Mathematical Biosciences and Engineering

MBE, 18(6): 9148–9162.
DOI: 10.3934/mbe.2021451
Received: 01 July 2021
Accepted: 09 October 2021
Published: 25 October 2021

http://www.aimspress.com/journal/MBE

Research article

Protein-ligand binding affinity prediction model based on graph attention network

Hong Yuan[1,2], Jing Huang[1,2] and Jin Li[1,2,*]

[1] School of Medical Information and Engineering, Southwest Medical University, Luzhou, China
[2] Medicine & Engineering & Informatics Fusion and Transformation Key Laboratory of Luzhou City, Luzhou, China

* **Correspondence:** Email: eddyblue@swmu.edu.cn.

Abstract: Estimating the binding affinity between proteins and drugs is very important in the application of structure-based drug design. Currently, applying machine learning to build the protein-ligand binding affinity prediction model, which is helpful to improve the performance of classical scoring functions, has attracted many scientists' attention. In this paper, we have developed an affinity prediction model called GAT-Score based on graph attention network (GAT). The protein-ligand complex is represented by a graph structure, and the atoms of protein and ligand are treated in the same manner. Two improvements are made to the original graph attention network. Firstly, a dynamic feature mechanism is designed to enable the model to deal with bond features. Secondly, a virtual super node is introduced to aggregate node-level features into graph-level features, so that the model can be used in the graph-level regression problems. PDBbind database v.2018 is used to train the model. Finally, the performance of GAT-Score was tested by the scheme C_s (Core set as the test set) and *CV* (Cross-Validation). It has been found that our results are better than most methods from machine learning models with traditional molecular descriptors.

Keywords: binding affinity; structure-based drug design; graph attention network; scoring function; machine learning

1. Introduction

Estimating the binding affinity between proteins and drugs is very important in the application of structure-based drug design (CADD), such as virtual screening, optimization of lead compounds and so on. After recognizing the binding conformation of the ligand (docking) and determining whether the ligand has biological activity related to the target protein (screening), the estimated affinity indicates the protein-ligand binding strength.

There are great differences in the success rate between the conformation with highest score and the best conformation (the conformation closest to the ligand crystal structure) predicted by the existing molecular docking software. The conformation with highest score is usually not the best conformation, which is caused by the defect of the scoring functions. Most of the molecular docking software is not designed for predicting the binding affinity of compounds, and the score is not representative of binding affinity, such as GOLD [1], GLIDE [2] and so on. The correlation between the score of these docking software and experimental binding affinity is very weak [3, 4].

Some researchers used machine learning methods to refit the weight of the descriptors used in the classical scoring functions. Li et al. [5] used a random forest algorithm based on Autodock [6]. Tanchuk et al. [7] combined the descriptors of Autodock and Autodock Vina [8], and adjusted the parameters by multiple linear regression. Although these methods improved the performance of classical scoring functions, their performance is still limited because there is no change made to the descriptors of the molecular docking software.

Then, researchers focused on generating and using of problem-specific molecular descriptors to build prediction models based on the machine learning methods. Commonly used descriptors include simplified molecular-input line-entry system (SMILES) strings [9], molecular fingerprint [10], and descriptors derived from quantum physical chemistry and differential topology [11]. However, these kinds of methods are limited by the feature engineering as the feature extraction methods directly affect the prediction results. Recently, deep learning has gained considerable attention as it allows the model to "learn" to extract features.

Deep learning has shown strong performance in image recognition [12], speech recognition [13] and natural language processing [14]. However, binding affinity prediction, based on the deep learning, faces a big challenge. Images are fixed-size grids, whereas the molecular conformation is a typical graphic structure and molecules are heterogeneous, which is hard to be processed by the deep learning methods that expect homogeneous input. Some researches [15–18] have presented the complex with a 3D grid, and utilized a 3D convolution [19] to produce a feature map of this representation. However, this model is sensitive to the orientation of the complex, and cannot identify the characteristics of the atomic bonds. Since the complex is composed of protein and ligand molecules, it can be presented as a graph where each node represents an atom of the molecule and each edge represents the bond between atoms. Graph neural network is the right choice to represent the molecular structure.

Graph neural networks (GNNs) are deep learning based methods that operate on graph domain. Nowadays, there are great developments in GNN [20,21]. Advances in this direction are often categorized as spectral approaches and spatial approaches. Spatial approaches can be categorized as basic spatial approaches and attention-based spatial approaches [22]. The attention mechanism has been successfully used in many sequence-based tasks such as machine translation [23], machine reading [24] and so on. There are also several models which try to generalize the attention operator

on graphs. Attention-based operators assign different weights for neighbors, so that they could alleviate noises and achieve better results.

In view of the advantages of attention-based spatial approaches, we have developed a protein-ligand binding affinity prediction model named GAT-Score based on the graph attention network (GAT) [25] which is a kind of attention-based spatial approaches. The protein-ligand complex is represented as a graph structure, and the atoms of protein and ligand are treated in the same matter. Two improvements are made to the original graph attention network. Firstly, a dynamic feature mechanism is designed to enable the model to deal with the bond features. Secondly, the virtual super node is introduced to aggregate node-level features into graph-level features, so that the model can be used in the graph-level regression problems. PDBbind 2018 database is used to train the model. Finally, the performance of GAT-Score is tested by the two schemes, C_s (Core set as the test set) and *CV* (Cross-Validation). The experimental results show that our model is better than most methods from machine learning models with traditional molecular descriptors.

