Protein 3D structure-based neural networks highly improve the accuracy in compound-protein binding affinity prediction

Binjie Guo[1,2,6], Hanyu Zheng[1,2,6], Haohan Jiang[1,2,6], Xiaodan Li[2], Naiyu Guan[2], Yanming Zuo[2], Yicheng Zhang[2], Hengfu Yang[4]*, Xuhua Wang[1,2,3,5]*

Affiliations:

1. Department of Neurobiology, Zhejiang University School of Medicine, Hangzhou, Zhejiang Province 310003, P. R. China
2. NHC and CAMS Key Laboratory of Medical Neurobiology, MOE Frontier Science Center for Brain Research and Brain-Machine Integration, School of Brain Science and Brain Medicine, Zhejiang University, Hangzhou, Zhejiang Province, 310003, P. R. China
3. Co-innovation Center of Neuroregeneration, Nantong University, Nantong, 226001 Jiangsu, PR China
4. School of Computer Science, Hunan First Normal University, Changsha, 410205 Hunan, PR China
5. Lead Contact
6. These authors contributed equally

* Correspondence: xhw@zju.edu.cn (X.W.), hengfuyang@hnfnu.edu.cn (H.Y.)

Abstract

Theoretically, the accuracy of computational models in compound-protein binding affinities (CPAs) could be improved by the introduction of protein 3D structure information. However, most of these models still suffer from low accuracy due to the lack of an efficient approach to encode informative protein features. The major challenge is how to combine the multi-modal information such as the residue sequence of the protein, residue atom coordinates and the torsion angles. To tackle this problem, we develop **F**ast **E**volutional **A**ttention and **T**horoughgoing-graph **N**eural **N**etworks (FeatNN) to facilitate the application of protein 3D structure information for predicting CPAs. Specifically, we established a novel end-to-end architecture to jointly embed torsion matrix, discrete distance matrix, and sequence information of protein and extract compound features with deep graph convolution layers. In addition, a new pairwise mapping attention mechanism is introduced to comprehensively learn potential interaction information between proteins and compounds. FeatNN considerably outperforms various state-of-the-art baselines in CPA prediction with the R^2 coefficient elevated by about 21.33%. Thus, FeatNN provides an outstanding method for highly accurate CPA prediction and facilitates high-throughput virtual screening of drug candidates.

Introduction

Effective high-throughput virtual screening using computational methods could greatly accelerate the drug candidate identification process through learning and estimating the compound-protein binding affinities (CPAs, they are commonly measured by IC_{50}, K_i and K_d. In this study, we use "KIKD" to refer to the combination by K_i-measured data and K_d-measured data due to their high homogeneity)[1,2], especially under the circumstance that a great number of data sources for compound and protein interaction are available through data sharing programs. For example, PubChem currently contains more than 111 million compounds structures and 271 million bioactivity data points[3], which substantially increases the potentiality for *in silico* prediction of CPAs. However, even with these abundant data, accurately predicting CPAs is still the fundamental challenge to apply to practical drug candidate screening. To increase the success rate of CPA prediction, the development of computational methods has proceeded along either structure-free or structure-based paths. Despite some substantial advancements, each path has met the challenge to further increase its accuracy in CPA prediction.

The structure-free models regard the compound (drug) as a list of atoms and protein (target) as a list of residues, and compute the pairwise atom-residue distances to form a [number of atoms]-by-[number of residues] pairwise interaction matrix, which, under a certain threshold, a compound atom is considered to be in contact (i.e. non-covalent interaction) with the corresponding protein residue. In the pairwise interaction matrix, each element is a binary value indicating whether the corresponding atom-residue pair has an interaction or not. These structure-free models[4-8] rely solely on the sequences of the compounds and proteins to learn the interactions from the pairwise matrices, with the aim to predict the binding affinities between them. For example, multi-layer 1-dimensional convolutional neural networks (1d-CNNs) are utilized to extract the features from residue sequences of protein[9-11]. The obtained matrices were used to represent the features of proteins and then acted as the input for another downstream deep learning (DL) model in the pipeline to ultimately learn the CPAs. However, the interaction between compound and protein is temporal since the structure of a

protein is dynamic, and the pairwise information being captured is only a temporarily short stage[12]. Feeding this information, with neglect of the 3D spatial structure underpins the non-covalent interactions between compounds and the binding pockets, to model training may not produce a robust outcome for CPA prediction. Therefore, models built upon this approach are unlikely to further improve their accuracy in CPA prediction.

Meanwhile, structure-based approaches were also proposed for CPA prediction, one of which is the docking simulations[13]. The docking method computationally simulates the potential binding sites and 3D structures of the compound-protein complexes during CPA prediction, and it heavily depends on the availability of high-quality 3D-structure data of proteins. Despite a few successful stories, this method is severely limited due to the scarcity of high-quality 3D-structure data of proteins[14]. To overcome this limitation, machine learning algorithm-based approaches are proposed to represent the 3D-structure information of proteins. These models were fed with not only the sequence data of the proteins but also their spatial three-dimensional information with the hope to gain an insightful ability to predict CPA[15]. For instance, the structure features of protein were extracted through 3D atomic representation in voxel space by applying 3D CNN[16], and compound features were embedded by graph convolutional networks (GCN)[17]. However, their accuracies in CPA prediction were not significantly improved, possibly because the pairwise interaction and structure features of protein were not comprehensively considered. Thus, we attempt to develop a new deep learning model that can utilize both pairwise interaction and protein 3D structure information for CPA prediction.

Inspired by the structure-free affinity prediction model and AlphaFold2's protein characterization method[18], we established an end-to-end neural network architecture, named Fast Evolutional Attention and Thoroughgoing-graph Neural Networks (FeatNN). Based on the state-of-the-art deep learning technology, this model is able to simultaneously address both the over-smoothing problem in deep graph CNNs and the high-dimensional protein

representation problem. The architecture of FeatNN mainly includes 3 modules, namely, the protein extractor module, compound extractor module and affinity prediction module (Fig.1). The evaluation of our model on KIKD and IC_{50} datasets show remarkable results with 21.33% improvement over the second greatest benchmark model in IC_{50} of the R^2 metrics and 17.07% in KIKD.

The main contributions of this study are listed as follows:

1) An embedding method (protein extractor module) is innovatively proposed to better represent the protein features by combining the residue sequence of the protein, the distance matrix and the torsion matrix.

2) An interactive attention mechanism is introduced in the affinity prediction module to comprehensively learn the potential interaction information between proteins and compounds.

3) The proposed model outperforms various state-of-the-art models by a great margin.

Results

The design logic of FeatNN with the input of protein 3D structure information

Previous studies suggested that the predicted pairwise interaction could be utilized in structure-free models to further improve their accuracy in CPA prediction[8]. However, these results might be misinterpreted, because their evaluation metrics, Area Under the Curve (AUC), was misused on the imbalanced data (for more details, please refer to Supplementary Note 1.1) and the predicted pairwise interaction matrices were intermediate results generated during the model training process, which were inappropriately fed into the original model for CPA prediction. Mathematically, any intermediate result generated during the machine learning model training process is only a representation of the input features (i.e., the graph representation of compound, residue sequence of the protein, and pairwise interaction matrix). When this representation is fed along with its upstream features to predict a final response (i.e., the CPA), it is still the representation of their original input features, in other words, there is no new information introduced to this model. In our experiments, we found the

action of feeding the intermediate result along with the original features into a structure-free model could not benefit CPA prediction (Supplementary Fig. 1), indicating that the structure-free models are incapable to improve their accuracy in CPA prediction without the input of new information. We speculated that the introduction of protein 3D structure information into the structure-free model might evade this dilemma.

To test this hypothesis, we introduce four pieces of information into a deep learning neural network model: the distances between atoms of proteins in the 3D space, the corresponding torsion angles, the residue sequences of the protein and compound features. Specifically, an end-to-end model was employed, and the general workflow of the model is depicted in Fig. 1. First, a protein extractor module converts the residue sequence of the protein, distance matrix and torsion matrix into two variables: a new matrix representing the residue sequence of the protein and a new distance matrix encoded with the torsion angles' information. The two outputs are then fed into an evolutionary attention module to learn which part of the matrix should be focused on. Meanwhile, the compound information goes through the compound extractor module, which consists of a multi-head vertex representation and a deep graph convolutional neural network. Notably, an auxiliary module is introduced into the multi-head vertex representation function to prevent the over-smoothing problem during the graph neural network training process[19]. To allow the remote amino acids to communicate with a certain node, a master node is employed to simultaneously capture both local and global features. Finally, the learned representation of compound features and protein features are input to the affinity prediction module, where the predicted affinity between the compound and protein is generated by a pipeline made of a trans-attention block and an attention pooling block. The detailed designs of the compound extractor module and protein extractor module are described in the following sections.

