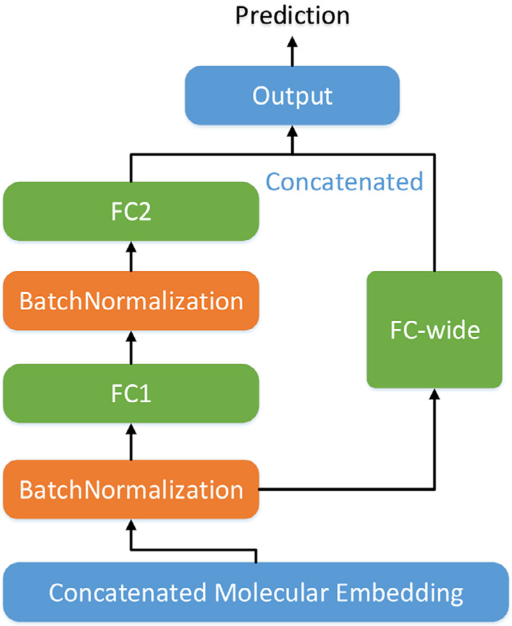

Fig. 7. Diagram of classification block for MSSGAT.

- For the raw molecular features $\boldsymbol{\alpha}^{[l]}$, we use $N = 3$ GAC blocks and $K = 4$ graph attention kernels for each GAC block. For the structural features via tree decomposition $\boldsymbol{b}^{[l]}$, we set $N = 2$ and $k = 4$. The sizes of $\boldsymbol{\alpha}^{[l]}$ and $\boldsymbol{b}^{[l]}$ are 44 and 128, respectively. We set 4 as a moderate number of heads for the multi-head attention mechanism.
- The DNN processing ECFPs consists of 1 input layer with 512 neurons, 1 output layer with 128 neurons, and 2 hidden layers with 256 and 128 neurons, respectively.
- The number of recurrences for the GRU operator is $M = 2$. The feature sizes of the current graph embedding $\mathbf{g}^{(n)}$ and the current state $\mathbf{G}^{(n)}$ are both 128.
- The classification block consists of 1 input layer with 384 neurons, 1 **FC1** layer with 64 neurons, 1 **FC2** layer with 192 neurons, 1 **FCwide** layer with 192 neurons, 1 output layer with 64 neurons, and 1 prediction layer with 2 neurons (to compute the probabilities of being an inhibitor and a non-inhibitor, respectively).
- The dropout rate for the fully connected layers in the DNN and the classification blocks is set as 0.1.
- The batch size L is set as 256.
- The initial global learning rate is set as $\eta_0 = 0.001$. In addition, to reduce the oscillation of the loss function in the later stage of training and make the network converge better, we use an exponential decay scheme of learning rate. Hence the learning rate for the tth epoch is $\eta_t = \eta_0 \gamma^t$, where γ is set as 0.9.
- The maximum number of epochs for training is set as 300, but we use an early stopping scheme to avoid overfitting and save training computation. If the performance of MSSGAT on the validation set does not improve during a certain number of epochs (called the “patience”), the training will be terminated in advance and the resulted model will be saved. We set the patience as 10 for MSSGAT in our experiments.

5. Experimental results

We conduct extensive experiments to verify the performance of MSSGAT, including ablation studies that analyze the effectiveness of each module in MSSGAT. Each data set is scaffold-split into three sets (training, validation, and test). We use ROC-AUC scores to evaluate MSSGAT and other competitors, including EAGCN [24], FraGAT [25], MSGG [26], AttentiveFP [36], weave [10], MPNN [37], NF [9], GCN, and GAT. We use three different random seeds in the experiments and average the final results. The hardware platform for this work is a Ubuntu 16.04 workstation equipped with an Intel Core i9-9820X CPU @ 3.30 GHz × 20, a 64 GB RAM, and an NVIDIA Geforce RTX 2080 Ti card. The entire MSSGAT is implemented with the PyTorch[2] and the Deep Graph Library[3] frameworks. The model parameters are initialized by the Xavier scheme [38]. The Adam optimizer [39] is used for optimization.

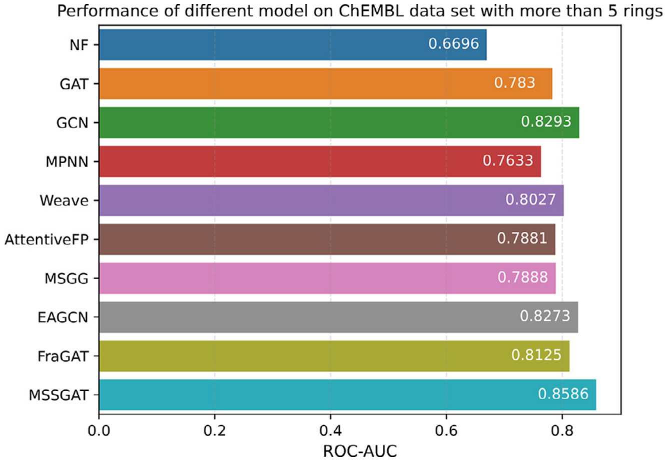

Fig. 8. Average ROC-AUCs on 9 ChEMBL data sets for different models.