2. Materials and methods

2.1. Data sets

The network was trained and tested with the protein-ligand complexes from the PDBbind database 2018 [26]. This database consists of 3D structures of molecular complexes and their corresponding binding affinities. Liu et al. [26] divided PDBBind complexes into 3 overlapping subsets: general set, refined set and core set. The general set (16126 complexes) includes all available data. The refined set (4463 complexes), which comprises complexes with higher quality, is subtracted from the general set. Finally, the complexes from the refined set are clustered by protein similarity, and 5 representative complexes are selected from each cluster. This fraction of the database is called the core set (285 complexes) and is designed as a high quality benchmark for structure-based CADD methods.

In order to evaluate our model with the core set of PBDbind 2013 [27], we needed to exclude all data that overlap with the 195 complexes in core set of PBDbind 2013. Therefore, a total of 87 overlapping complexes were excluded from the training set. Five complexes were part of the general set and 82 were part of the refined set.

2.2. The construction of GAT-Score model

2.2.1. Representation of the complex

According to the definition of the graph neural network [28], a graph is represented as $G = (V, E)$ where V is the set of nodes, $V_i \in V$ denotes a node in V, and E is the set of edges, $E_{ij} \in E$ denotes an edge between node V_i and V_j. N denotes the number of nodes and $\mathbf{A}$ denotes the adjacency matrix. A molecule is represented as a graph (Figure 1), the nodes of which stand for the atoms and the edges of which represent bonds between two atoms. Whether there exists an edge between the protein atom and ligand atom depends on the distance d_{PL} between the two atoms. d_{PL} is considered to be a hyper parameter. The characteristics of the graph are specified by the atomic features and the bond features. Adjacency matrix only represents the connectivity between atomic

pairs and does not include explicit edge features. We used the open source Python toolkit RDKit [29] to acquire the features of nodes and edges and the adjacency matrix of the graph.

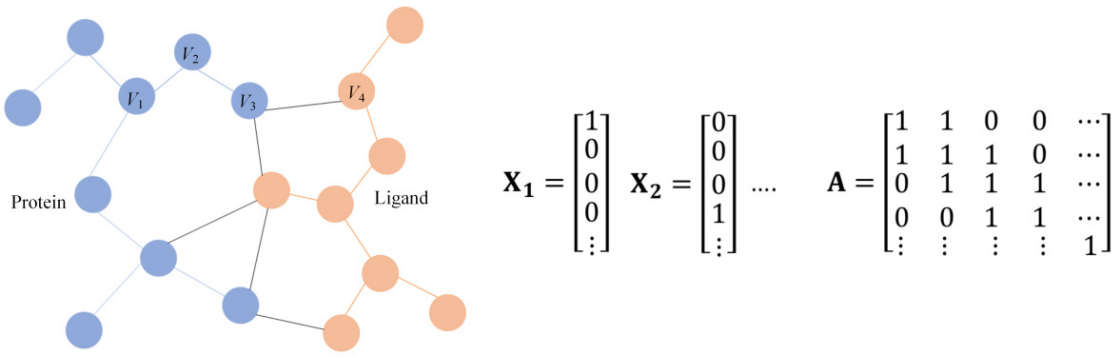


Figure 1. Graph structure representation of the complex. The blue nodes denote the atoms of the protein, and the orange nodes denote the atoms of the ligand. $\mathbf{X_1}$ denotes the feature vector of node V_1, and $\mathbf{X_2}$ denotes the feature vector of node V_2. $\mathbf{A}$ denotes the adjacency matrix.

Inspired by Stepniewska-dziubinska [15] and Liu [30], the following feature representation is used (Table 1).

Table 1. Features of the complex.

Type	Feature	Data type	Size
atomic feature	Atomtype	9 integer (0 or 1)	9
atomic feature	Hybridization	integer (1,2 or 3)	1
atomic feature	Hydrophobic	integer (0 or 1)	1
atomic feature	Aromatic	integer (0 or 1)	1
atomic feature	Acceptor	integer (0 or 1)	1
atomic feature	Donor	integer (0 or 1)	1
atomic feature	Ring	integer (0 or 1)	6
atomic feature	Partialcharge	float	1
atomic feature	Moltype	integer (1 or −1)	1
bond feature	Bond type	integer (0, 1, 2 or 3)	1
bond feature	Same ring	integer (0 or 1)	1
bond feature	Distance	float	1

The following atomic features are used:

1) Atom type: B, C, N, O, P, S, Se, halogen and metal, one-hot with 9 bits, denoted as Atomtype.

2) Atom hybridization: Hybridization, encoding with 1 integer (1, 2, 3).

3) Properties defined with SMARTS patterns: Hydrophobic, aromatic, acceptor, donor, encoding with 1 bit (1 if present)

4) Whether the atom is in an aromatic ring with size 3 to 8, one-hot encoding by 6 bits.

5) Partial charge: represented by 1 float. The partial charges were obtained by pybel—A Python

module that simplifies access to the Open Babel API.

6) Distinguish between the protein and ligand: Moltype, represented by 1 integer (1 for ligand, −1 for proteins).