Compound extractor module

Deep GCN and multi-head attention representation are illustrated in the compound extractor module (Fig. 2a). In the graph network, the compound information is extracted using graph representation, in which the main nodes (each atom in the compound) and the master node (a node sum of all-atom in the compound) are employed to aggregate the local and global information of the compound, respectively. Deep graph convolution unit (Fig. 2b) and multi-head attention representation strategy are used to update the main node information, in the gate warp strategy interactively regulates and updates the information between main nodes and the master node. The gated recurrent unit (GRU) is responsible for aggregating the multi-layer information in the compound extractor module and updating the features of the main nodes and the master node.

In GCN, the message passing unit gathers the information of a node's neighbors and passes it to that node for local feature updating purposes (Fig. 2b, Supplementary Fig. 7). Here, we apply a master node to maintain the global feature for nodes over long distances. This helps mitigate the over-smoothing problem in GCN (detailed explanation for the over-smoothing problem, please refer to Supplementary Note 1.2). Moreover, through applying the master node, the number of layers in GCN can be deepened to extract better features of the compounds, thus contributing to the multi-head attention representation (Supplementary Fig. 8). Finally, the local information and global information representation of compounds are jointly input into the affinity prediction module with the protein features learned from the protein extractor module, which benefits the CPAs prediction in the end.

Protein extractor module

Traditional networks could only update one type of data source at a time, while the multimodality mechanism could learn more comprehensive information from a variety of data sources. In contrast to previous work, we innovatively aggregate the protein's sequence and structure features to predict affinity by the protein aggregating unit (Prot-Aggregating). And a

mechanism employed by the evolutionary attention module (Evo-Attention) can update these two properties mutually. The two units jointly construct the backbone of the protein extractor module (Fig. 3a). The discrete distance matrix (DDM) updates sequence features by summing over its columns (Supplementary Fig. 5). The torsion matrix is aggregated into sequence features through the operation of linear mapping and Hadamard product in the protein aggregating unit.

Message communication from the evolving discrete distance matrix to sequence features in the Evo-attention unit (Supplementary Fig. 6) is enabled by an enormous amount of matrix multiplication, which serves as the core of the protein encoder module (Fig. 3b). Embedded distance matrix (Embedded DM) transfers into distance vectors that hold the same shape with sequence features through operations of column sum and row sum. A merging matrix is constructed by multiplying the embedded sequence features with the distance vectors through batch-dot-product operation, which is then added to the features of embedded DM. Sequence features are finally renovated by attention mechanism and gate unit updating methods. Sequence features are then projected to structure information evolutionarily through outer sum operation and gate unit updating method. Such an intricate network architecture satisfies the feature extraction of multimodality pattern, ensuring that the overall Evo-attention unit can fully mix information between features of sequence and structure, and is sufficient for accurate affinity prediction.

Affinity prediction module

Based on an end-to-end architecture, the extracted protein and compound features from the upstream model are fed into the affinity prediction model (Fig. 3c), and they are further extracted by the trans-attention unit and attention pooling method (Supplementary Fig. 10). All the information is fused into a pairwise matrix through outer sum operation to realize the mutual mapping and collaborative update of the information between protein and compound. The trans-attention unit learns the potential interactions of them (Supplementary Fig. 9), and

all these features are flattened by the attention pooling method. Then the global features of the compound are aggregated into the flattened local features. Finally, the prediction of the affinity value is given.

Construction of the Benchmark Dataset and Evaluate Dataset

To test the performance of our model, we first constructed a benchmark dataset based on PDBbind (version 2020, the general set)[20], which is a small dataset containing more than 12,600 compound-protein pairs. Furthermore, we generated an evaluation dataset based on BindingDB (version Feb 6, 2022, the general set)[21], which contains more than 218,600 compound-protein pairs. Both datasets contain a high-quality set of protein-ligand complexes with structural data and corresponding binding affinities. An affinity value of a certain measurement type (i.e., K_i, K_d, or IC_{50}) of each complex was provided. For complexes in the PDBbind dataset, we extracted the pairwise non-covalent interactions information between compounds and proteins by using PLIP[22]. After considering the atom types, distances and bond angles, PLIP recognized seven types of non-covalent interactions, including hydrogen bond, hydrophobic interaction, p-stacking, p-cation, salt bridge, water bridge and halogen bond. Next, we examined the normality of the affinity data. Because extreme data might affect the whole model, we excluded the affinity data beyond the range of $(\mu - 3\sigma, \mu + 3\sigma)$. For both datasets, we use five-fold cross-validation, and the train-valid-test splitting ratio is approximately 7:1:2.

FeatNN outperforms other state-of-the-art models in binding affinities prediction

The performance of seven models trained over the same dataset was compared (Fig. 4). Apart from our model (FeatNN), the baseline methods are BACPI[23], MONN[8] and four variant models (i.e., GATGCN, GCNNet, GATNet and GINConvNet) of GraphDTA[24]. Because some of the compounds/proteins tend to be highly similar and homologous, to avoid information leakage from the test set data in the model training process, we followed the clustering strategy that proposed by previous studies[8,25]. Four different clustering thresholds are used to

split the train-test sets in the control group experiment. They were 0.3, 0.4, 0.5 and 0.6, which indicate the minimum distance between each cluster. For example, a 0.3 clustering threshold means that any compounds/proteins from two different clusters are at least 30% different in their respective structures. In terms of the compound-clustered test group, FeatNN outperformed the second greatest model (MONN) by 21.33% in IC_{50} of the R^2 metrics and 17.07% in KIKD (Fig. 4 and Tab.1). FeatNN also beats the other modules in RMSE and Spearman correlation by a considerable margin (Fig. 4 and Tab.1). As to the protein-clustered test group, FeatNN still topped the group in every threshold, with an average R^2 of 0.552 under IC_{50}, exceeding the second greatest model MONN by 2.79%. Some models could not have consistent performance over both test groups. For example, the MONN model ranked second under the compound-clustered test group in almost all three evaluation metrics. However, it was less competitive with MONN and sometimes less competitive to GATGCN in the protein-clustered test group. Remarkably, our model constantly outperformed other models in all aspects of this experiment (Fig. 4 and Tab.1). This result is consistent even after averaging the performance of different clustering thresholds (Tab. 1).

The prediction of FeatNN is validated by BingDB-derived Dataset

Since compound-protein complexes generally have limited structural availability, we further tested our model on a large-scale compound-protein interaction dataset. FeatNN and MONN have been tested on the large dataset with more than 218,600 compound-protein pairs, both modules were evaluated with more than 153,000 training samples and more than 43,700 test samples [21]. To make a fair comparison, we evaluated binding affinity prediction by averaging predictions from several single models of FeatNN, GraphDTA and BACPI on this BindingDB dataset. In this study, we did not specifically optimize the hyper-parameters of FeatNN and other control models over the BindingDB dataset, and directly used the hyper-parameter settings derived previously from the PDBbind-derived benchmark dataset. And, the performance of FeatNN and MONN were evaluated in terms of RMSE, R^2 and Pearson correlation, as listed in Tab. 2. When evaluating the single models, FeatNN achieved the best

RMSE correlation (0.765), best Pearson correlation (0.850) and best R^2 correlation (0.719) with 5-fold cross-validation on the benchmark dataset. These comparison results suggested that FeatNN can achieve better performance than the state-of-the-art baseline methods by the application of protein 3D structure information for predicting CPAs.

Discussion

In our model, introducing the protein 3D structure information is the key for significantly improving the accuracy in CPA prediction. Structure-free approaches such as BACPI and DeepAffinity utilized only 1d-CNN for protein representation learning, thus inherently suffering from the incapability of capturing the correlation between connected amino acid atoms and long-distance relationships. In contrast, in our protein extractor module, the comprehensive features (e.g., residue sequence of the protein, the torsion angles and the between-atom distance) are fused to be learnt with an interactive attention module, which can make the model aware of where to focus on. This information implicitly comprises the pose of protein when ligands are formed between the compound and protein. In this context, the much more comprehensive protein representations lead to a better model.

Although theoretically appealing to introduce protein 3D structure information, we overcame numerous obstacles in the development of FeatNN. First, although graph representation could effectively represent compound properties, increasing the number of model layers is difficult due to the over-smoothing problem (during the update, connected nodes tend to be very similar, so a model could not distinguish different classes). We elegantly overcame this limitation by introducing a residual connection structure, in which the input was directly added into the output of specific layers in the network[26,27]. Moreover, a master node is jointly employed to learn both local and global features during the training process, thus the over-smoothing problem was further mitigated.

Second, the Euclidean distance information between protein residues in the traditional distance matrix could not be discriminated in a small scope. In this study, the protein distance matrix is discretely coded, and these continuous values are divided into 40 mapping intervals that conform to a normal distribution in statistics. Between 3.25 Å and 50.75 Å, the distance matrix is mapped to 38 intervals with equal distance and width (step of each interval is 1.25 Å). Two additional intervals are added to store any larger distance (when the distance between residues is greater than 50.75 Å) and smaller distance (when the distance between residues is less than 3.25 Å). Meanwhile, the sequence information and torsion angle information of protein are introduced to the protein extractor module, and the discrete distance matrix and protein residue sequence information are further characterized.