5.1. Comparison results on ChEMBL data sets from ChEMBL

Experimental results on 9 ChEMBL benchmarks are presented in Table 3 and the average results are shown in Fig. 8. In brief, MSSGAT achieves the best average result on all the anti-cancer molecule data sets. It significantly outperforms all the competitors with an average ROC-AUC score of 0.8586. Hence the improved features of “raw + tree decomposition + ECFP” provide sufficient and useful structural information ranging from single atom features to substructural features, and finally to cluster features. Besides, the GAC, DNN and readout blocks effectively process and integrate the improved structural features to achieve better performance. Thus MSSGAT as a multi-level substructural feature extraction method can significantly improve the classification performance of biological macromolecules containing polycyclic structures.

To examine the extendability of MSSGAT, we also train MSSGAT by the HIV data set and test it on the ZINC data base.[4] Fig. 9 shows the histogram of ring numbers of the molecules on HIV. It indicates that about 15% are polycyclic molecules (i.e. molecules containing no less than 5 rings) and 85% are oligocyclic molecules (i.e. molecules containing less than 5 rings). As for the test set, we randomly sample 500 polycyclic molecules for each of the 10 most common scaffolds from ZINC, resulting in 5000 polycyclic molecules labelled with 10 different scaffolds. Then we use the Uniform Manifold Approximation and Projection (UMAP) [40] to visualize the embeddings of these 5000 polycyclic molecules from the last embedding layer of MSSGAT, shown in Fig. 10(a). Although there are only a small proportion of training samples are polycyclic molecules, the embeddings of the polycyclic molecules in the test

[2] https://pytorch.org/.

[3] https://www.dgl.ai/.

[4] http://zinc.docking.org.

Table 3
Prediction results on 9 ChEMBL data sets for various models. All the models have been tested for 3 times on each test set and the average results are presented. The best result on each data set is bold and the second best result is underlined.

ModelROC-AUCData Set	267	203	340	279	2842	325	333	4005	2971
NF	0.7403	0.6737	0.5884	0.7110	0.7899	0.6697	0.5827	0.6618	0.6091
GAT	<u>0.8005</u>	**0.8811**	0.8283	0.6686	0.8232	0.6234	0.9154	0.8629	0.6434
GCN	0.7667	0.8238	0.8232	**0.8418**	0.8214	0.8252	0.8566	**0.9009**	0.804
MPNN	0.6728	0.7471	0.7923	0.8043	0.8055	0.6411	0.8194	0.8157	0.7716
Weave	0.7939	0.8116	**0.9269**	0.7136	0.7612	0.7350	0.8802	<u>0.8750</u>	0.7273
AttentiveFP	0.7252	0.7806	0.7949	0.7561	0.7901	0.6986	**0.9412**	0.8707	0.7351
EAGCN	0.7576	0.8285	0.8607	0.8021	0.8474	<u>0.8443</u>	0.8297	0.8480	<u>0.8277</u>
MSGG	0.7230	0.7904	0.8353	0.7746	0.7784	0.7186	0.9113	0.8169	0.7510
FraGAT	0.7310	0.8236	0.8164	0.7735	<u>0.8661</u>	0.7808	<u>0.9167</u>	0.8074	0.7970
MSSGAT	**0.8125**	<u>0.8345</u>	<u>0.8948</u>	<u>0.8162</u>	**0.8687**	**0.9080**	0.8915	0.8418	**0.8592**

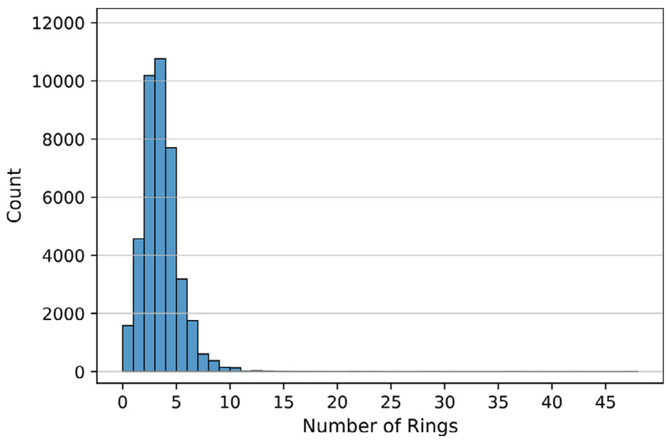

Fig. 9. Histogram of ring numbers of molecules on the HIV data set. There are 6333 polycyclic (containing no less than 5 rings) and 34794 oligocyclic (containing less than 5 rings) molecules, respectively.

set are well discriminated with a high Silhouette index [41]. On the other hand, we also sample 5000 oligocyclic molecules from ZINC and visualize their embeddings in the same way as the polycyclic molecules, shown in Fig. 10(b). MSSGAT is relatively less effective in oligocyclic molecules with a smaller Silhouette index, since they may not take good advantage of the multi-level molecular substructures of MSSGAT.

5.2. Comparison results on 4 benchmark data sets from MoleculeNet

We use 4 more benchmark data sets (with scaffold split) from MoleculeNet [3] to verify the generalization ability of MSSGAT, shown in Table 4. MSSGAT achieves the best results on 3 data sets and ranks the third on BBBP. It indicates that MSSGAT can effectively process molecular substructural features and has a good generalization ability, since the scaffold split is challenging to the generalization ability of a model.