The following bond features (edge features) are used:

1) The number of bonds between atomic pair, represented by an integer.

2) Whether the two atoms is in the same ring, represented by 1 or 0.

3) Euclidean distance between atom and atom is calculated by the three-dimensional coordinates of two atoms.

The node feature vector is formed by the atomic feature and bond feature by dynamic feature mechanism which is described in the next section. The node feature is a 25 dimensional vector and changes dynamically according to the different aggregation nodes.

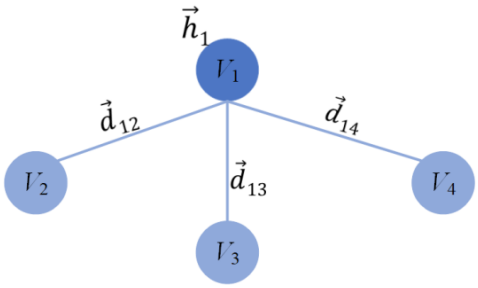
2.2.2. Dynamic feature mechanism

GAT [25] is not designed to make use of the edge information in the graph, because it only uses the connectivity of the nodes. The value of 1 in the adjacency matrix means that there exists an edge between the two nodes, and the value of 0 means that there does not exist an edge (Figure 1). However, the edge in the molecular graph in our study has a lot of information, such as Euclidean distance between two atoms, the type of atomic bond and whether two bonded atoms are in the same ring. Making full use of the edge information is helpful to extract the features of the graph effectively. Therefore, we designed dynamic feature mechanism to make improvement to the attention layer of GAT. The attention layer has sufficient expressive power to transform the input features into higher-level features. One of the steps is computing attention coefficients e_{ij} (Eq (1)), which indicates the importance of the features of node V_j to node V_i.

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j) \tag{1}$$

where $\mathbf{W}$ denotes a weight matrix which is applied to every node to execute a learnable linear transformation. Attention mechanism a, which is a single-layer feedforward neural network and applying the LeakyReLU nonlinearity, is performed to compute attention coefficients. Consequently, the attention coefficients are used to compute a linear combination of the features corresponding to then, to serve as the final output features for every node (after potentially applying the eLU nonlinearity). The feature of node V_i and V_j is denoted as $\vec{h}_i$ and $\vec{h}_j$. The detailed explanation of attention mechanism is in the paper of Velikovi et al. [25]

In our study, in order to get the edge information, we concatenated the edge feature of E_{ij} to the atomic feature of V_j as $\vec{h}_j$. Since V_j connects different neighbour nodes, the node feature of V_j changes dynamically when calculating the attention coefficient between V_j and different neighbour nodes. In the operation of aggregating neighbour nodes in the graph attention layer, we can effectively aggregate the edge features.



$$\vec{h}_1 \leftarrow \begin{cases} \text{concat}\,(\vec{h}_1,\ \vec{d}_{12}\,), & \text{for} \quad e_{21} \\ \text{concat}\,(\vec{h}_1,\ \vec{d}_{13}\,), & \text{for} \quad e_{31} \\ \text{concat}\,(\vec{h}_1,\ \vec{d}_{14}\,), & \text{for} \quad e_{41} \end{cases}$$

Figure 2. Dynamic feature mechanism.

The input of the attention layer is a set of node feature matrix $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}$, $\vec{h}_i \in \mathbb{R}^F$ and a set of edge features $\mathbf{d} = \{\vec{d}_{ij}\}$, $\vec{d}_{ij} \in \mathbb{R}^M$. Where N is the number of nodes, F is the characteristic number of each node, and $\vec{d}_{ij}$ is the feature vector of the edge of any node V_j and V_i, and M is the characteristic number of each edge. In the attention layer with dynamic feature mechanism, $\vec{h}_i$, $\vec{h}_j$ in the original formula in GAT [25] are changed as follows.

$$\vec{h}_i \leftarrow concat(\vec{h}_i, \vec{d}_{ii}) \qquad \vec{h}_j \leftarrow concat(\vec{h}_j, \vec{d}_{ji}) \tag{2}$$

Next, we use an example to illustrate. The edge feature vector of the edge E_{ij} is [distance, bondtype, same ring], and the edge feature vector of the edge E_{ii} is [0,0,0]. As shown in Figure 2, suppose node V_1 connects three neighbour nodes, V_2, V_3, V_4, and the features of three edges are $\vec{d}_{12}$, $\vec{d}_{13}$, $\vec{d}_{14}$. For V_2, V_3, V_4, The feature vectors of V_1 are different. When V_2, V_3, V_4 aggregates their neighbour node V_1 respectively, the feature vector of V_1 changes dynamically.

2.2.3. Virtual super node

Learning graph-level is a central problem of molecule classification and regression. In the attention layer, higher-level features are learned for every node, but a graph-level presentation is needed to make graph-level prediction. The process is called readout. There are two kinds of methods for readout operation: statistic-based method and learning-based method. Statistic-based methods are the most common, such as Mean, Sum and Max. These methods without additional parameters are simple and effective, but it is obvious that these methods are prone to loss the information of the graph. Therefore, a learning-based approach is adopted.

