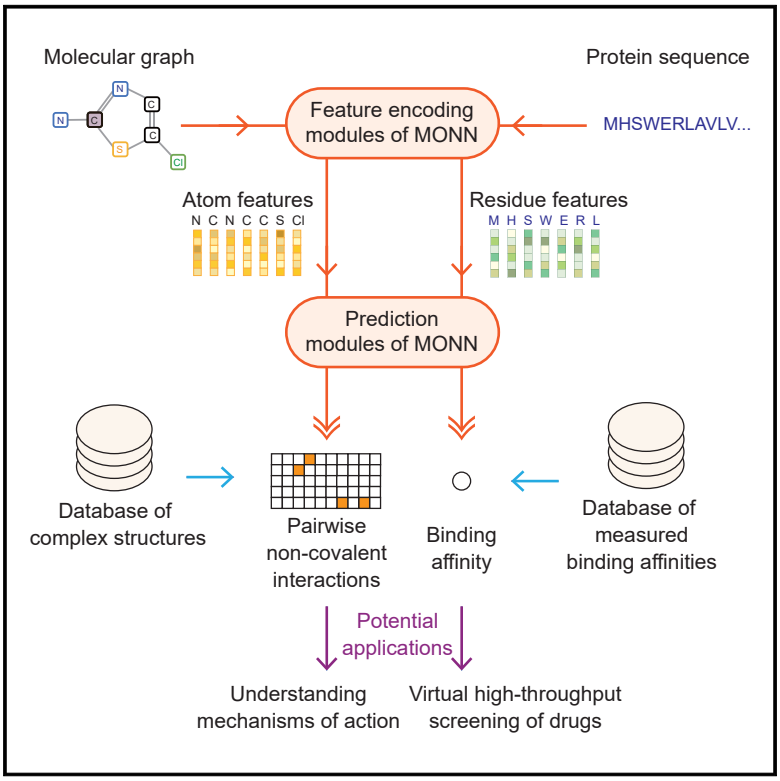
Cell Systems

MONN: A Multi-objective Neural Network for Predicting Compound-Protein Interactions and Affinities

Graphical Abstract



Highlights

- MONN models compound-protein interactions from structure-free information
- MONN predicts both inter-molecular non-covalent interactions and binding affinities
- MONN outperforms other methods, on large datasets with or without atomic structures
- Predictions of MONN can be validated by known chemical rules

Authors

Shuya Li, Fangping Wan, Hantao Shu, Tao Jiang, Dan Zhao, Jianyang Zeng

Correspondence

zhaodan2018@tsinghua.edu.cn (D.Z.), zengjy321@tsinghua.edu.cn (J.Z.)

In Brief

Identifying compound-protein interactions is one of the essential challenges in drug discovery. We developed MONN, a multi-objective neural network, which not only accurately predicts the binding affinities but also successfully captures the non-covalent interactions between compounds and proteins. MONN can prove to be a useful tool in exploring compound-protein interactions.

Li et al., 2020, Cell Systems *10*, 308–322
April 22, 2020
https://doi.org/10.1016/j.cels.2020.03.002

CellPress

CellPress
OPEN ACCESS

Cell Systems
Focus on RECOMB

Methods

MONN: A Multi-objective Neural Network for Predicting Compound-Protein Interactions and Affinities

Shuya Li,[1,5] Fangping Wan,[1,5] Hantao Shu,[1] Tao Jiang,[2,3] Dan Zhao,[1,*] and Jianyang Zeng[1,4,6,*]
[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China
[2]Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA
[3]Bioinformatics Division, BNRIST/Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[4]MOE Key Laboratory of Bioinformatics, Tsinghua University, Beijing 100084, China
[5]These authors contributed equally
[6]Lead Contact
*Correspondence: zhaodan2018@tsinghua.edu.cn (D.Z.), zengjy321@tsinghua.edu.cn (J.Z.)
https://doi.org/10.1016/j.cels.2020.03.002

SUMMARY

Computational approaches for understanding compound-protein interactions (CPIs) can greatly facilitate drug development. Recently, a number of deep-learning-based methods have been proposed to predict binding affinities and attempt to capture local interaction sites in compounds and proteins through neural attentions (i.e., neural network architectures that enable the interpretation of feature importance). Here, we compiled a benchmark dataset containing the inter-molecular non-covalent interactions for more than 10,000 compound-protein pairs and systematically evaluated the interpretability of neural attentions in existing models. We also developed a multi-objective neural network, called MONN, to predict both non-covalent interactions and binding affinities between compounds and proteins. Comprehensive evaluation demonstrated that MONN can successfully predict the non-covalent interactions between compounds and proteins that cannot be effectively captured by neural attentions in previous prediction methods. Moreover, MONN outperforms other state-of-the-art methods in predicting binding affinities. Source code for MONN is freely available for download at https://github.com/lishuya17/MONN.