Nevertheless, 3D structural information not only exists in the protein but the compound also does. In this study, we only introduced the protein structural information, and experiments to introduce the compound geometry information can be conducted in the future. We believe that the strategy developed for protein feature extracting in our model could also be utilized to extract the geometry information of the compound. The graph representation of the compounds was not utilized in this study, because the aforementioned strategy is too computationally intensive. Nonetheless, only applying the protein feature extractor in this study has already achieved a remarkable result. We will try to implement both protein and compound extractors in our future experiments to test this hypothesis.

In addition, other protein properties like the residue type of the binding ligand, secondary structure and physicochemical characteristics are also likely very important features. Introducing these features into our model might further improve its performance. The challenge is how to convert these features to a computer recognizable format, which is left to be tackled in our future study.

Conclusion

The proposed FeatNN model introduces the torsion matrix and distance matrix into the protein extractor module, and a deep GCN with a master node in the compound extractor module to predict the affinity of a given pair of compound and protein. As the 3D structure information of most proteins is available now, FeatNN could provide a powerful tool for virtual drug screening. More importantly, FeatNN outperforms the benchmark models by a significant margin, which demonstrates great potential in reducing the considerable time and expense in drug candidate screening experiments. Our team will be employing this model to discover new drugs in real cell and animal tests for validation in future studies.

Methods

Full algorithm details

Pseudocodes for each module are available in the supplementary method.

Notation for the operations between vectors and matrix

Definition of operation and variables are listed as follows. We use $\otimes$ for outer product, $\oplus$ for outer sum, $\odot$ for element-wise product, namely, the Hadamard product, $\sigma(\cdot)$ for sigmoid activation function $\sigma(x) = 1/(1+e^{-x})$, $tanh(\cdot)$ for tanh activation function $tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$, $f(\cdot)$ for the GELU activation function $GELU(x) = 0.5x(1 + erf(\frac{x}{\sqrt{2}}))$, where $erf(\cdot)$ serves as the Gauss Error Function, $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$, $softmax(x_i)$ for the soft-max function $exp(x_i)/\sum_i exp(x_i)$.

1. Compound Extractor Module

In this study, graph representation of compound is utilized to descript the specific correlation between atom features and bond features. Given graph representation $\{V, E\}$, namely, vertex and edge are used to represent atom and bond features in compound respectively. To be more specific,

$\{V\} = \{\text{"element name", "aromatic type", "vertex degree", "atom valence"}\}$ in which features are encoded by one-hot-encoding strategy then concatenate into an all-one vector as $\{F_i^{atom} \in R^{\mathrm{h}}\}_{i=1}^{N_a}$ for each atom. Similarly, $\{E\} = \{\text{"bond type", "shape"}\}$ are also applied, obtaining embedded bond features' vector as $\{F_j^{bond} \in R^{\mathrm{h}}\}_{j=1}^{N_b}$, where $i = 1,2,\ldots,N_a$, $j = 1,2,\ldots,N_b$, h stands for dimension of hidden size, N_a is number of compound's atom, N_b is number of protein's residue. Original atom features are defined as $F^0 \in R^{N_a \times \mathrm{h}}$, master node features are defined as the summary of atom features, that is, $F^{master} = \sum_{i=1}^{N_a} F_i^{atom}$. Considering there're l_c graph convolution layers where $l_c = 1,2,\ldots,l_{comp}$ and k_c attention heads where $k_c = 1,2,\ldots,k_{comp}$, l_{comp} is the total number of graph convolution's layer, k_{comp} is the total number of compound features' attention head. Atom features, bond features, and master features in $l_c th$ layer are defined as $F_{atom}^{l_c}$, $F_{bond}^{l_c}$ and $F_{master}^{l_c}$, variables V with k_c

heads are defined as V^{k_c}, for example, $F_{atom}^{l_c,k_c}$ means atom features in $l_c th$ layer of graph convolution neural network with head number of k_c.

1.1 Multi-head Attention Block

Main vertex (atom) features are obtained with a multi-head attention mechanism and element-wise product. Main vertex features are updated as $v_{comain}^{l_c}$ in each layer:

$$v_{tmain}^{l_c,k_c} = softmax(W_{vm}^{l_c,k_c}(tanh(W_{vmain}^{l_c,k_c}F_{atom}^{l_c-1}) \odot F_{master}^{l_c-1})) \otimes W_{ms}^{l_c,k_c}F_{atom}^{l_c-1}$$

$$v_{main}^{l_c} = tanh(W_{cmat}^{l_c}[v_{tmain}^{l_c,k_c}]_{k_c})$$

$$v_{comain}^{l_c} = dropout(F_{atom}^{l_c-1})$$

where $W_{vmain}^{l_c,k_c} \in R^{\mathrm{h}\times\mathrm{h}}$, $W_{vm}^{l_c,k_c} \in R^{\mathrm{h}\times\mathrm{h}}$, $W_{ms}^{l_c,k_c} \in R^{\mathrm{h}\times\mathrm{h}}$, $[\cdot]_{k_c}$ indicates integrates the information from multi-head attention.

1.2 Graph Convolution Neural Networks

The atom features are sequentially updated using message passing unit and graph warp unit at each iteration of graph convolution neural networks.

$$mt_{main}^{l_c} = W_{lu2}^{l_c}[v_{comain}^{l_c}, \sum_{v_k \in Neighbor(v_i)} f(W_{ln}^{l_c}[v_{comain}^{l_c}, F_{bond}^{l_c}]_{h+h+bn})]_{h+h}$$

where bn serves as the shape or size of bond neighbors, $[\cdot,\cdot]_m$ indicates the concatenate operation on different sizes of dimension, $W_{lu2}^{l_c} \in R^{2\mathrm{h}\times\mathrm{h}}$, $W_{ln}^{l_c} \in R^{\mathrm{h}+(h+bn)}$.

To avoid the over-smoothing problem in the graph convolution process, we use initial vertex features F^0 as identity information and residual connection pathway:

$$r^{l_c} = (1-\alpha)mt_{main}^{l_c} + \alpha F^0$$

$$v_{comp}^{l_c} = (\theta W_{fu}^{l_c} r^{l_c} + (1-\theta)r^{l_c}) + v_{comain}^{l_c}$$

where $W_{fu}^{l_c} \in R^{\mathrm{h}\times\mathrm{h}}$, both α and θ are the hyperparameters.

Next, define $ht_{master}^{l_c} = W_{mas}^{l_c}F_{master}^{l_c-1}$, $ht_{mas2m}^{l_c} = W_{mas2m}^{l_c}F_{master}^{l_c-1}$, both main vertex and master node features can be updated mutually through graph warp unit and GRU layers, $GRUs$ are used to determine the proportion of main vertex and mater node features updated at layer l_c.

$$g_{main}^{l_c} = \sigma(W_{zm1}^{l_c} v_{comp}^{l_c} + W_{zm2}^{l_c} ht_{mas2m}^{l_c})$$

$$ht_{main}^{l_c} = g_{main}^{l_c} ht_{mas2m}^{l_c} + (1 - g_{main}^{l_c}) v_{comp}^{l_c}$$

$$v_{comain}^{l_c} = GRU_{main}(ht_{main}^{l_c}, v_{comain}^{l_c-1})$$

With the same process, master node features also updated as $v_{comaster}^{l_c}$ in each graph convolution layer, that is,

$$g_{master}^{l_c} = \sigma(W_{zs1}^{l_c} ht_{master}^{l_c} + W_{zs2}^{l_c} v_{main}^{l_c})$$

$$ht_{master}^{l_c} = g_{master}^{l_c} v_{main}^{l_c} + (1 - g_{master}^{l_c}) ht_{master}^{l_c}$$

$$v_{comaster}^{l_c} = GRU_{master}(ht_{master}^{l_c}, v_{comaster}^{l_c-1})$$

where $W_{mas2m}^{l_c} \in R^{\mathrm{h}\times\mathrm{h}}$, $W_{mas}^{l_c} \in R^{\mathrm{h}\times\mathrm{h}}$, $W_{zs1}^{l_c} \in R^{\mathrm{h}\times\mathrm{h}}$, $W_{zs2}^{l_c} \in R^{\mathrm{h}\times\mathrm{h}}$.

After the iterations of the graph convolutional network, final features of main vertex and mater features are given as $v_{comain}^{l_{comp}}$ and $v_{comaster}^{l_{comp}}$, defined as F^{fatom} and $F^{fmaster}$ in infra.

2. Protein Extractor Module

2.1 Protein Embedding Module

Sequence and distance features are embedded through word embedding strategy, torsion features are embedded through a linear layer as well. Protein embedding module takes sequence features $\{F_n^{seq} \in R^{pe}\}_{n=1}^{N_{res}}$, discrete distance matrix $\{\boldsymbol{F}_n^{DDM} \in R^h\}_{n=1}^{N_{res}\times N_{res}}$ and torsion matrix $\{\boldsymbol{F}_n^{TM} \in R^h\}_{n=1}^{N_{res}}$ of protein as input data. A linear layer is used to update these features, that is,

$$l_n^{seq} = W_{tc} F_n^{seq}$$

$$\boldsymbol{E}_{i,j}^{DDM} = W_{td} F_n^{DDM}$$

$$l_n^{TM} = W_{tm}(f(W_{te}(\boldsymbol{F}_n^{TM})))$$

where N_{res} serves as the length of amino acid in each protein, N_{ts} stands for the initial dimension size of torsion size, h stands for hidden size and e for embed size, pe stands for pre embed size, $W_{tc} \in R^{pe\times\mathrm{h}}$, $W_{td} \in R^{h\times\mathrm{e}}$, $W_{te} \in R^{N_{ts}\times h}$, $W_{tm} \in R^{h\times h}$.