In order to learn graph-level feature and utilize GAT network for graph-level properties prediction, inspired by Scarselli et al. [28], we introduced a virtual super node that is connected with all nodes in the graph by a directed edge (Figure 3). Since the virtual super node is directly connected with all nodes in graph, it can easily learn global feature through one graph attention layer. The directed edge pointed to the virtual super node from other genuine nodes, indicates that the virtual super node could learn features from all other genuine nodes (Figure 3(a)), while none of the genuine nodes would be affected by the virtual super node (Figure 3(b)). Consequently, the virtual super node could learn the global feature while the genuine nodes keep learning local features. Since the feature of the graph is more complex than that of the node, we used a longer vector as the feature of the virtual super node. Therefore, we can deal with graph-level regression as we do with node-level regression.

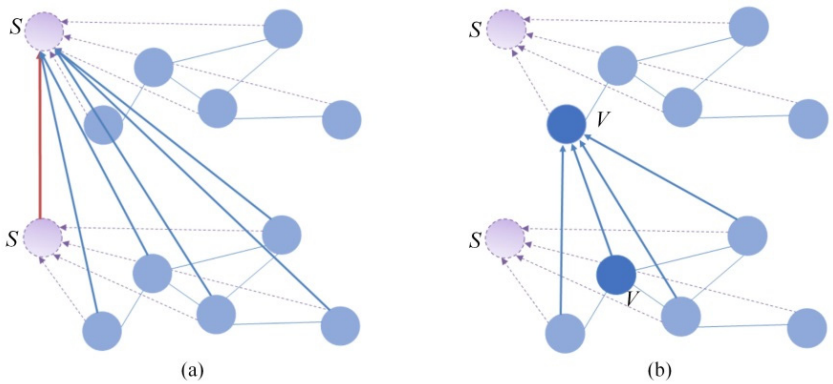


Figure 3. Graphical representation of the major graph operation. The dark blue node represents the specific node *V*, the light blue nodes represent the genuine nodes, the dotted node represents the virtual super node *S*. (a) the virtual super node *S* aggregates all nodes. (b) the specific node *V* aggregates the neighbour nodes.

2.2.4. The architecture of GAT-Score

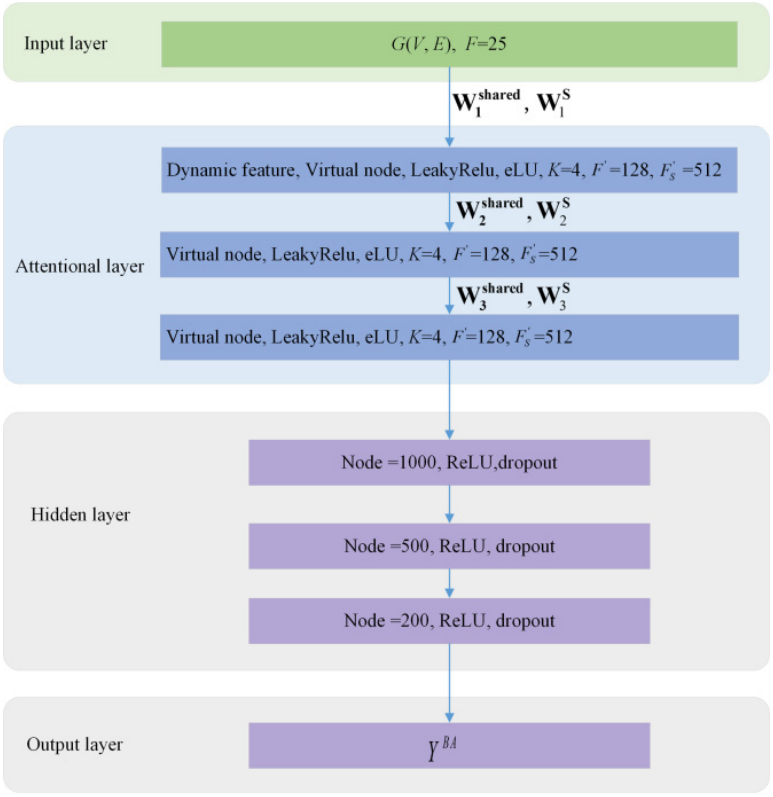


Figure 4. The network architecture of GAT-Score.

The network architecture of GAT-Score is shown in Figure 4. The neural network we proposed here consists of eight layers including an input layer, three attention layers, three hidden layers and an output layer. The input layer includes node features (25 dimensional vectors) and connectivity

information. Since the number of atoms varies between molecules, standard batch normalization cannot be applied to a graph convolution network directly. We extended standard batch normalization to node-level batch normalization. We normalized the feature of each node, and made it zero mean and unit variance. Three attention layers extract low-level representations. The dynamic feature mechanism is introduced into the first attention layer. The weight matrixes, denoted as $\mathbf{W}_1^{\text{shared}}$, $\mathbf{W}_2^{\text{shared}}$ and $\mathbf{W}_3^{\text{shared}}$, are applied to the genuine nodes for learnable linear transformation in the three attention layers respectively, while the weight matrixes, denote as $\mathbf{W}_1^S$, $\mathbf{W}_2^S$ and $\mathbf{W}_3^S$, are applied to the virtual super nodes. Our model used glorot_normal as initial method and the eLU activation [31]. To stabilize the learning process of self-attention, we have employed multi-head attention (with $K = 4$ heads) [25] by each node on its neighbourhood. K independent attention mechanisms execute the transformation, and then their features are concatenated. The attention layer produces a new set of the genuine node features (the dimension of the feature vector is $F' \times 4 = 128 \times 4 = 512$) and the virtual node features (the dimension of the feature vector is $\mathrm{F}_S' \times 4 = 512 \times 4 = 2048$). The feature of the virtual node in the last attention layer is used as an input for a dense layer with 1000 neurons, which is connected with two dense layers with 500 and 200 neurons. In order to improve generalization, dropout with drop probability of 0.2 is used for all dense layers. The dense layers are composed of rectified linear units (ReLU). ReLU is chosen because it speeds up the learning process compared with other types of activations. The output layer is a regression task (Binding affinity prediction). The mean square error (MSE) is used as the loss function. In the training process, the parameters of neurons are optimized by Adam optimizer to minimize the final loss function. The learning rate was set to 0.0001. The detailed parameters of the model are shown in Table 2.