INTRODUCTION

Elucidating the mechanisms of compound-protein interactions (CPIs) plays an essential role in drug discovery and development (Kola and Landis, 2004; Paul et al., 2010). Although various experimental assays (Inglese and Auld, 2008) have been widely applied for drug candidate screening and property characterization, identifying hit compounds from a large-scale chemical space is often time and resource consuming. To relieve this bottleneck, computational methods are typically used to reduce time and experimental efforts in drug development (Chen et al., 2016). For example, it has been shown that effective high-throughput virtual screening can greatly accelerate the lead discovery process (Rester, 2008).

Apart from the binding and functional assays, structure determination of compound-protein complexes can shed light on the molecular mechanisms of CPIs and thus significantly promote the lead optimization process. For instance, based on the molecular basis of CPIs revealed by the complex structures, drug developers can gain better insights into understanding how to improve the design of candidate compounds, for the purpose of enhancing binding specificities or avoiding side effects (Price et al., 2017). However, determining the atomic resolution structures of protein-ligand complexes through currently available experimental techniques, such as X-ray crystallography (Svergun et al., 2001), nuclear magnetic resonance (NMR) (Wüthrich, 1989), and cryoelectron microscopy (cryo-EM) (Nogales and Scheres, 2015), is still time-consuming in practice, resulting in only a limited number of solved structures (Berman et al., 2000). Therefore, a natural question arises: can computational virtual screening methods also provide useful mechanistic insights about CPIs in addition to predicting their binding affinities?

Molecular docking (e.g., AutoDock Vina [Trott and Olson, 2010] and GOLD [Verdonk et al., 2003]) and molecular dynamics (MD) simulations (Salsbury, 2010) have been popularly used in virtual screening of compounds interacting with proteins (Sousa et al., 2013). These methods have inherently good interpretability, as they can predict potential binding poses as well as binding affinities. Despite a number of successful stories about the applications of these structure-based computational methods, they still suffer from several limitations. One major limitation lies in their heavy dependence on the available high-quality 3D-structure data of the protein targets to handle fine-scale structural data during the simulation process. In addition, these molecular docking and MD simulation-based methods generally require tremendous computational resources.

To overcome the current limitations of the structure-based computational methods, a number of structure-free models (Cichonska et al., 2017; Airola and Pahikkala, 2018; Tsubaki et al., 2019; Gao et al., 2018; Karimi et al., 2019; Wan et al., 2019; Öztürk et al., 2018) have been developed for CPI prediction. An example is the similarity-based methods that take similarity matrices as descriptors of both compounds and proteins (Cichonska et al., 2017; Airola and Pahikkala, 2018). These methods mainly focus on the global similarities of entire compounds or proteins, while ignoring the detailed compositions of each molecule. Conversely, deep-learning-based methods (Tsubaki et al., 2019; Gao et al., 2018; Karimi et al., 2019; Öztürk et al., 2018) fully exploit the local features of both input compound structures and protein sequences to predict their binding affinities. DeepDTA (Öztürk et al., 2018) and DeepAffinity (Karimi et al., 2019) are representatives of deep-learning-based models that require only simplified molecular-input line-entry system (SMILES) strings of compounds and primary sequences of proteins as input. They employ the widely used deep neural network architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to process and extract contextual features from the input sequence data. Another group of methods rely on graph-based representations to encode the molecular features of compounds (Tsubaki et al., 2019; Gao et al., 2018), in which vertices represent atoms and edges represent chemical bonds. The graph convolution algorithms (Lei et al., 2017) are applied accordingly to extract useful molecular features from such graph representations of compounds. Although these structure-free methods can successfully predict the binding affinity between each pair of compound and protein, their interpretability is still limited due to the lack of structural information.

A fraction of these structure-free methods make use of neural attentions, which have been widely used in the deep-learning community to guide models to focus on those "important" features, and thus increase the interpretability of the prediction results (Vaswani et al., 2017; Santos et al., 2016). For the CPI prediction tasks (Tsubaki et al., 2019; Gao et al., 2018; Karimi et al., 2019), attentions are expected to be able to capture the local binding sites mediated by non-covalent interactions (e.g., hydrogen bonds and hydrophobic effects) between compounds and proteins. Although these methods demonstrated that real binding sites of compounds or proteins were enriched in their attention-highlighted regions in a few examples, systematic comparison and evaluation on this learning capacity are still lacking, probably due to the absence of benchmark datasets and evaluation standards. In this work, we constructed a benchmark dataset containing pairwise non-covalent interactions between atoms of compounds and residues of proteins for more than 10,000 compound-protein pairs and comprehensively evaluated the interpretability of different neural attention-based frameworks. Tests on our constructed benchmark dataset showed that current neural-attention-based approaches have difficulty in automatically capturing the accurate local non-covalent interactions between compounds and proteins without extra supervised guidance.