5.3. Training process for MSSGAT

To analyze the training process of MSSGAT, we show it on the training and validation sets of BACE and BBBP in Fig. 11. The loss curves of MSSGAT on the training and validation sets tend to be smooth after training for about 50 epochs and 15 epochs, respectively. The ROC-AUC curves of MSSGAT on both BACE and BBBP are close to 1.0, thus MSSGAT can be efficiently trained.

To further validate the prediction performance of MSSGAT, we use the UMAP to visualize the latent spaces of MSSGAT and MPNN on BACE, shown in Fig. 12. We can see that MSSGAT learns discriminative embeddings for inhibitor identification, while MPNN could not separate the two classes well.

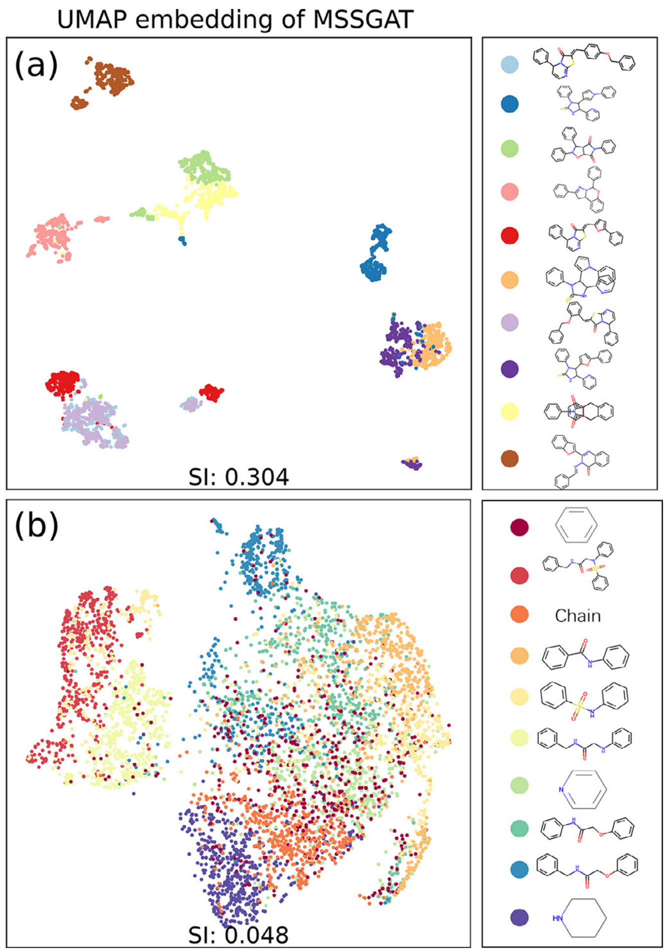

Fig. 10. Visualization of the molecular embeddings of MSSGAT on the ZINC data base. MSSGAT is trained by the HIV data set beforehand. A higher Silhouette index indicates a better discrimination. (a) and (b) represent the embeddings of polycyclic molecules and oligocyclic molecules, respectively.

5.4. Ablation experiments for MSSGAT

In order to analyze the contribution of each module to the whole MSSGAT, we conduct ablation studies on the HIV and the 9 ChEMBL data sets. The HIV data set contains about as