Aggregating protein sequence, distance and torsion features together is a novel strategy before extracting protein features.

$$\boldsymbol{E}_n^{seq} = (((l_n^{seq} + \sum_{i=1}^{N_{res}} F_n^{DDM}) \odot l_n^{seq}) + \sum_{i=1}^{N_{res}} \boldsymbol{F}_n^{DDM}) \odot l_n^{TM}$$

Ultimately, embedded sequence vector $\{E_n^{seq} \in R^{\mathrm{h}}\}_{n=1}^{N_{res}}$ and embedded discrete distance matrix $\{\boldsymbol{E}_{i,j}^{DDM} \in R^{\mathrm{e}}\}_{i=1,j=1}^{N_{res}\times N_{res}}$ are obtained from protein embedding block.

2.2 Evo-attention Module

We use an evolutional updating strategy to update sequence and structure features in the evo-attention model block by combining information from the protein embedding block. Sequence features $\{E_n^{seq} \in R^{\mathrm{h}}\}_{n=1}^{N_{res}}$ and structure features $\{\boldsymbol{E}_{i,j}^{DDM} \in R^{\mathrm{e}}\}_{i=1,j=1}^{N_{res}\times N_{res}}$ are embedded from protein embedding algorithm, i, j served as the row and column of the embedded discrete distance matrix, respectively. We define the input features in the layer l_p as $F_{seq}^{l_p-1}$ and $\boldsymbol{DDM}_{i,j}^{l_p-1}$, where $l_p = 1,2,\ldots,L_p$.

We use gate mechanism to gather more useful information from input features, eg. h for the hidden size, e for embed size,k for the number of attention-head and k_h for the hidden size of attention-head where $k_h = h/k$.

:

$$g_{seq}^{l_p} = \sigma(W_{h\to h} F_{seq}^{l_p-1})$$

$$g_{pair}^{l_p} = \sigma(W_{e\to k} \boldsymbol{DDM}_{i,j}^{l_p-1})$$

Protein features are processed with concatenating row and column pooling features of $\boldsymbol{DDM}_{i,j}^{l_p-1}$, gathering these features with $W_{gather}^{l_p} \in \mathrm{R}^{2\mathrm{e}\times\mathrm{h}}$ and $W_{query}^{l_p} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$. We use the outer product to update and map informations from input features, use multi-head attention to learn the diversified correlation of $\boldsymbol{DDM}_{i,j}^{l_p-1}$:

$$s_{att}^{k,l_p} = trans_{h\to k,k_h}(W_{gather}^{l_p}[\sum_{i=1}^{n} \boldsymbol{DDM}_{i,j}^{l_p-1}, \sum_{j=1}^{n} \boldsymbol{DDM}_{i,j}^{l_p-1}]_{h+h}) \otimes trans_{h\to k,k_h}(W_{query}^{l_p} F_{seq}^{l_p-1})$$

$$\alpha_{weight}^{k,l_p} = softmax(s_{att}^{k,l_p} + W_{att}^{k,l_p} \boldsymbol{DDM}_{i,j}^{l_p-1})$$

where $[\cdot,\cdot]_m$ indicates concatenate operation between row and column features of $\boldsymbol{DDM}_{i,j}^{l_p-1}$, $trans_{h\to k,k_h}(\cdot)$ indicates the operation of splitting vectors from h to (k,k_h), and $W_{att}^{k,l_p} \in \mathrm{R}^{\mathrm{e}\times\mathrm{k}}$. We aggregate features with the outer product (create a symmetric matrix with a high correlation of discrete distance matrix) and gate. The updated features of sequence and DDM in $l_p th$ layer evo-attention are given as $F_{seq}^{l_p}$, $\boldsymbol{DDM}_{i,j}^{l_p}$, that is,

$$F_{seq}^{l_p} = W_{seq}^{l_p}(g_{seq}^{l_p} \odot (\alpha_{weight}^{k,l_p} \otimes W_{value}^{k,l_p} F_{seq}^{l_p-1}) + (1 - g_{seq}^{l_p}) \odot f(W_{nlin}^{k,l_p} F_{seq}^{l_p-1}))$$

$$\boldsymbol{DDM}^{l_p} = W_{DDM}^{l_p}(g_{pair}^{l_p} \odot (f(W_{s2p}^{l_p} F_{seq}^{l_p-1}) \oplus f(W_{s2p}^{l_p} F_{seq}^{l_p-1})) + (1 - g_{pair}^{l_p}) \odot \alpha_{weight}^{k,l_p})$$

where $W_{s2p}^{l_p} \in \mathrm{R}^{\mathrm{h}\times\mathrm{k}}$, $W_{value}^{k,l_p} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$, $W_{nlin}^{k,l_p} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$, $W_{seq}^{l_p} \in \mathrm{R}^{\mathrm{k}\times k_h}$, $W_{DDM}^{l_p} \in \mathrm{R}^{\mathrm{k}\times\mathrm{e}}$.

Finally, to elevate the convergence rate and generalization, we apply the layer normalization and dropout operation on the output sequence features of the evo-attention block, that is,

$$F_{new-seq}^{l_p} = layernorm(Dropout(F_{seq}^{l_p-1}) + F_{seq}^{l_p-1})$$

where $laynorm(x)$ is the layer norm operation $\frac{x-E[x]}{\sqrt{Var[x]+\epsilon}}\gamma + \beta$, as γ and β serves as the learnable parameters, and $\epsilon = 10^{-5}$ to avoid the denominator being zero.

After calculating L_p iterations of protein encoder, we obtain the final features representation $\{F_{seq,i}^{l_p} \in \mathrm{R}^{\mathrm{h}}\}_{i=1}^{N_{res}}$ and $\{\boldsymbol{DDM}_{i,j}^{l_p} \in \mathrm{R}^{\mathrm{e}}\}_{i=1,j=1}^{N_{res}\times N_{res}}$.

3. Affinity Prediction Module

The affinity prediction module integrates mutual information between compound and protein in non-covalent interaction affinity prediction. Given atom features $\{F_i^{fatom} \in R^{\mathrm{h}}\}_{i=1}^{N_a}$ and master node features $\{F^{fmaster} \in R^{\mathrm{h}}\}$ from the compound extractor, as well as protein features $\{F_i^{seq} \in \mathrm{R}^{\mathrm{h}}\}_{i=1}^{N_{res}}$ extracted from protein extractor. Considering there are l_{aff} layers in mutual transform attention blocks, using tokens $F_{atom}^{l_A}$ and $F_{seq}^{l_A}$ stand for the lth layer's features of compound and protein where $l_A = 1,2,\dots,l_{aff}$. Particularly, the first layer of both

compound and protein features are separately transformed into a compatible space by a single-linear-layer, that is, $F_{atom}^{1} = f(W_{atom}F_{i}^{fatom})$ and $F_{seq}^{1} = f(W_{res}F_{i}^{seq})$, where $i = 1,2,..,N_a$, $j = 1,2,..,N_{res}$, $W_{atom} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$, $W_{res} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$.

To elevate the diversification of these features, a multi-head attention mechanism is also applied in this block, that is,

$$v_{comp}^{l_A,k_A} = softmax(W_{\mathrm{kcom}}^{l_A}F_{atom}^{l_A-1})trans_{h\to k,k_h}(F_{atom}^{l_A-1})$$

$$v_{prot}^{l_A,k_A} = softmax(W_{\mathrm{kpro}}^{l_A}F_{seq}^{l_A-1})trans_{h\to k,k_h}(F_{seq}^{l_A-1})$$

where $k_A = 1,2,\dots N_{head}$, indicates the number of multi-head features. $W_{\mathrm{kcom}}^{l_A} \in \mathrm{R}^{\mathrm{h}\times k_A}$, $W_{\mathrm{kpro}}^{l_A} \in \mathrm{R}^{\mathrm{h}\times k_A}$.

An interactional information matrix $\boldsymbol{S}_{mut}^{l_A} \in \mathrm{R}^{N_{atom}\times N_{res}\times\mathrm{h}}$ is calculated through outer sum operation between features of compound and protein.

$$\boldsymbol{S}_{mut}^{l_A} = softmax\left(F_{atom}^{l_A-1} \oplus F_{seq}^{l_A-1}\right)$$

$$F_{atom}^{l_A} = Dropout(tanh(W_{\mathrm{mc}}^{l_A}v_{comp}^{l_A,k_A}) \odot \boldsymbol{S}_{mut}^{l_A}(W_{p2c}^{l_A}F_{seq}^{l_A-1})) + F_{atom}^{l_A-1}$$

$$F_{seq}^{l_A} = Dropout(tanh(W_{mp}^{l_A}v_{prot}^{l_A,k_A}) \odot \boldsymbol{S}_{mut}^{l_A}(W_{c2p}^{l_A}F_{atom}^{l_A-1})) + F_{seq}^{l_A-1}$$

where $W_{\mathrm{mc}}^{l_A} \in \mathrm{R}^{k_A\times\mathrm{h}}$, $W_{\mathrm{mp}}^{l_A} \in \mathrm{R}^{k_A\times\mathrm{h}}$, $W_{p2c}^{l_A} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$, $W_{c2p}^{l_A} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$.