Based on this observation, we developed MONN, a multi-objective neural network, to learn both pairwise non-covalent interactions and binding affinities between compounds and proteins. MONN is a structure-free model that takes only graph representations of compounds and primary sequences of proteins as input, with capacity to handle large-scale datasets with relatively low computational complexity. The input information is processed by graph convolution networks and CNNs, but different from previous CPI prediction methods in the following aspects: (1) MONN uses a graph warp module (Ishiguro et al., 2019) in addition to a traditional graph convolution module (Lei et al., 2017) to learn both a global feature for the whole compound and local features for individual atoms of the compound to better capture the molecular features of compounds; (2) MONN contains a pairwise interaction prediction module, which can capture the non-covalent interactions between atoms of a compound and residues of a protein with extra supervision from the labels extracted from available high-quality 3D compound-protein complex structures; and (3) in MONN, the pairwise non-covalent interaction prediction results are further utilized to benefit the prediction of binding affinities, by effectively incorporating the shared information between compound and protein features into the downstream affinity prediction module.

Comprehensive cross-validation tests on our constructed benchmark dataset demonstrated that MONN can successfully learn the pairwise non-covalent interactions derived from high-quality structural data, even using the 3D structure-free information as input. We also used an additional test dataset constructed from the protein data bank (PDB) (Berman et al., 2000) to further validate the generalization ability of MONN. Moreover, extensive tests showed that MONN can achieve superior performance in predicting CPI-binding affinities over other state-of-the-art structure-free models. In addition, although the chemical rules, such as the correlation of hydrophobicity scores between compounds and proteins and the preference of atom and residue types for hydrogen bonds and π-stacking interactions, are not explicitly incorporated into the prediction framework, such features can still be effectively captured by MONN. All these results suggested that MONN can provide a useful tool for effectively modeling CPIs both locally and globally, and thus greatly facilitate the drug discovery process.

RESULTS

The Network Architecture of MONN Is Designed for Solving a Multi-objective Machine Learning Problem

MONN is an end-to-end neural network model (Figures 1 and 2) with two training objectives, whose main concept and key methodological terms are explained in Primer (Box 1) and Glossary (Box 2). One objective of MONN is to predict the non-covalent interactions between the atoms of a compound and the residues of its protein partner. We first define a pairwise interaction matrix to describe the non-covalent interactions between the input compound and protein pair. More specifically, for a compound with N_a non-hydrogen atoms and a protein with N_r residues, their pairwise interaction matrix $\boldsymbol{P}$ is defined as an $N_a \times N_r$ binary matrix, in which each element P_{ij} ($i = 1, 2, \cdots, N_a$ and $j = 1, 2, \cdots, N_r$) indicates whether there exists a non-covalent interaction (1 for existence, and 0 otherwise) between the i-th atom of the compound and the j-th residue of the protein when forming a complex structure. The interaction sites of the compound or protein can be then derived from this pairwise interaction matrix by maximizing over rows or columns (Figure 1A). The other objective of MONN is to predict the binding affinities (e.g., K_i, K_d, or