Table 2. Features of the complex.

Super parameter	Value
Bath size	40
Optimizer	Adam
Learning rate	0.0001
Decay rate	0.97
Dropout rate	0.2
Number of epochs	80
d_{PL}	4 Å

2.2.5. Training process and model selection

Epoch is defined as the number of times that the whole training set is repeatedly trained. In the graph neural network, it is impossible to feed all the samples to the model at one time because of the huge number of parameters and samples, so it is necessary to train in batches. In the training phase, the batch size of each iteration is set to 40. It is necessary to choose an appropriate epoch value, because less epochs may lead to underfitting of the model, and more epochs may lead to overfitting of the model. The network structure trained 80 epochs on the training set. Each iteration can produce a model, and each model may be the final model. The root mean square error (RMSE) of the validation set is used to show the performance of each model. RMSE decreases with the increase of training times, and the model will converge at a certain epoch. A model with the minimum RMSE is chosen when it converges.

3. Results

For each complex in the test set, binding affinity (in pK_d/pK_i) was predicted and compared to the real value. Prediction error was measured with RMSE (in pK_d/pK_i). The correlation between the scores and experimentally measured binding constants was assessed with Pearson correlation coefficient (R_p) and Spearman correlation coefficient (R_s).

An important goal in this work is to objectively estimate the generalization accuracy of the proposed GAT-Score based on their familiarity with the test proteins and ligands. More specifically, our objective is to investigate how our model performs on test protein targets that are (i) already present in the training set (but bound to different ligands) and (ii) partially or fully unrepresented in their training samples. Accordingly, inspired by Hossam et al. [32], we designed two different training-test sampling strategies from the 16126 complex general set of PDBbind to evaluate our model in real-world modeling scenarios based on the degree of overlap between the training and test proteins and ligands.

3.1. Multifold cross-validation: novel complexes with known and novel protein targets and ligands

One of the testing schemes is based on 10-fold cross-validation (CV) where the general set of PDBbind 2018 is shuffled randomly and then partitioned into 10 nonoverlapping subsets of equal size. Upon training and validation, one out of the 10 subsets is used for testing, and the remaining nine are combined for training. Once the training and test round completes for this fold, the same process is repeated for the other nine folds one at a time. In a typical 10-fold CV experiment on a data set of 16126 complexes, the sizes of the 10 folds for training and test are (16126×9/10 ≈ 14513) and (16126×1/10 ≈ 1613) complexes, respectively. Every protein and ligand family is not necessarily present in both the training and test sets across the 10 folds due to the randomness of CV. Therefore, some proteins and ligands in the test set may actually be "novel" for the model while others may be present in the training data.

Table 3. The mean R_p, R_s and RMSE (in pK_d/pK_i) of GAT-Score and RF-Score on the scheme CV and C_s.

		GAT-Score	RF-Score	p-value
CV	R_p	0.772	0.719	< 2.2e–16
	R_s	0.764	0.702	< 2.2e–16
	RMSE	1.47	1.55	___
C_s	R_p	0.778	0.728	< 2.2e–16
	R_s	0.769	0.712	< 2.2e–16
	RMSE	1.43	1.52	___

We compared GAT-Score with RF-Score on the experimental scheme CV. RF-Score is the most popular machine learning based scoring function recently, and it outperforms classical scoring functions and other feature-based machine learning methods in affinity scoring. RF-Score is often used as a benchmark method. GAT-Score and RF-Score have been repeated for 30 times respectively and a median R_p, R_s and RMSE (for test set) are obtained as the final results. The results are illustrated in Table 3. GAT-Score achieved better performance than RF-Score in R_p and R_s. Maybe it is because that RF-Score is based on feature engineering and its prediction performance is greatly

affected by feature selection, whereas automatic feature extraction of GAT-Score based on GAT is helpful to improve the performance of our model.

3.2. PDBbind core test set: novel complexes with known protein targets and ligands

The core set has been a popular benchmarking test set for many molecular docking software and scoring functions [33–37]. The core set is denoted as C_s. The corresponding training set for C_s, referred to as the primary training set and denoted as P_r, was formed by removing all C_s complexes from the total 16126 complexes in the general set. As a result, P_r contains (16126 – 285 =) 15841 complexes that are disjoint from the complexes in the core set. The validation set is formed by randomly selecting 1000 complexes form P_r to help evaluate the variance of our model.