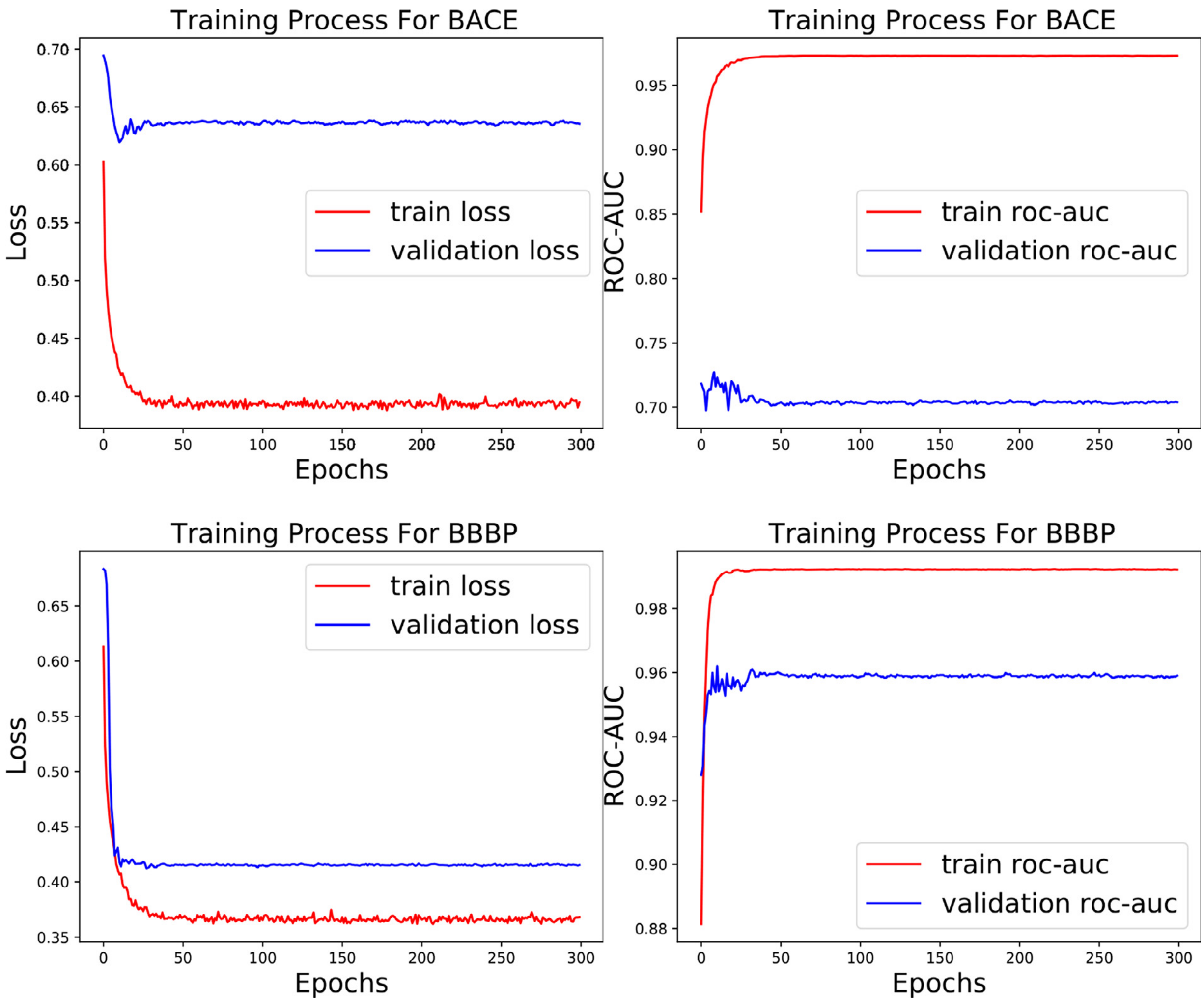

Fig. 11. Training Processes of MSSGAT on BACE and BBBP.

Table 4
Prediction results on 4 benchmark data sets (with scaffold split) for various models. All the models have been tested for 3 times on each test set and the average results are presented. The best result on each data set is bold and the second best result is underlined. **OOM**: Out of Memory.

ModelROC-AUCData Set	BACE	SIDER	BBBP	HIV
NF	0.6099	0.5173	0.6333	0.6971
GAT	0.6704	0.5435	0.6583	0.7733
GCN	0.6132	0.5713	0.6836	0.7770
MPNN	0.6870	0.5235	0.6723	0.7181
Weave	0.6440	0.5351	0.6596	0.7457
AttentiveFP	0.6587	0.5619	0.659971	0.7503
MSGG	0.8740	0.5278	0.7530	OOM
EAGCN	0.8337	0.6063	**0.8399**	0.7497
FraGAT	0.7896	0.5788	0.6913	0.7341
MSSGAT	**0.8805**	**0.6170**	0.7264	**0.7870**

Table 5
Ablation experiments on input features and the readout module for MSSGAT on the HIV data set. ROC-AUC scores are used in evaluation.

Model	Validation	Test
MSSGAT	**0.8209 ± 0.025**	**0.7828 ± 0.020**
Tree-only	0.8034 ± 0.003	0.7540 ± 0.004
Raw-only	0.8038 ± 0.007	0.7663 ± 0.010
ECFP-only	0.7598 ± 0.001	0.7184 ± 0.002
MSSGAT(GRU)	**0.8209 ± 0.025**	**0.7828 ± 0.020**
MSSGAT(LSTM)	0.8197 ± 0.011	0.7547 ± 0.021
MSSGAT(Concat + FC)	0.7915 ± 0.017	0.7451 ± 0.021

many as 40 thousand samples, thus it is reliable for ablation experiments. The results in Table 5 show that MSSGAT with the whole “raw + tree decomposition + ECFP” features outperforms the single ECFP module, the single tree decomposition module and the single raw molecular graph module on the validation and the test sets. Next, we retain our “raw + tree decomposition + ECFP” features but try different readout modules (GRU, LSTM, and Concat + FC). The results in Table 5 indicate that our GRU readout module outperforms other readout modules to some extent.

To further examine whether tree decomposition is effective in extracting substructural features from molecules or other extraction methods could be better, we adopt a common fragmentation algorithm *rdkit.Chem.Fragmentonbonds* [28] in RDkit for ablation experiments. Similar to the fragmentation in FraGAT [25], we retain all ring structures and break all acyclic single bonds to obtain the corresponding fragments by the function *rdkit.Chem.Fragmentonbonds*, and substitute these fragments for the

Table 6
Ablation experiments on feature extraction methods for MSSGAT on 9 ChEMBL data sets. ROC-AUC scores are used in evaluation. The best result on each data set is bold and the second best result is underlined. “AVG” indicates the average ROC-AUC score of a model on 9 ChEMBL data sets. MSSGAT^a: MSSGAT with tree decomposition features. MSSGAT^b: MSSGAT with common fragmentation features.