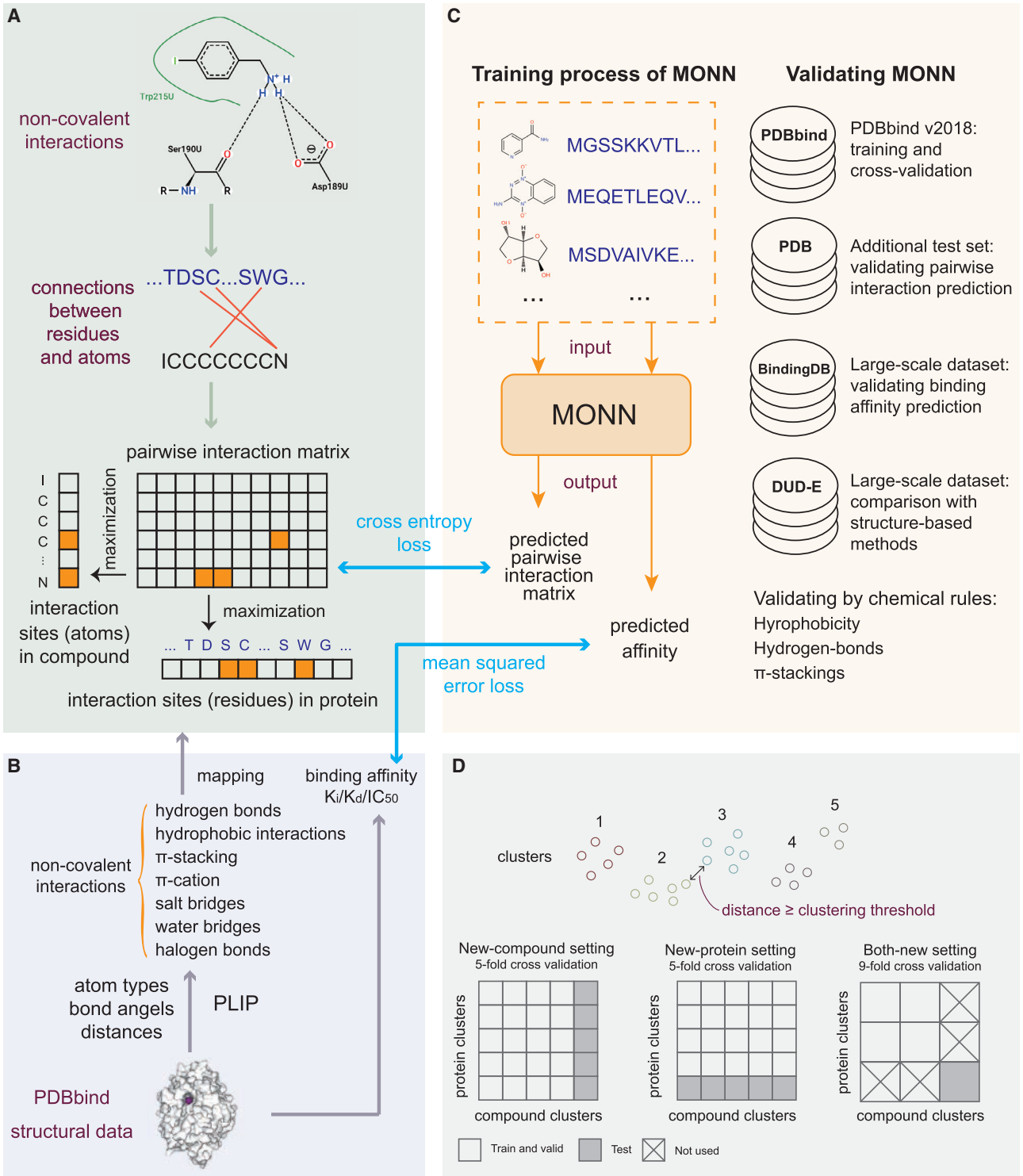


Figure 1. The Concept of MONN

(A) Non-covalent interactions between compounds and proteins. The pose view was generated by https://poseview.zbh.uni-hamburg.de/5z1c.

(B) Construction of the PDBbind-derived benchmark dataset.

(C) Training and validation of MONN.

(D) Three settings of the clustering-based cross-validation.