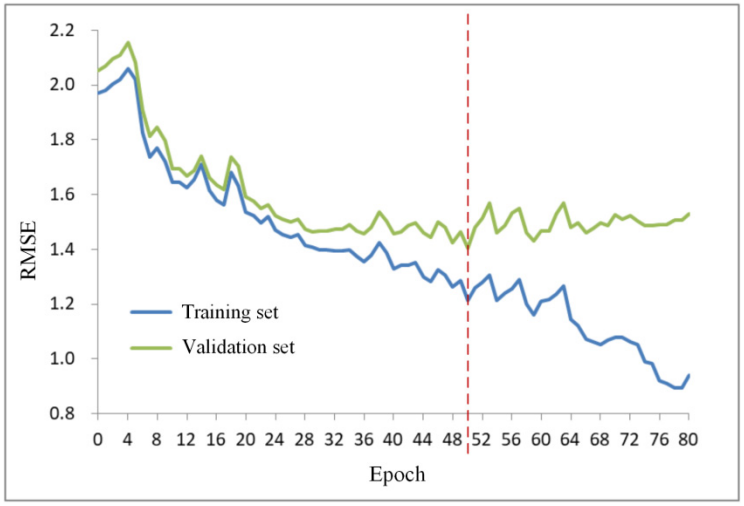


Figure 5. The RMSE of different epochs in C_s scheme.

Due to the overlap between their training and test proteins, the core test set complexes are considered targets that are "known" to scoring functions. More specifically, for each protein in the protein clusters of C_s, there is at least one identical protein present in the primary P_r training data, albeit bound to a different ligand. In addition, every ligand in the core test set has at least one training ligand that shares the same ligand cluster. Therefore, the complexes in C_s have some degree of similarity to training complexes in P_r in terms of both the protein and ligand present.

Table 4. GAT-Score's performance in one experiment. The unit of RMSE and SD is pK_d/pK_i.

Dataset	RMSE	SD	Rp
Test	1.389	1.36	0.786
Validation	1.43	1.40	0.764
Training	1.23	1.18	0.803

In the C_s scheme, the model was trained for 80 epochs on the training set, as shown in Figure 5. Each iteration can produce a model with different parameters. We used RMSE to evaluate the quality

of each model. After 50 epochs of training, the model began to overfit, and the error on the validation set began to increase slowly and steadily. The best set of weights of the network, obtained after 50 epochs of training, was saved and used as the final model. Model performance was evaluated on all subsets of the data. For each complex in the dataset, the predicted affinity was compared with its real value. R_p, RMSE and SD (standard deviation) for three subsets are illustrated in Table 4. As expected, the network achieved the minimum error on the training set (RMSE is 1.23), which is used to fit the weights of the network. R_p on the validation set is 0.764, while R_p on the core set is 0.786 (Figure 6). When we repeated GAT-Score and RF-Score for 30 times respectively, a median R_p of GAT-Score is 0.778 and R_s of GAT-Score is 0.769, which is still better than RF-Score. The detailed results are illustrated in Table 3.

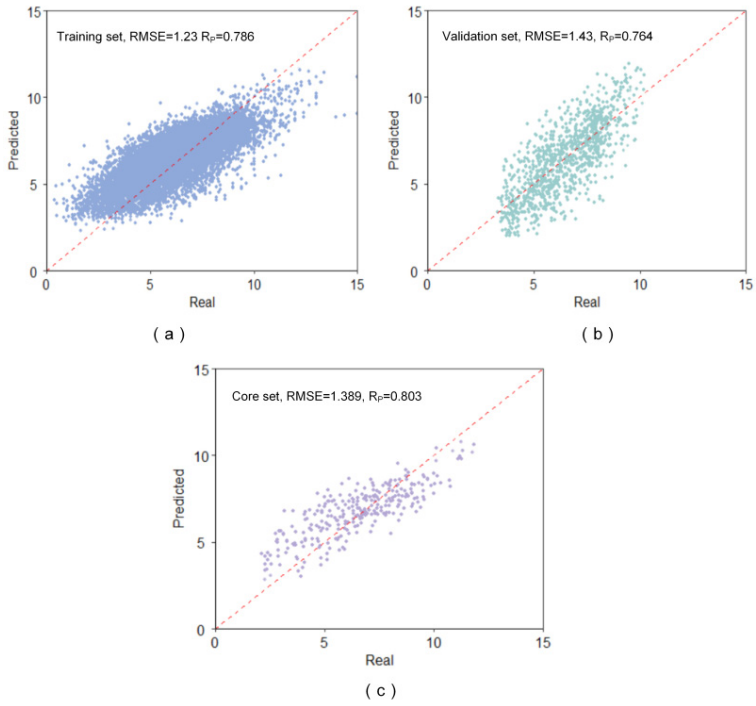


Figure 6. The results of C_s scheme in one experiment.

Furthermore, to have a better understanding of the performance of our models, we used the core set of PDBbind 2013 as the test set, and compared R_p with the state-of-the-art results in literature [38,39]. The results are illustrated in Figure 7. It can be seen that our model is better than classical scoring functions and some machine-learning based methods. GAT-Score is not better than the recently proposed models, such as PerSpect-GBT, PSH-ML and FPRC, but R_p is very close.