ModelROC-AUCData Set	267	203	340	279	2842	325	333	4005	2971	AVG
MSSGAT^a (Ours)	0.8125	**0.8345**	**0.8948**	0.8162	**0.8687**	**0.9080**	0.8915	0.8418	**0.8592**	**0.8586**
MSSGAT^b	0.8079	0.8169	0.8682	**0.8266**	0.8518	0.8387	0.9192	**0.8523**	0.8153	0.8441
Tree-only	0.8054	0.7766	0.8865	0.7723	0.8342	0.7845	0.9135	0.833	0.7540	0.8178
Raw-only	**0.8135**	0.8138	0.8698	0.7567	0.8041	0.7448	**0.9549**	0.7695	0.7877	0.8128
ECFP-only	0.7543	0.7955	0.8295	0.8157	0.8477	0.7759	0.855	0.8451	0.7911	0.8122

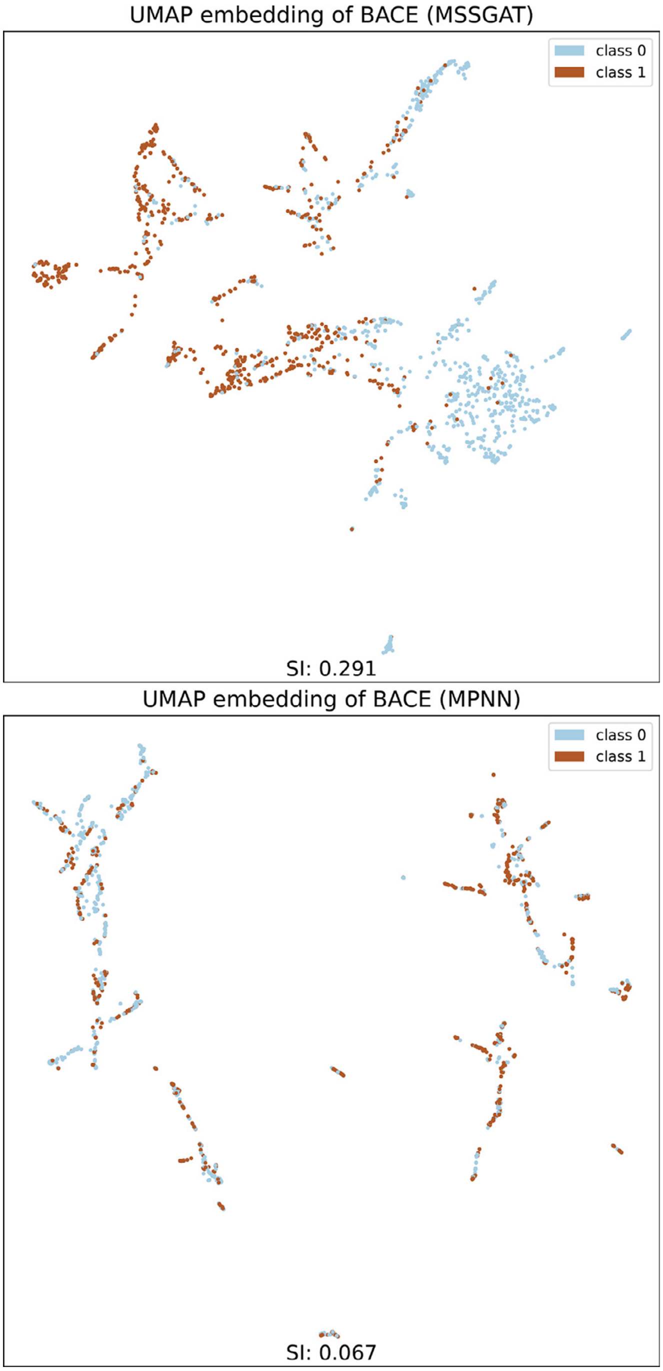

Fig. 12. Latent space visualizations via UMAP for MSSGAT (Upper) and MPNN (Lower) on BACE. A higher Silhouette index indicates a better discrimination.

tree decomposition features in MSSGAT. We denote our default MSSGAT with tree decomposition features and the altered MSSGAT with the common fragmentation features as MSSGAT^a and MSSGAT^b, respectively. The results on the 9 ChEMBL data sets in Table 6 show that MSSGAT^a outperforms MSSGAT^b in most cases. We also visualize the fragment features of MSSGAT^a and MSSGAT^b in Fig. 13, which indicate that MSSGAT^a provides more fragments and finer segmentations than MSSGAT^b. This may be the reason why MSSGAT^a is better than MSSGAT^b. To summarize this subsection, MSSGAT is effective in expoiting multi-level molecular substructures from the proposed “raw + tree decomposition + ECFP” features according to the above ablation experiments.

5.5. Important structure visualization

To further explore what information MSSGAT can provide on molecular structures, we visualize some molecules on the BACE data set and label the most important structures according to the attention scores (weights) in the prediction step of MSSGAT. Specifically, We extract the attention scores from the last GAC block of the “tree decomposition” branch of the well-trained MSSGAT. Then we sort the attention scores of all the tree nodes, and visualized the largest one (colored orange in Fig. 14). It indicates that MSSGAT allocates major attention to some common structures (e.g., carbon-oxygen double bonds, fluorine atoms and structures with ammonia), which may be an interesting reference for drug designers from a different perspective of machine learning.