Protein and compound features are eventually calculated after the $l_{aff}th$ iteration. Before affinity prediction, feature aggregating operation between master node features and main graph features should be considered with the help of attention pooling, that is,

$$C_{final} = softmax(W_{cfm}tanh(W_{mam}F_{atom}^{l_{aff}}))F_{atom}^{l_{aff}}$$

$$C_{aggre} = [C_{final}, F^{fmaster}]_{\mathrm{h+h}}$$

Same operations also happened to protein features, that is,

$$P_{final} = softmax(W_{pfm}tanh(W_{fpro}F_{seq}^{l_{aff}}))F_{seq}^{l_{aff}}$$

where $[\cdot,\cdot]_m$ indicates concatenate operation between hidden size of main vertex features and master node features, $W_{mam} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$, $W_{fpro} \in \mathrm{R}^{\mathrm{h}\times\mathrm{h}}$, $W_{pfm} \in \mathrm{R}^{\mathrm{h}\times 1}$, $W_{cfm} \in \mathrm{R}^{\mathrm{h}\times 1}$,

particularly, the main function of W_{pfm} and W_{cfm} is to flatten features between both compound and protein.

Finally, with a single linear mapping layer, the affinity value is calculated by vectors of C_{aggre} and P_{final}, that is,

$$affinity = W_{aff}(f(C_{aggre}P_{final}))$$

where $W_{aff} \in \mathrm{R}^{2\mathrm{h}^2 \times 1}$.

QUANTIFICATION AND STATISTICAL ANALYSIS

Evaluation Metrics

We use eight metrics that were commonly used on this problem to evaluate the prediction performance of our model, both the eight metrics are defined as following :

R^2 Score, RMSE (Root Mean Square Error), MAE (Mean Absolute Error) are often used in the regression model. They describe the distance between predicted values and true values. RMSE is the standardized value of MSE, which is usually used as training loss in machine learning studies. It is defined as:

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)}$$

The R^2 score is a dimensionless score describing the effectiveness of the model. It compares the prediction to random guess according to the average of true values:

$$\begin{aligned} R^2(y, \hat{y}) &= 1 - \frac{SS_{residual}}{SS_{total}} \\ &= 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y_i})^2} \end{aligned}$$

MAE describes the absolute error of predict values and true values:

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MedAE is the median form of MAE, it describes the median value of the residues:

$$MedAE(y, \hat{y}) = Med_n(|y_i - \hat{y}_i|)$$

We use two coefficients that describe the correlation between the predicted values and true values: Pearson product-moment correlation coefficient and Spearman's rank correlation coefficient.

The Pearson correlation coefficient describes the linear correlation between two values, it is defined as:

$$\begin{aligned} Pearson(y, \hat{y}) &= \frac{Cov(y, \hat{y})}{\sigma_y \sigma_{\hat{y}}} \\ &= \frac{\sum_i (y_i - \overline{y})\left(\hat{y}_i - \overline{\hat{y}_i}\right)}{\sqrt{\sum_i (y_i - \overline{y_i})^2}\sqrt{\sum_i \left(\hat{y}_i - \overline{\hat{y}_i}\right)^2}} \end{aligned}$$

The Spearman's rank correlation coefficient is the rank form of the Pearson correlation coefficient. It describes the dependencies of two variables (e.g. as one increases the other also increases), it is related to function monotonicity:

$$Spearman(y, \hat{y}) = 1 - \frac{6\sum_i d_i^2}{n(n^2 - 1)}$$

which d_i is the rank difference between y_i and $\hat{y}_i$.

Acknowledgments

This study was supported by the Scientific and Technological Innovation 2030 Program of China - major projects (2021ZD0200408 to X.W.), the Natural Science Foundation of Zhejiang Province (LR20H090002 to X.W.), the National Natural Science Foundation of China (81971866 to X.W.), the Leading Innovative and Entrepreneur Team Introduction Program of Zhejiang (2019R01007 to X.W.), Fundamental Research Funds for the Central Universities (K20210195 to X.W.).

Conflict of Interest

Zhejiang University has filed a patent application related to this work, with X.W., B.G., H.Z., and H.J. listed as inventors.

Author Contributions

B.G., H.Z. and H.J. contributed equally to this work. X.W., B.G. and H.Z. conceptualized and designed the study. B.G., H.Z., and H.J. conducted the experiments and collected the data. B.G., H.Z., H.J., X.W., H.Y., X.L., N.G. and Y. Z. analyzed and interpreted the data. B.G., H.Z., and X.W. drafted the paper. All authors critically revised the manuscript and approved the final version for submission.

Data Availability

The data to support the findings of this study are included in the paper, and further data are available from the corresponding author upon reasonable request.

Code Availability

https://github.com/StarLight1212/FeatNN.

References

1 Chen, X. *et al.* Drug-target interaction prediction: databases, web servers and computational models. *Brief Bioinform* **17**, 696-712, doi:10.1093/bib/bbv066 (2016).

2 Rester, U. From virtuality to reality - Virtual screening in lead discovery and lead optimization: a medicinal chemistry perspective. *Curr Opin Drug Discov Devel* **11**, 559-568 (2008).

3 Kim, S. *et al.* PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res* **49**, D1388-D1395, doi:10.1093/nar/gkaa971 (2021).

4 Ru, X., Ye, X., Sakurai, T. & Zou, Q. NerLTR-DTA: Drug-target binding affinity prediction based on neighbor relationship and learning to rank. *Bioinformatics*, doi:10.1093/bioinformatics/btac048 (2022).

5 Bleakley, K. & Yamanishi, Y. Supervised prediction of drug-target interactions using bipartite local models. *Bioinformatics* **25**, 2397-2403, doi:10.1093/bioinformatics/btp433 (2009).

6 Cao, D. S. *et al.* Computational Prediction of DrugTarget Interactions Using Chemical, Biological, and Network Features. *Mol Inform* **33**, 669-681, doi:10.1002/minf.201400009 (2014).

7 Ozturk, H., Ozkirimli, E. & Ozgur, A. A comparative study of SMILES-based compound similarity functions for drug-target interaction prediction. *BMC Bioinformatics* **17**, 128, doi:10.1186/s12859-016-0977-x (2016).

8 Li, S. *et al.* MONN: A Multi-objective Neural Network for Predicting Compound-Protein Interactions and Affinities. *Cell Systems* **10**, 308-322.e311, doi:https://doi.org/10.1016/j.cels.2020.03.002 (2020).

9 Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J. & Koes, D. R. Protein-Ligand Scoring with Convolutional Neural Networks. *J Chem Inf Model* **57**, 942-957, doi:10.1021/acs.jcim.6b00740 (2017).

10 Lee, I., Keum, J. & Nam, H. DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Comput Biol* **15**, e1007129, doi:10.1371/journal.pcbi.1007129 (2019).

11 Rifaioglu, A. S. *et al.* DEEPScreen: high performance drug-target interaction prediction with convolutional neural networks using 2-D structural compound representations. *Chem Sci* **11**, 2531-2557, doi:10.1039/c9sc03414e (2020).

12 Gao, H. *et al.* Study of the Structural Dynamics of the E. coli 70S Ribosome Using Real-Space Refinement. *Cell* **113**, 789-801, doi:10.1016/s0092-8674(03)00427-6 (2003).

13 Ballester, P. J. & Mitchell, J. B. A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics* **26**, 1169-1175, doi:10.1093/bioinformatics/btq112 (2010).

14 Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W. & Kanehisa, M. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* **24**, i232-240, doi:10.1093/bioinformatics/btn162 (2008).

15 Fang, X. *et al.* Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence* **4**, 127-134, doi:10.1038/s42256-021-00438-4 (2022).

16 Jones, D. *et al.* Improved Protein-Ligand Binding Affinity Prediction with Structure-Based Deep Fusion Inference. *J Chem Inf Model* **61**, 1583-1592, doi:10.1021/acs.jcim.0c01306 (2021).

17 Kipf, T. N. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* **abs/1609.02907** (2016).

18 Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583-589, doi:10.1038/s41586-021-03819-2 (2021).

19 Ishiguro, K., Maeda, S.-I. & Koyama, M. Graph Warp Module: an Auxiliary Module for Boosting the Power of Graph Neural Networks in Molecular Graph Analysis. doi:10.48550/ARXIV.1902.01020 (2019).