Due to the complexity of deep neural network, it is always criticized for its lack of interpretability. However, for a better understanding of the protein-ligand interactions, the interpretability of protein-ligand interaction prediction model is very important. Therefore, designing architectures that have the ability of interpretation or visualization of protein-ligand interactions is both a challenge and an opportunity for the application of GNN in drug discovery. Our model used the attention mechanism to extract the important information of the graphs and produce the characteristics with attention perception. Attention-based operators assign different weights for the neighbors. The different weights represent the different importance of edges and atoms. Analysing

the learned weights may help to improve the explanatory ability. Therefore, we can further study the visualization of this model and the analysis method of the key structural characteristics for protein-ligand binding, which can help to clarify the reason for protein-ligand binding. Although the performance of our model is not the best among the current affinity prediction methods, it may contribute to improving the interpretability of the model.

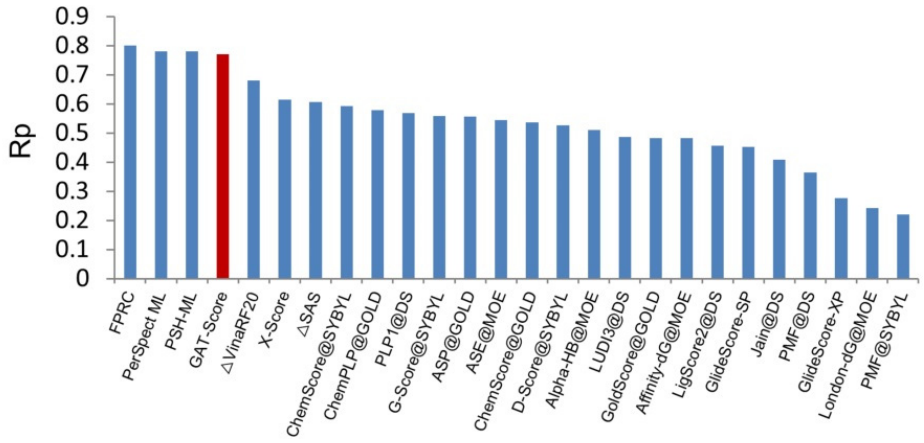


Figure 7. The comparison of our GAT-Score model and the state-of-art machine learning models, for the prediction of protein-ligand binding on PDBbind 2013.

The experiments on the scheme C_s and CV show the good performance of GAT-Score. Although both C_s and CV test sets contain fully or partially overlapping targets with the training set, it has been found that more than 90% of recent drug targets are known with complexes already deposited in PDB database [40], which the complexes in our training set come from. This means that our model will be applied to known targets in the vast majority for real-world drug development applications.

4. Conclusions

In this paper, we developed GAT-Score, a graph attention network based model to predict protein-ligand binding affinity. We firstly utilized the graph attention mechanism of GAT to enable this model to assign different importance (weight) to different nodes within a neighbourhood; therefore, our model can produce the features with attention perception. In order to learn the information of bonds in the molecular, we proposed dynamic feature mechanism to solve the problem that GAT cannot deal with the edge feature. In order to learn graph-level representation, we introduced the virtual super node that is connected with all nodes in the graph by a directed edge and modify the graph operation to help the model learn graph-level features. Thus, the model can handle graph-level regression in the same way as node-level regression. The experiments C_s and CV shows that GAT-Score achieves a better performance than the classical scoring functions and most machine learning based methods with traditional molecular descriptors.

Acknowledgments

The project was funded by Medicine & Engineering & Informatics Fusion and Transformation Key Laboratory of Luzhou City, Project of Sichuan health information society (NO. 2021004) and the Introducing Talent Start-up Foundation of Southwest Medical University (NO. 40/00040142).

Conflict of interest

The authors declare no competing interests.

References

1. M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, R. D. Taylor, Improved protein–ligand docking using GOLD, *Proteins*, **52** (2003), 609–623.
2. R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, et al., Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy, *J. Med. Chem.*, **47** (2004), 1739–1749.
3. W. Zhe, H. Sun, X. Yao, L. Dan, T. Hou, Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power, *Phys. Chem. Chem. Phys.*, **18** (2016), 1–27.
4. Z. Gaieb, S. Liu, S. Gathiaka, M. Chiu, H. Yang, C. Shao, et al., D3R Grand Challenge 2: blind prediction of protein–ligand poses, affinity rankings, and relative binding free energies, *J. Comput. Aid. Mol. Des.*, **32** (2018), 1–20.
5. H. Li, K. S. Leung, M. H. Wong, P. J. Ballester, Improving AutoDock Vina using random forest: the growing accuracy of binding affinity prediction by the effective exploitation of larger data sets, *Mol. Biol.*, **34** (2015), 115–126.
6. G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, et al., Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function, *J. Comput. Chem.*, **19** (1998), 1639–1662.
7. V. Y. Tanchuk, V. O. Tanin, A. I. Vovk, G. Poda, A new, improved hybrid scoring function for molecular docking and scoring based on AutoDock and AutoDock Vina, *Chem. Biol. Drug Des.*, **87** (2016), 618–625.
8. O. Trott, A. J. Olson, Software news and update AutoDock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading, *J. Comput. Chem.*, **31** (2010), 455–461.
9. A. A. Toropov, A. P. Toropova, R. G. Diaza, E. Benfenati, G. Gini, SMILES–based optimal descriptors: QSAR modeling of estrogen receptor binding affinity by correlation balance, *Struct. Chem.*, **23** (2011), 529–544.
10. M. Wójcikowski, M. Kukiełka, M. M. Stepniewska-Dziubinska, P. Siedlecki, Development of a protein-ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions, *Bioinformatics*, **35** (2018), 1334–1344.
11. C. Zixuan, G. W. Wei, R. L. Dunbrack, TopologyNet: Topology based deep convolutional and multi–task neural networks for biomolecular property predictions, *PLoS Comput. Biol.*, **13** (2017), e1005690.