6. Conclusion

In this work, we develop a novel Molecular SubStructure Graph ATtention (MSSGAT) network to capture substructural interacting information with structural feature extraction including raw molecular graphs, tree decomposition features, and Extended-Connectivity FingerPrints (ECFP). MSSGAT consists of several GAC, DNN and readout blocks that could effectively process molecular structural features and exploit the relationships between different molecular cliques of tree decomposition. Furthermore, MSSGAT uses both low-level and high-level features in classification to improve generalization ability, and adopts the dropout technique to relieve the gradient vanishing problem. Experimental results show that MSSGAT outperforms other state-of-the-art competitors in most cases. MSSGAT could also reveal important molecular structures by examining the attention scores, which gives a reference for drug designers from the perspective of machine learning. In summary, MSSGAT is an effective tool for molecular property identification and worth further investigations. Since MSSGAT is designed mainly for large and polycyclic molecules, it is relatively less effective in oligocyclic molecules. Future works could be designing different models for molecules with different sizes or finding more general molecular features for molecules with different substructures.

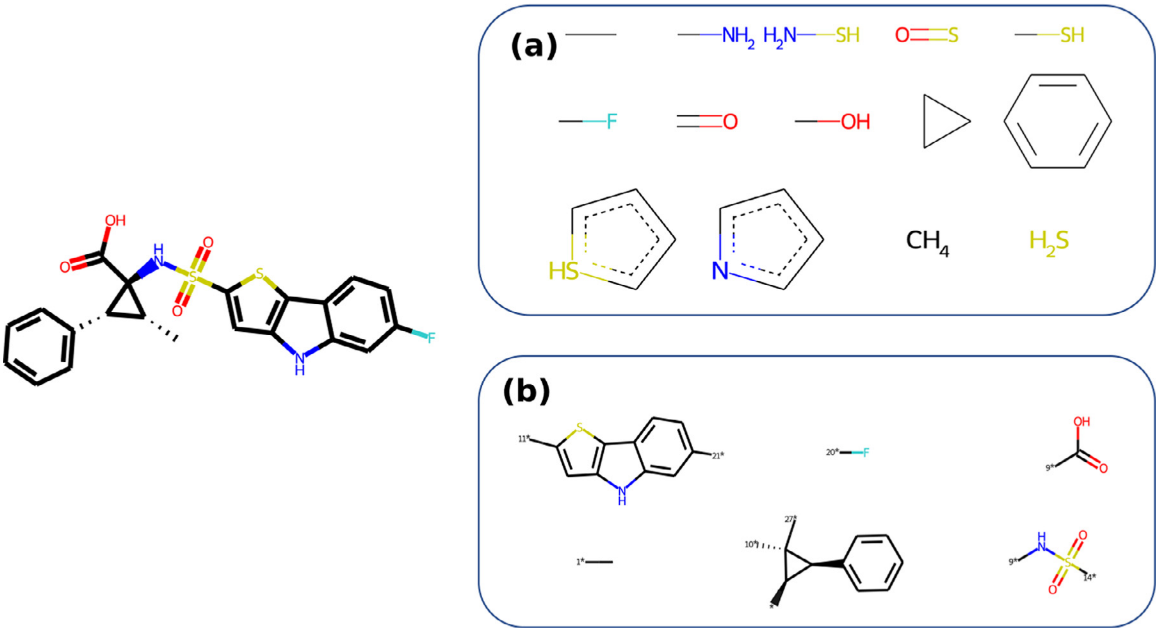

Fig. 13. Fragment features for: (a) MSSGAT^a. (b) MSSGAT^b.