20 Wang, R., Fang, X., Lu, Y., Yang, C. Y. & Wang, S. The PDBbind Database:? Methodologies and Updates. *Journal of Medicinal Chemistry* **48**, 4111-4119 (2005).

21 Gilson, M. K. *et al.* BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res* **44**, D1045-1053, doi:10.1093/nar/gkv1072 (2016).

22 Salentin, S., Schreiber, S., Haupt, V. J., Adasme, M. F. & Schroeder, M. PLIP: fully automated protein-ligand interaction profiler. *Nucleic Acids Res* **43**, W443-447, doi:10.1093/nar/gkv315 (2015).

23 Öztürk, H., Özgür, A. & Ozkirimli, E. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics* **34**, i821-i829, doi:10.1093/bioinformatics/bty593 (2018).

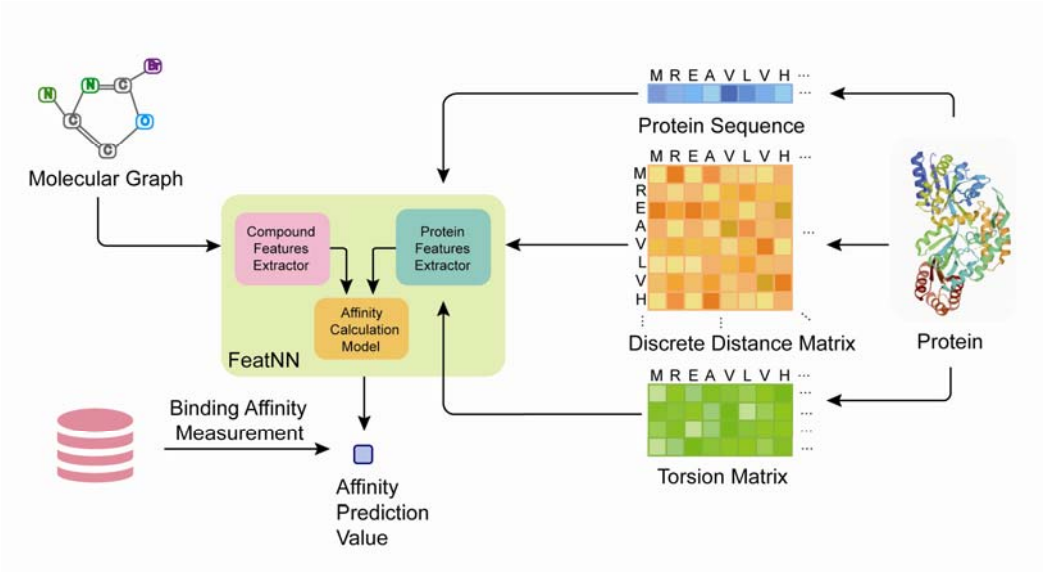
24 Nguyen, T. *et al.* GraphDTA: predicting drug-target binding affinity with graph neural networks. *Bioinformatics* **37**, 1140-1147, doi:10.1093/bioinformatics/btaa921 (2021).

25 Mayr, A. *et al.* Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem Sci* **9**, 5441-5451, doi:10.1039/c8sc00148k (2018).

26 He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *CoRR* **abs/1512.03385** (2015).

27 Chen, M., Wei, Z., Huang, Z., Ding, B. & Li, Y. Simple and Deep Graph Convolutional Networks. *arXiv [cs.LG]* (2020).

Figures



Abstract figure

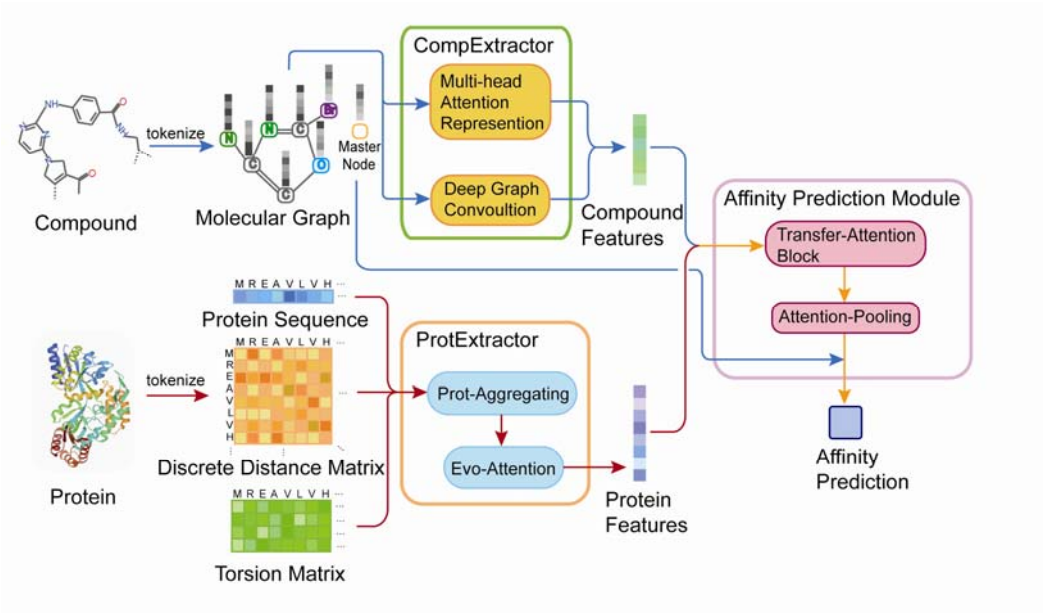


Fig.1 | The architecture overview of FeatNN. Given a compound-protein pair, a deep graph convolution module and multi-head attention blocks are first used to extract the atom features from the input molecular graph, an embedding module and multi evolutional-attention blocks are first used to extract the residue features from the input protein sequence. Then, these extracted atom and residue features are processed by trans-attention blocks, which also enables one to construct the links between atoms of the compound and residues of the protein. Finally, information from atom features, residue features are integrated to predict the binding affinity.

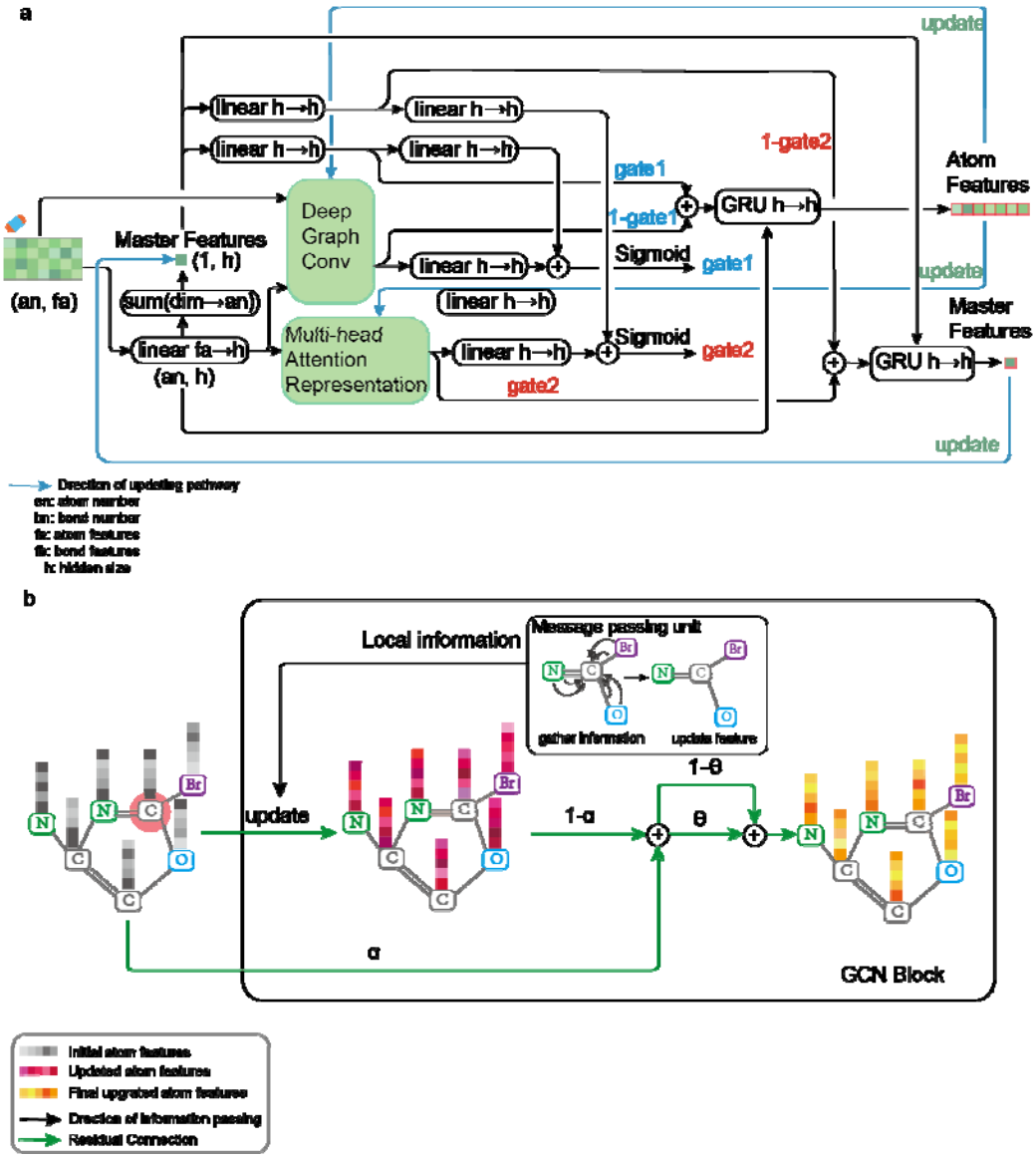


Fig. 2 | The compound extractor module of FeatNN

(A) The compound extractor module to extract atom features from an input compound.