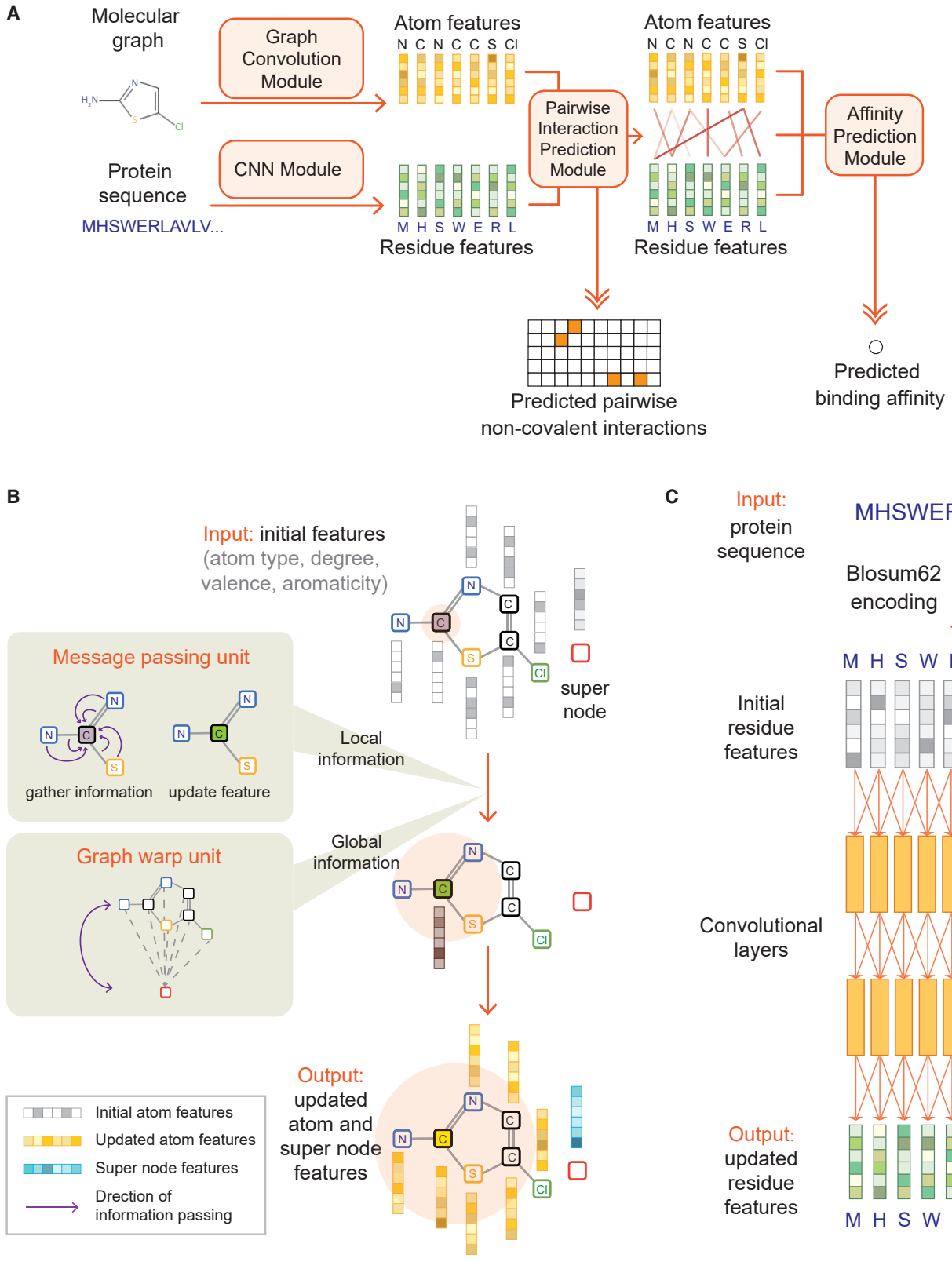


Figure 2. The Network Architecture of MONN

(A) The architecture overview of MONN. Given a compound-protein pair, a graph convolution module and a CNN module are first used to extract the atom and residue features from the input molecular graph and protein sequence, respectively. Then, these extracted atom and residue features are processed by a pairwise

(legend continued on next page)

Box 1. Primer

One of the key steps in drug discovery is to characterize the small molecule ligands of protein targets. However, it is still difficult to experimentally measure large-scale CPIs in an efficient way. Computational methods have been developed in this field to facilitate the discovery of hit or lead ligands for protein targets. Currently, there are still several challenges in establishing the computational models for compound-protein interaction prediction:

- The first challenge is the accuracy of predictions, which is also the most important goal of computational methods. Machine learning methods rely heavily on the amount and quality of training data to learn the regulations of the chemical or biological objects. However, there exists certain bias in many datasets for drug discovery. For example, similar compounds or proteins may be overly presented in the datasets, resulting from characterizing analogs of lead compounds and investigating important target proteins identified earlier. Such bias may possibly lead to overfitting of computational models and thus inaccurate reports of model performances.
- The second challenge is the limited accessibility of structural data at atomic resolution. Most structure-based machine learning methods for CPI prediction heavily rely on the compound-protein complex structures as their input, thus limiting their applications. In addition, these structure-based CPI prediction methods, including traditional CADD methods, generally require immense computational resource when processing the enormous atom coordinates in the 3D structures.
- The third challenge is the interpretability of current deep-learning-based models, especially for those structure-independent models. Although deep neural networks are powerful, they are also well known for their black-box nature. Despite that some of them attempted to seek for explanations through the attention mechanism (which tells the "focuses" of computational models), their performance is still quite limited, according to our systematic tests on a benchmark dataset.

In this paper, we developed a computational framework, called MONN, to address these challenges (Figure 1):

- First, through comprehensive evaluation of our model, we demonstrated that MONN can achieve superior performance than existing state-of-the-art CPI prediction methods. To avoid the bias introduced by similar compounds and proteins in training data, clustering-based cross-validation schemes were used to evaluate the ability of our model to make prediction for those compounds or proteins that are dissimilar with training data.
- Second, our model takes only protein sequences and chemical structures of compounds as input. Although during the training process, we can incorporate the non-covalent interaction labels derived from structural data to provide additional support for the affinity prediction task, during the application phase, our model does not require structural information as its input.
- Third, the network architecture of our model allows the extraction of contextual features from individual molecular components (i.e., atoms of compounds and residues of proteins), followed by the prediction of local non-covalent interactions between compounds and proteins. In addition to the successful prediction of binding affinities by MONN, the predicted local interactions can help provide useful mechanistic insights underlying the CPI events.

IC_{50}), which can also be regarded as a global measurement of the binding strength, between a protein and its ligand. The (predicted) binding affinity can be denoted by a real number $a \in \mathbb{R}$.

An input chemical compound with N_a atoms can be represented by a graph $G = \{V, E\}$, where each node $v_i \in V$, $i = 1, 2, \cdots, N_a$, corresponds to the i-th non-hydrogen atom in the compound, and each edge $e_{i_1,i_2} \in E$, $i_1, i_2 \in \{1, 2, \cdots, N_a\}$, corresponds to a chemical bond between the i_1-th and the i_2-th atoms. An input protein with N_r residues can be represented by a string of its primary sequence, denoted by $S = (r_1, r_2, \cdots, r_{N_r})$, where each r_j, $j = 1, 2, \cdots, N_r$, is either one of the 20 standard amino acids, or a letter "X" for any non-standard amino acid. Given a graph representation of a compound and a string representation of a protein sequence, our model is expected to output a predicted pairwise non-covalent interaction matrix $\boldsymbol{P} \in \mathbb{R}^{N_a \times N_r}$ and an estimated binding affinity value $a \in \mathbb{R}$.

MONN consists of four modules: (1) a graph convolution module for extracting the features of both individual atoms and the whole compound from a given molecular graph (Figure 2B), (2) a CNN module for extracting the features of individual residues from a given protein sequence (Figure 2C), (3) a pairwise interaction prediction module for predicting the probability of the non-covalent interaction between any atom-residue pair from the previously learned atom and residue features (Figure 3A), and (4) an affinity prediction module for predicting the binding affinity between the given pair of compound and protein, using the previously extracted molecular features, as well as the derived pairwise interaction matrix (Figure 3B). The graph convolution module and the CNN module effectively extract information from the local contexts for atoms of compounds and residues of proteins, and the pairwise interaction prediction module infers the potential non-covalent interactions from the previously learned local features. The basic idea of the affinity prediction module is to integrate information from both compounds and proteins to benefit the prediction of their binding affinities. During this process, the predicted non-covalent interactions are used to enable information sharing between the components of

interaction prediction module to derive the predicted pairwise interaction matrix, which also enables one to construct the links between atoms of the compound and residues of the protein. Finally, an affinity prediction module is used to integrate information from atom features, residue features, and the previously derived pairwise interactions to predict the binding affinity.