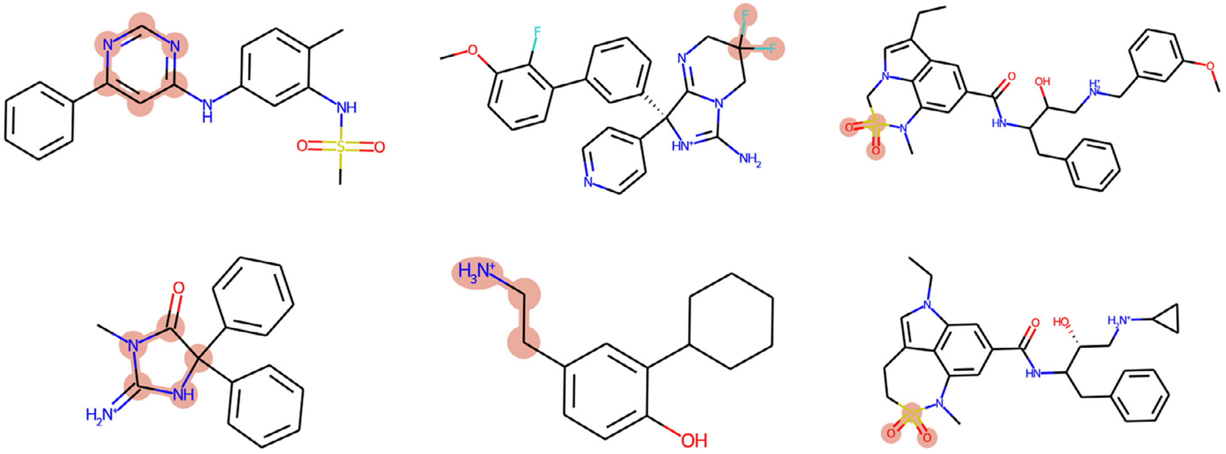

Fig. 14. Important structure visualization on the BACE data set. The atoms in orange represent the most important components indicated by MSSGAT. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 62176103, 61703182, in part by the Science and Technology Planning Project of Guangzhou, China under Grants 201902010041, 202102021173, 202102080307, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515011476, in part by the Science and Technology Planning Project of Guangdong under Grants 2021B0101420003, 2020B0909030005, 2020B1212030003, 2020ZDZX3013, 2019B1515120010, 2019A101002015, 2019KTSCX010, 2018KTSCX016, 2021A1515011873, in part by the Project of Guangdong Key Lab of Traditional Chinese Medicine Information Technology under Grant 2021B1212040007, in part by the Project of Guangxi Key Laboratory of Trusted Software under Grant kx202007, and in part by the High Performance Public Computing Service Platform of Jinan University.

Supplementary materials

Supplementary data associated with this article can be found, in the online version, at https://github.com/leaves520/MSSGAT.

References

[1] R.G. Hill, D. Richards, Drug Discovery and Development E-Book: Technology in Transition, Elsevier Health Sciences, 2021.
[2] A. Varnek, I. Baskin, Machine learning methods for property prediction in chemoinformatics: quo vadis? J. Chem. Inf. Model. 52 (6) (2012) 1413–1437.
[3] Z. Wu, B. Ramsundar, E.N. Feinberg, J. Gomes, C. Geniesse, A.S. Pappu, K. Leswing, V. Pande, Moleculenet: a benchmark for molecular machine learning, Chem. Sci. 9 (2) (2018) 513–530.
[4] T. Ching, D.S. Himmelstein, B.K. Beaulieu-Jones, A.A. Kalinin, B.T. Do, G.P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M.M. Hoffman, Opportunities and obstacles for deep learning in biology and medicine, J. R. Soc. Interface 15 (141) (2018) 20170387.
[5] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, T. Blaschke, The rise of deep learning in drug discovery, Drug Discov. Today 23 (6) (2018) 1241–1250.