(B) The deep graph convolution module for encoding the molecular features of an input compound.

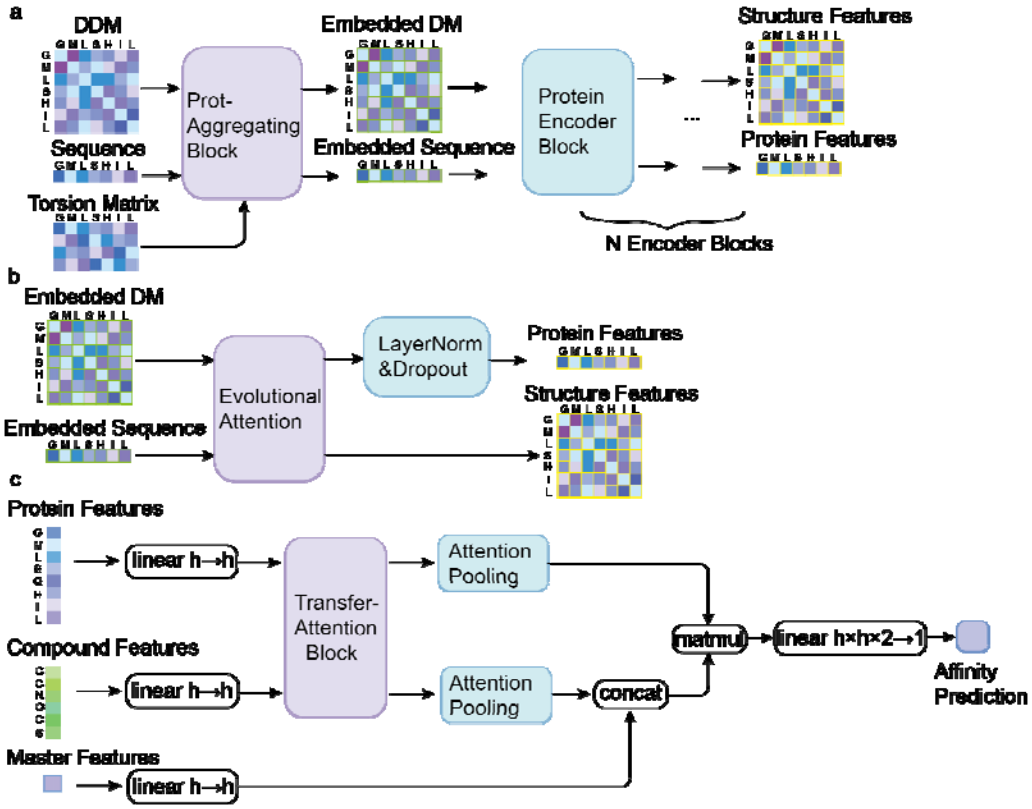


Fig. 3 | The Protein Extractor Module and Prediction Module of FeatNN

(a) The Protein Extractor Module to extract protein features from an input protein.

(b) The Protein Encoder Module from the protein embedding block.

(c) The affinity prediction module.

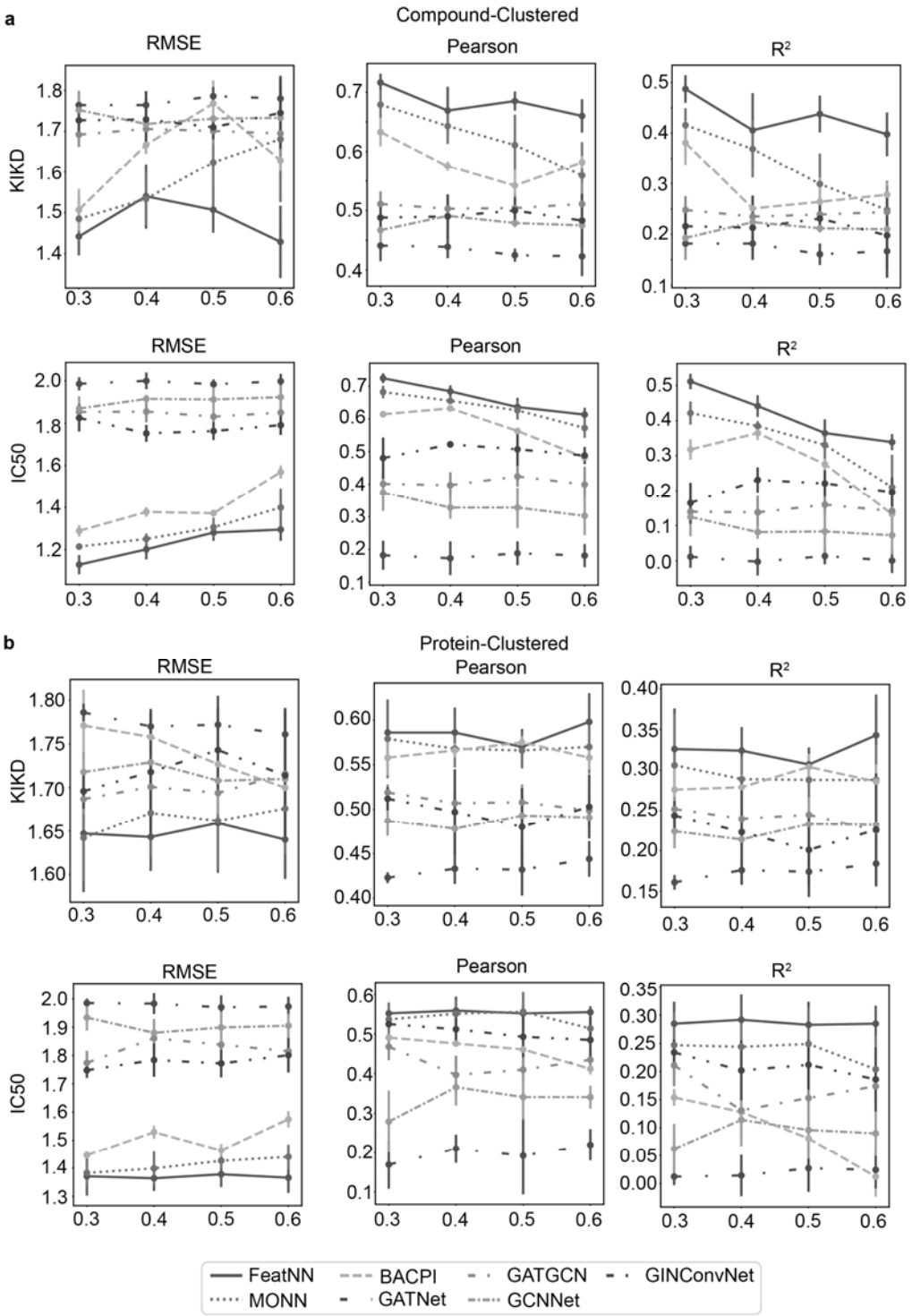


Fig. 4. Evaluation on FeatNN, BACPI, GraphDTA (GATNet, GATGCN, GCNNet, GINConvNet) and MONN on PDBBind-Datasets. (a) was tested on the new compounds with affinity measured by IC_{50} and KIKD respectively. Similarly, (b) was tested on the new protein data.

Our model, FeatNN, outperforms the other models in all three metrics (i.e., RMSE, Pearson and R^2) at all clustering thresholds.

Tables

Tab. 1. Model performance comparisons on PDBBind database with strategy of compound-clustered and protein-clustered. The models are ordered by their performance on the compound-clustered test group in terms of in the R^2 for IC50. FeatNN outperforms the other models by significant margins in all metrics and on both affinity measurements. The closer the R^2 value is to 1 or -1, the better the forecast value approximates the real value.