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

(C) The CNN module for encoding the features of an input protein sequence. More details can be found in STAR Methods.

Box 2. Glossary

Pairwise Interactions
To computationally describe the non-covalent interactions between a compound-protein pair, we first regard the compound as a list of atoms and the protein as a list of residues. A pairwise interaction matrix is represented by a [number of atoms]-by-[number of residues] matrix in which each element is a binary value indicating whether the corresponding atom-residue pair has an interaction or not.

Clustering Threshold
We use hierarchical clustering for splitting all the compounds (proteins) into groups (i.e., clusters) based on their similarities. A clustering threshold determines the minimal distance between clusters. For example, a threshold 0.3 for compound clusters means that any two compounds from different clusters have at least 30% difference in their chemical structures.

Convolutional Neural Networks (CNNs)
CNNs are neural networks widely used for processing image-like or sequence-like inputs (e.g., text strings describing chemical structures and protein sequences). For each position of a sequence, CNNs extract its local contextual features through capturing diverse sequence patterns of the surrounding regions (Alipanahi et al., 2015; Öztürk et al., 2018).

Recurrent Neural Networks (RNNs)
RNNs are neural networks designed for processing sequential data. Unlike CNNs that mainly focus on the detection of local patterns, RNNs scan the whole sequences to capture the long-range features of individual positions (i.e., the features related to distant positions) (Lipton et al., 2015; Karimi et al., 2019).

Graph Convolution Networks
In graph convolution networks, each atom of a compound can be regarded as a node. Different atoms can share information through their chemical bonds. The basic idea of graph convolution is to iteratively gather information from the neighbors of each node (i.e., atom), so that each single atom is aware of the molecular substructures around it (Lei et al., 2017; Tsubaki et al., 2019).

Graph Warp Unit
We use a variant of graph convolution network (Ishiguro et al., 2019), which extracts not only local features from neighbors of individual nodes but also global feature of a graph through a graph warp unit. In particular, a virtual node (also called super node) is introduced to connect with all the other nodes (i.e., atoms) in a graph representing the compound structure. Such a design allows remote atoms to directly communicate with each other through the virtual node, resulting in the detection of the global feature of the whole compound.

Neural Attentions
Neural attentions are generally designed to capture the importance of different input positions to the final prediction in deep-learning models. They are often realized by calculating a "weight" for each input position, which thus can provide certain interpretability about the contributions of individual input positions to the final prediction results (Vaswani et al., 2017; Santos et al., 2016).

compounds and proteins. Details about each module of MONN and the training process can be found in STAR Methods.

Systematic Evaluation Indicates the Limited Interpretability of Neural Attentions in CPI Prediction Models

A number of deep-learning-based methods (Tsubaki et al., 2019; Gao et al., 2018; Karimi et al., 2019; Wan et al., 2019; Öztürk et al., 2018) have been developed previously for modeling CPIs from 3D structure-free inputs. Despite their success in predicting binding affinities with relatively low computational complexity, interpretability is still considered as a challenge for these structure-independent methods. Several recent studies (Tsubaki et al., 2019; Gao et al., 2018; Karimi et al., 2019) sought interpretability by incorporating neural attentions (i.e., weighing the contributions of individual elements in the given input to the final predictions) into their model architectures. The attention weight can be regarded as a measure of importance of the feature at each position (e.g., an atom or a residue), and thus such an attention mechanism is expected to be able to explain the interaction sites between compounds and proteins. For example, Tsubaki et al. developed an end-to-end neural network with attentions for protein sequences (Tsubaki et al., 2019). They showed two examples in which the attention-highlighted regions were able to capture the real interaction sites in proteins. The method developed by Gao et al. involved both compound and protein attentions (Gao et al., 2018). By visualizing the attention weights, the authors demonstrated that the derived attention-highlighted regions derived from their model can successfully identify the interaction interface in a compound-protein complex. DeepAffinity reported an enrichment of true interaction sites in those regions with high attention scores in protein sequences for several examples (Karimi et al., 2019).