[6] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. 28 (1) (1988) 31–36.
[7] M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, F. Wang, Graph convolutional networks for computational drug development and discovery, Brief. Bioinform. 21 (3) (2020) 919–935.
[8] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, in: Proceedings of The 33rd International Conference on Machine Learning, vol. 48, 2016, pp. 2014–2023.
[9] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 2, 2015, pp. 2224–2232.
[10] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, J. Computer-Aided Mol. Des. 30 (8) (2016) 595–608.
[11] Q. Zhang, J. Chang, G. Meng, S. Xu, S. Xiang, C. Pan, Learning graph structure via graph convolutional networks, Pattern Recognit. 95 (2019) 308–318, doi:10.1016/j.patcog.2019.06.012.
[12] J. Li, D. Cai, X. He, Learning graph-level representation for drug discovery, arXiv preprint arXiv:1709.03741(2017).
[13] J. Ding, R. Cheng, J. Song, X. Zhang, L. Jiao, J. Wu, Graph label prediction based on local structure characteristics representation, Pattern Recognit. 125 (2022) 108525, doi:10.1016/j.patcog.2022.108525.
[14] X. Fan, M. Gong, Y. Xie, F. Jiang, H. Li, Structured self-attention architecture for graph-level representation learning, Pattern Recognit. 100 (2020) 107084, doi:10.1016/j.patcog.2019.107084.
[15] W. Jin, R. Barzilay, T. Jaakkola, Junction tree variational autoencoder for molecular graph generation, in: Proceedings of the 35th International Conference on Machine Learning, vol. 80, 2018, pp. 2323–2332.
[16] D. Rogers, M. Hahn, Extended-connectivity fingerprints, J. Chem. Inf. Model. 50 (5) (2010) 742–754.
[17] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 1412.3555(2014).
[18] A. Gaulton, L.J. Bellis, A.P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, J.P. Overington, Chembl: a large-scale bioactivity database for drug discovery, Nucleic Acids Res. 40 (D1) (2011) D1100–D1107.
[19] J. Ma, R.P. Sheridan, A. Liaw, G.E. Dahl, V. Svetnik, Deep neural nets as a method for quantitative structure–activity relationships, J. Chem. Inf. Model. 55 (2) (2015) 263–274.
[20] J. You, R.D. McLeod, P. Hu, Predicting drug-target interaction network using deep learning model, Comput. Biol. Chem. 80 (2019) 90–101.
[21] X. Li, Y. Xu, L. Lai, J. Pei, Prediction of human cytochrome p450 inhibition using a multitask deep autoencoder neural network, Mol. Pharm. 15 (10) (2018) 4336–4345.
[22] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, in: NIPS'14, vol. 2, 2014, pp. 2672–2680.
[23] A. Kadurin, A. Aliper, A. Kazennov, P. Mamoshina, Q. Vanhaelen, K. Khrabrov, A. Zhavoronkov, The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology, Oncotarget 8 (7) (2017) 10883.
[24] C. Shang, Q. Liu, Q. Tong, J. Sun, M. Song, J. Bi, Multi-view spectral graph convolution with consistent edge attention for molecular modeling, Neurocomputing 445 (2021) 12–25.
[25] Z. Zhang, J. Guan, S. Zhou, FraGAT: a fragment-oriented multi-scale graph attention model for molecular property prediction, Bioinformatics 37 (18) (2021) 2981–2987.
[26] S. Wang, Z. Li, S. Zhang, M. Jiang, X. Wang, Z. Wei, Molecular property prediction based on a multichannel substructure graph, IEEE Access 8 (2020) 18601–18614.
[27] E.B. Lenselink, N. Ten Dijke, B. Bongers, G. Papadatos, H.W. Van Vlijmen, W. Kowalczyk, A.P. IJzerman, G.J. Van Westen, Beyond the hype: deep neural networks outperform established methods using a chembl bioactivity benchmark set, J. Cheminform. 9 (1) (2017) 1–14.
[28] G. Landrum, et al., Rdkit: Open-source cheminformatics software, 3(04) (2006) 2012. http://www.rdkit.org.
[29] M. Rarey, J.S. Dixon, Feature trees: a new molecular similarity measure based on tree matching, J. Computer-Aided Mol. Des. 12 (1998) 471–490.
[30] J. McAuley, T. Caetano, Exploiting within-clique factorizations in junction-tree algorithms, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, vol. 9, 2010, pp. 525–532.
[31] R.E. Carhart, D.H. Smith, R. Venkataraghavan, Atom pairs as molecular features in structure-activity studies: definition and applications, J. Chem. Inf. Comput. Sci. 25 (2) (1985) 64–73.
[32] R. Nilakantan, N. Bauman, J.S. Dixon, R. Venkataraghavan, Topological torsion: a new molecular descriptor for SAR applications. comparison with other descriptors, J. Chem. Inf. Comput. Sci. 27 (2) (1987) 82–85.
[33] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations (ICLR), 2017.
[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations (ICLR), 2018.
[35] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[36] Z. Xiong, D. Wang, X. Liu, F. Zhong, X. Wan, X. Li, Z. Li, X. Luo, K. Chen, H. Jiang, M. Zheng, Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism, J. Med. Chem. 63 (16) (2019) 8749–8760.
[37] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: International Conference on Machine Learning, PMLR, 2017, pp. 1263–1272.
[38] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
[39] D.P. Kingma, J.L. Ba, Adam: A method for stochastic gradient descent, in: International Conference on Learning Representations, 2015, pp. 1–15.
[40] L. McInnes, J. Healy, N. Saul, L. Grossberger, Umap: uniform manifold approximation and projection, J. Open Source Softw. 3 (29) (2018) 861.
[41] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.

Xian-Bin Ye received the B.Sc in Pharmaceutical Engineering from the School of Pharmacy, Guangdong Pharmaceutical University Guangzhou, China in 2018. He is currently a postgraduate student in College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests focus on graph neural network, drug discovery, artificial intelligence and bioinformatics.

Quanlong Guan is Professor in the faculty of computer science and core member of the research institute for Guangdong intelligent education at Jinan University, China. He is directing the Guangdong R&D Institute for the big data of service and application on education. His research interests include the application of artificial intelligence, information technology in education, data protection and processing.

Weiqi Luo received his B.S. degree from Jinan University in 1982 and Ph.D. degree from South China University of Technology in 2000. Currently, he is a professor with School of Information Science and Technology in Jinan University, Guangzhou. His research interests include net work security, big data, artificial intelligence, etc. He has published more than 100 high-quality papers in international journals and conferences

Liangda Fang received the Ph.D. degree from Sun Yat-sen University, Guangzhou, in computer science, in 2015. He is currently an Assistant Professor with the Department of Computer Science, Jinan University, Guangzhou. His current research interests include artificial intelligence, knowledge representation and reasoning, and automated planning.

Zhao-Rong Lai received the B.Sc. in mathematics, M.Sc. in computational science, Ph.D. in statistics, all from the School of Mathematics, Sun Yat-Sen University, Guangzhou, China, in 2010, 2012 and 2015,respectively. He is currently an Associate Professor with the Department of Mathematics, Jinan University, Guangzhou, China. He was an invited Senior Program Committee member of IJCAI 2021 (Session Chair as well) and IJCAI 2020. His research interests include machine learning, image processing, multivariate statistics, and portfolio optimization.

Jun Wang received his Ph.D. from Peking University in 2016. He was a Visiting Scholar in ETH Zurich in 2015. He was working as an engineer at the Institute of Software, Chinese Academy of Sciences in 2011–2012. He joined IBM Research China as a Research Scientist during 2016–2018. Since 2018, he is a Senior algorithm researcher in PingAn Technology. His research interests include Deep Learning, Medical Imaging.