Type	**threshold**	**Module**	**R^2**		**RMSE**		**Pearson**	
			IC_{50}	**KIKD**	**IC_{50}**	**KIKD**	**IC_{50}**	**KIKD**
Compound-Cluster	0.3	FeatNN	**0.512(0.022)**	**0.487(0.027)**	**1.13(0.045)**	**1.442(0.046)**	**0.724(0.015)**	**0.716(0.015)**
		MONN	0.422(0.033)	0.416(0.033)	1.215(0.015)	1.485(0.054)	0.682(0.019)	0.679(0.024)
		BACPI	0.318(0.029)	0.381(0.043)	1.289(0.027)	1.507(0.051)	0.614(0.01)	0.633(0.024)
		GATNet	0.011(0.031)	0.182(0.032)	1.986(0.031)	1.764(0.034)	0.182(0.044)	0.441(0.026)
		GATGCN	0.139(0.04)	0.248(0.027)	1.853(0.044)	1.692(0.03)	0.401(0.04)	0.511(0.021)
		GCNNet	0.124(0.055)	0.193(0.043)	1.869(0.058)	1.752(0.047)	0.374(0.057)	0.467(0.026)
		GINConvNet	0.164(0.059)	0.216(0.036)	1.825(0.064)	1.727(0.04)	0.48(0.063)	0.488(0.026)
	0.4	FeatNN	**0.442(0.031)**	**0.406(0.073)**	**1.202(0.047)**	**1.54(0.079)**	**0.684(0.018)**	**0.669(0.04)**

	MONN	0.385(0.014)	0.369(0.057)	1.251(0.025)	1.534(0.064)	0.655(0.01)	0.643(0.03)
	BACPI	0.365(0.02)	0.251(0.02)	1.378(0.022)	1.667(0.022)	0.632(0.009)	0.575(0.008)
	GATNet	-0.003(0.039)	0.182(0.016)	2.000(0.039)	1.764(0.017)	0.173(0.051)	0.439(0.019)
	GATGCN	0.137(0.047)	0.235(0.015)	1.855(0.051)	1.706(0.017)	0.397(0.04)	0.503(0.012)
	GCNNet	0.081(0.019)	0.224(0.021)	1.915(0.019)	1.718(0.023)	0.327(0.034)	0.491(0.021)
	GINConvNet	0.231(0.036)	0.213(0.063)	1.752(0.04)	1.729(0.069)	0.522(0.01)	0.490(0.037)
0.5	FeatNN	**0.365(0.039)**	**0.438(0.036)**	**1.281(0.039)**	**1.507(0.056)**	**0.636(0.028)**	**0.685(0.016)**
	MONN	0.331(0.045)	0.299(0.061)	1.306(0.045)	1.624(0.109)	0.626(0.028)	0.611(0.051)
	BACPI	0.276(0.017)	0.264(0.048)	1.372(0.016)	1.768(0.056)	0.563(0.007)	0.542(0.027)
	GATNet	0.013(0.023)	0.161(0.021)	1.984(0.023)	1.786(0.022)	0.188(0.036)	0.425(0.011)
	GATGCN	0.159(0.03)	0.239(0.029)	1.831(0.033)	1.701(0.032)	0.424(0.031)	0.504(0.022)
	GCNNet	0.083(0.06)	0.212(0.008)	1.912(0.062)	1.731(0.009)	0.327(0.062)	0.479(0.007)
	GINConvNet	0.221(0.038)	0.231(0.026)	1.763(0.042)	1.710(0.029)	0.507(0.051)	0.500(0.022)
0.6	FeatNN	**0.339(0.023)**	**0.398(0.043)**	**1.295(0.053)**	**1.429(0.088)**	**0.613(0.021)**	**0.660(0.028)**
	MONN	0.21(0.093)	0.248(0.057)	1.399(0.092)	1.681(0.155)	0.572(0.03)	0.559(0.057)
	BACPI	0.132(0.033)	0.278(0.024)	1.569(0.03)	1.629(0.027)	0.482(0.009)	0.582(0.011)

	GATNet	0(0.035)	0.167(0.034)	1.998(0.035)	1.78(0.036)	0.181(0.035)	0.423(0.033)
	GATGCN	0.141(0.06)	0.244(0.051)	1.85(0.064)	1.695(0.057)	0.399(0.055)	0.511(0.036)
	GCNNet	0.072(0.054)	0.21(0.019)	1.923(0.056)	1.733(0.021)	0.302(0.059)	0.475(0.012)
0.6	GINConvNet	0.196(0.042)	0.198(0.083)	1.791(0.046)	1.745(0.087)	0.488(0.026)	0.483(0.045)

Type	**Threshold**	**Model**	**R^2**		**RMSE**		**Pearson**	
			IC_{50}	**KIKD**	**IC_{50}**	**KIKD**	**IC_{50}**	**KIKD**
Protein-Cluster	0.3	FeatNN	0.285(0.039)	0.326(0.05)	1.371(0.068)	1.647(0.067)	0.552(0.027)	0.586(0.036)
		MONN	0.247(0.058)	0.306(0.063)	1.383(0.046)	1.642(0.049)	0.537(0.042)	0.579(0.044)
		BACPI	0.154(0.015)	0.276(0.034)	1.446(0.013)	1.771(0.041)	0.491(0.009)	0.558(0.02)
		GATNet	0.012(0.015)	0.161(0.009)	1.985(0.016)	1.786(0.01)	0.169(0.061)	0.423(0.006)
		GATGCN	0.211(0.037)	0.252(0.015)	1.774(0.042)	1.687(0.017)	0.468(0.034)	0.519(0.009)
		GCNNet	0.062(0.045)	0.224(0.021)	1.934(0.046)	1.718(0.023)	0.279(0.078)	0.487(0.017)
		GINConvNet	0.234(0.023)	0.244(0.018)	1.748(0.027)	1.696(0.02)	0.525(0.009)	0.512(0.013)
	0.4	FeatNN	0.292(0.045)	0.324(0.029)	1.364(0.045)	1.643(0.039)	0.559(0.035)	0.586(0.028)

		MONN	0.244(0.048)	0.289(0.027)	1.399(0.06)	1.671(0.05)	0.551(0.035)	0.568(0.021)
		BACPI	0.128(0.026)	0.279(0.015)	1.529(0.023)	1.758(0.019)	0.476(0.005)	0.566(0.008)
		GATNet	0.014(0.037)	0.176(0.014)	1.983(0.037)	1.77(0.015)	0.21(0.036)	0.433(0.017)
		GATGCN	0.131(0.056)	0.24(0.018)	1.861(0.06)	1.701(0.02)	0.397(0.048)	0.507(0.013)
		GCNNet	0.114(0.047)	0.214(0.045)	1.88(0.049)	1.729(0.049)	0.366(0.046)	0.478(0.028)
		GINConvNet	0.202(0.053)	0.223(0.065)	1.784(0.059)	1.718(0.072)	0.512(0.029)	0.497(0.048)
	0.5	FeatNN	0.283(0.041)	0.307(0.021)	1.378(0.045)	1.659(0.057)	0.552(0.021)	0.57(0.02)
		MONN	0.249(0.065)	0.288(0.028)	1.426(0.027)	1.662(0.055)	0.556(0.05)	0.566(0.02)
		BACPI	0.081(0.029)	0.304(0.016)	1.464(0.023)	1.727(0.02)	0.461(0.008)	0.575(0.007)
		GATNet	0.027(0.042)	0.174(0.031)	1.97(0.042)	1.772(0.033)	0.192(0.098)	0.432(0.029)
		GATGCN	0.153(0.035)	0.245(0.022)	1.838(0.038)	1.694(0.025)	0.41(0.035)	0.508(0.02)
		GCNNet	0.096(0.052)	0.233(0.033)	1.899(0.055)	1.708(0.037)	0.341(0.057)	0.493(0.035)
		GINConvNet	0.212(0.044)	0.201(0.044)	1.772(0.049)	1.743(0.048)	0.493(0.051)	0.48(0.044)
	0.6	FeatNN	0.285(0.032)	0.343(0.05)	1.366(0.054)	1.64(0.045)	0.555(0.015)	0.598(0.032)
		MONN	0.204(0.059)	0.288(0.043)	1.44(0.045)	1.676(0.053)	0.514(0.03)	0.57(0.03)
		BACPI	0.012(0.036)	0.286(0.021)	1.575(0.029)	1.7(0.025)	0.412(0.014)	0.558(0.013)

		GATNet	0.024(0.033)	0.184(0.023)	1.973(0.034)	1.761(0.025)	0.22(0.04)	0.444(0.02)
		GATGCN	0.174(0.035)	0.225(0.04)	1.815(0.038)	1.716(0.045)	0.435(0.027)	0.499(0.018)
		GCNNet	0.09(0.041)	0.232(0.013)	1.905(0.042)	1.71(0.014)	0.341(0.029)	0.491(0.012)
		GINConvNet	0.186(0.057)	0.226(0.07)	1.801(0.061)	1.714(0.077)	0.485(0.058)	0.503(0.036)

Tab. 2. Performance Evaluation of Different Prediction Approaches on the BindingDB Dataset

	R^2	RMSE	Pearson
FeatNN	**0.719(0.003)**	**0.765(0.004)**	**0.850(0.001)**
MONN	$0.706_{(0.004)}$	$0.783_{(0.005)}$	$0.844_{(0.002)}$
BACPI	0.577(0.005)	0.935(0.006)	0.769(0.002)
GATGCN	0.543(0.015)	0.992(0.016)	0.742(0.012)
GCNNet	0.510(0.023)	1.030(0.023)	0.717(0.015)
GINConvNet	0.451(0.124)	1.080(0.119)	0.669(0.094)
GATConvNet	0.327(0.027)	1.200(0.024)	0.585(0.001)