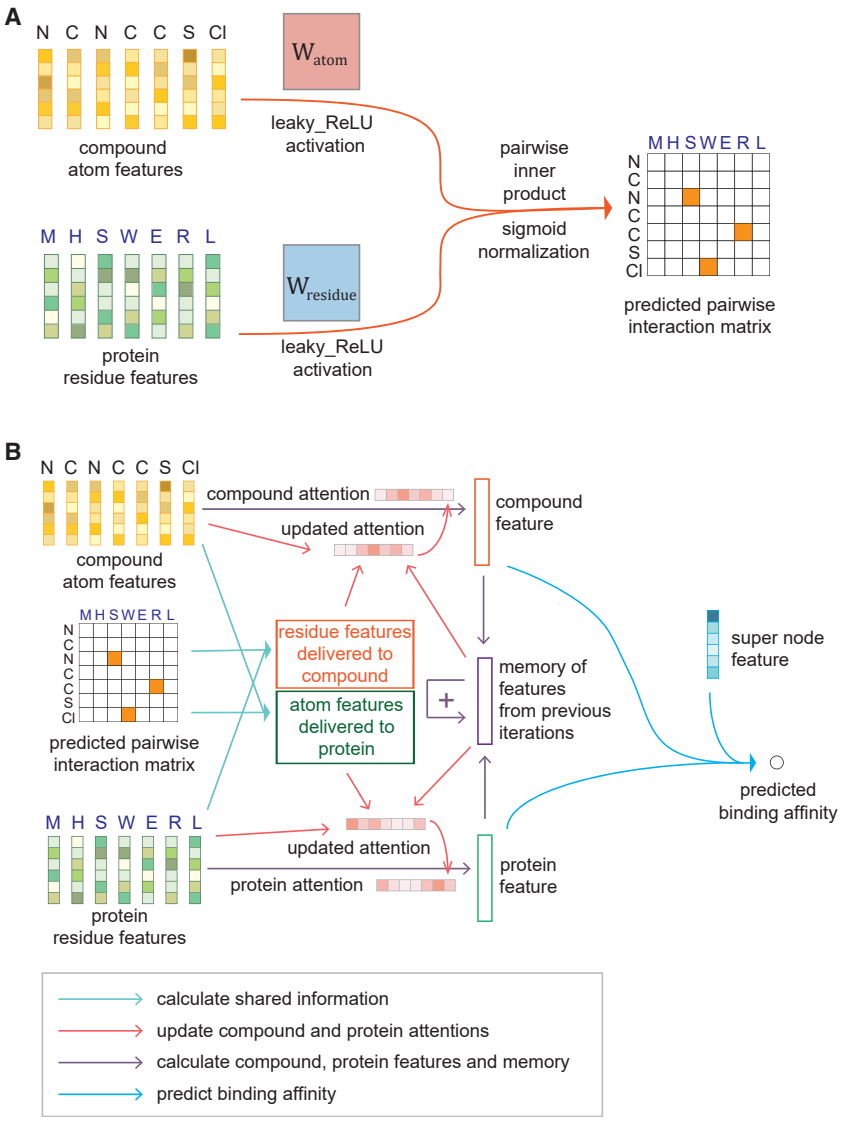


Figure 3. The Prediction Modules of MONN

(A) The pairwise interaction prediction module. Here, W_{atom} and $W_{residue}$ stand for the weight parameters of two single-layer neural networks that need to be learned.

(B) The affinity prediction module. More details can be found in STAR Methods.

However, there are still several limitations in the previous studies about the interpretability of the deep-learning-based CPI prediction methods. First, in these studies, the interpretability of neural attentions was evaluated only through one or several examples, and not comprehensively assessed by a large-scale benchmark dataset. In addition, the evaluations were conducted only by visualizing the attention weights (Tsubaki et al., 2019; Gao et al., 2018) or calculating the enrichment scores (Karimi et al., 2019), thus lacking a unified standard for systematically evaluating the interpretability of different attention-based models. More importantly, in these existing studies, attention weights were mainly used to infer the positions of the interaction sites, but the exact matchings between them (i.e., the pairwise interactions) still remained unknown.

To overcome these limitations, we conducted a systematic analysis to evaluate the interpretability of the neural attentions. We first constructed a benchmark dataset containing labels of pairwise non-covalent interactions for about 13,000 compound-protein complexes with available atomic resolution structures (STAR Methods; Table S1).

Then the interpretability was evaluated from the following three aspects: the ability of attentions to capture the interaction sites in compounds (at atom level), the interaction sites in proteins (at residue level), and the pairwise interactions between compounds and proteins. For these binary classification problems, we mainly used the average area under receiver operator characteristic curve (AUC) scores (i.e., averaging over all the compound-protein pairs in the test data) for performance evaluation. In addition, as in DeepAffinity (Karimi et al., 2019), we also calculated the enrichment score, which was defined as the fold change of the precision score of the trained model over the expected precision of random predictions (more details on these metrics can be found in STAR Methods).

Four different types of neural attentions used in existing compound-protein interaction prediction models were evaluated, including the method by Tsubaki et al. that calculates attentions only for proteins (Tsubaki et al., 2019), the method by Gao et al. that uses a bilinear function to generate compound and protein attentions for inferring their interaction sites (Gao et al., 2018), as well as a soft alignment matrix for inferring the pairwise interactions, and the separate and joint attentions proposed in DeepAffinity (Karimi et al., 2019) that calculate attentions for individual sites in the compounds or proteins and for their pairwise combinations, respectively. More details about the implementations of these neural attentions can be found in STAR Methods. The attention weights were obtained after training the models using the binding affinity labels, that is, without extra supervision from the pairwise interaction labels. The clustering-based cross-validation procedure (Mayr et al., 2018) was used during the training process, which ensured that similar compounds (or/and proteins) in the same clusters were not shared between training and test sets.