12. H. Zhai, Research on image recognition based on deep learning technology, in *International Conference on Advanced Materials and Information Technology Processing*, (2016), 266–270.
13. B. J. Abbaschian, D. Sierra-Sosa, A. Elmaghraby, Deep learning techniques for speech emotion recognition, from databases to models, *Sensors*, **21** (2021), 1249.
14. P. Klosowski, Deep learning for natural language processing and language modelling, in *2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, (2018), 223–228.
15. M. M. Stepniewska-dziubinska, P. Zielenkiewicz, P. Siedlecki, Development and evaluation of a deep learning model for protein–ligand binding affinity prediction, *Bioinformatics*, **34** (2018), 3666–3674.
16. Y. Li, M. A. Rezaei, C. Li, X. Li, D. Wu, DeepAtom: A framework for protein-ligand binding affinity prediction, in *IEEE International Conference on Bioinformatics and Biomedicine*, (2019), 303–310.
17. M. Ragoza, J. Hochuli, E. Idrobo, J. Sunseri, D. R. Koes, Protein-ligand scoring with convolutional neural networks, *J. Chem. Inf. Model.*, **57** (2017), 942–957.
18. I. Wallach, M. Dzamba, A. Heifets, AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery, preprint, arXiv:1510.02855.
19. D. Mishkin, N. Sergievskiy, J. Matas, Systematic evaluation of convolution neural network advances on the Imagenet, *Comput. Vis. Image Und.*, **161** (2017), 11–19.
20. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Netw. Learn. Syst.*, **32** (2019), 4–24.
21. J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, et al., Graph neural networks: A review of methods and applications, *AI Open*, **1** (2021), 57–81.
22. S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, *Comput. Soc. Netw.*, **6** (2019), 1–23.
23. D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in *International Conference on Learning Representations*, 2015.
24. J. Cheng, L. Dong, M. Lapata, Long short-term memory-networks for machine reading, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (2016), 551–561.
25. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, preprint, arXiv:1710.10903.
26. M. Segler, T. Kogej, C. Tyrchan, M. P. Waller, Generating focused molecule libraries for drug discovery with recurrent neural networks, *ACS Cent. Sci.*, **4** (2018), 120–131.
27. Y. Li, M. Su, Z. Liu, J. Li, J. Liu, L. Han, et al., Assessing protein–ligand interaction scoring functions with the CASF–2013 benchmark, *Nat. Protoc.*, **13** (2018), 666–680.
28. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Trans. Neural Netw.*, **20** (2009), 61–80.
29. *RDKit*, Available from: http://www.rdkit.org/.
30. K. Liu, X. Sun, L. Jia, J. Ma, H. Xing, J. Wu, et al., Chemi–net: a graph convolutional network for accurate drug property prediction, *Int. J. Mol. Sci.*, **20** (2018), 3389.
31. S. Ioffe, C. Szegedy, Batch normalization: Acceleration deep network training by reducing internal covariate shift, in *International Conference on Machine Learning*, (2015), 448–456.

32. M. A. Hossam, R. M. Nihar, Task-specific scoring functions for predicting ligand binding poses and affinity and for screening enrichment, *J. Chem. Inf. Model*, **58** (2018), 119–132.
33. H. Ashtawy, N. Mahapatra, A comparative assessment of predictive accuracies of conventional and machine learning scoring functions for protein-ligand binding affinity prediction, *IEEE ACM Trans. Comput. Biol. Bioinf.*, **12** (2010), 335–347.
34. Y. Li, L. Han, Z. Liu, R. Wang, Comparative assessment of scoring functions on an updated benchmark: 2. Evaluation methods and general results, *J. Chem. Inf. Model.*, **54** (2014), 1717–1736.
35. T. Cheng, X. Li, Y. Li, Z. Liu, R. Wang, Comparative assessment of scoring functions on a diverse test set, *J. Chem. Inf. Model.*, **49** (2009), 1079–1093.
36. H. M. Ashtawy, N. R. Mahapatra, BgN-Score and BsN-Score: Bagging and boosting based ensemble neural networks scoring functions for accurate binding affinity prediction of protein–ligand complexes, *BMC Bioinf.*, **16** (2015), 1–12.
37. H. M. Ashtawy, N. R. Mahapatra, Machine-learning scoring functions for identifying native poses of ligands docked to known and novel proteins, *BMC Bioinf.*, **16** (2015), S3.
38. Z. Meng, K. Xia, Persistent spectral–based machine learning (PerSpect ML) for protein–ligand binding affinity prediction, *Sci. Adv.*, **7** (2021), eabc5329.
39. M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li, et al., Comparative assessment of scoring functions: the CASF-2016 update, *J. Chem. Inf. Model.*, **59** (2018), 895–913.
40. *RCSB PDB*, Available from: http://www.rcsb.org/.

AIMS Press