Three cross-validation settings were used in the evaluation, including the new-compound setting, in which the test compounds were never seen in the training process, the new-protein setting, in which the test proteins were never seen in the training data, and the both-new setting, in which both compounds and proteins in the test data were never seen during training. More details about the cross-validation procedures can be found in STAR Methods.

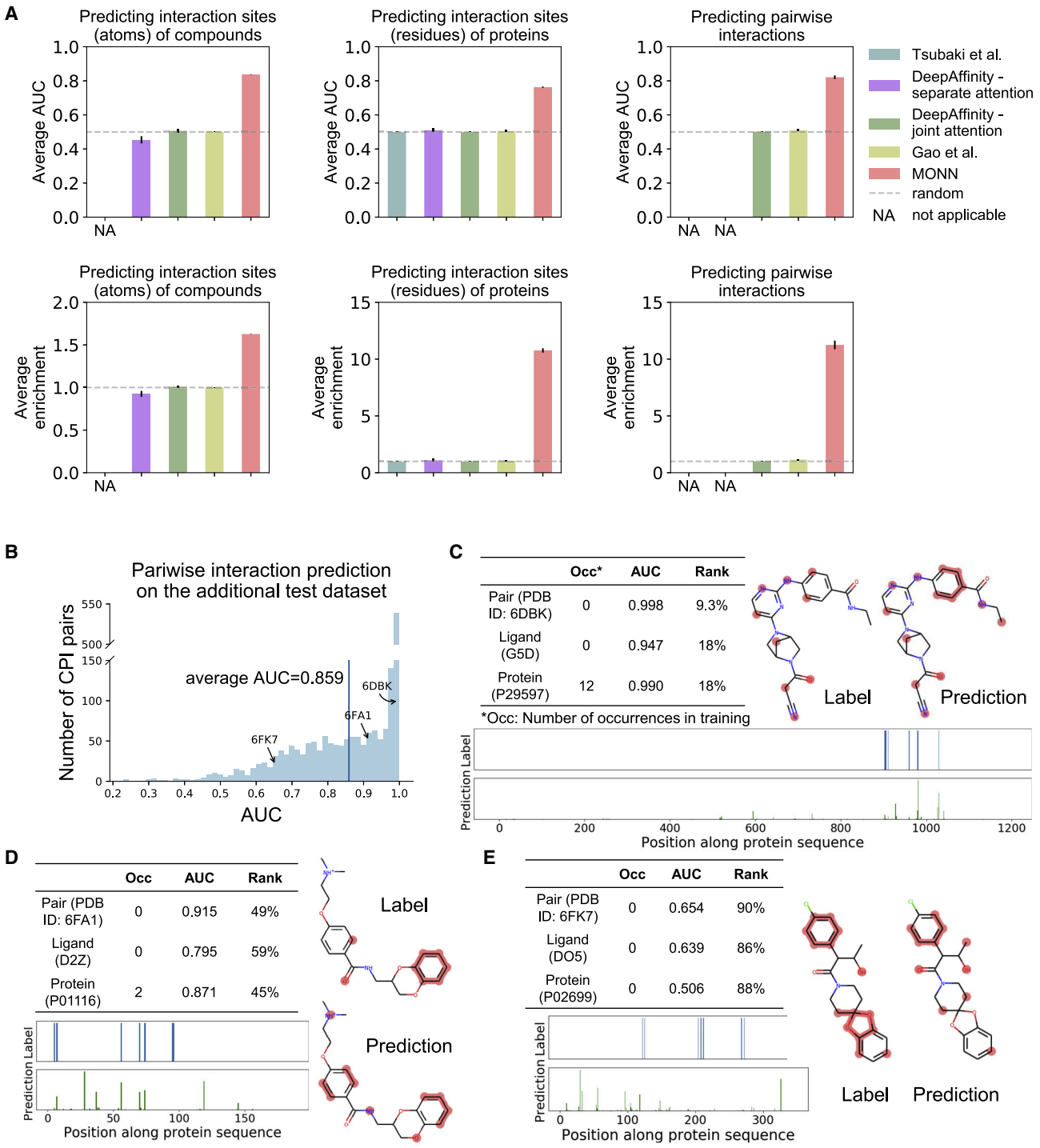


Figure 4. Performance Evaluation on the Interpretability of Different Neural Attentions and MONN for Predicting Non-covalent Interactions between Compounds and Proteins

(A) Evaluation on the PDBbind-derived benchmark dataset. Average AUC scores and average enrichment scores were used for evaluating the prediction of interaction sites (atoms) in compounds under the new-compound setting, interaction sites (residues) in proteins under the new-protein setting, and pairwise non-covalent interactions between compounds and proteins under the both-new setting. The mean values and standard deviations over 10 repeats of cross-validation with clustering threshold 0.3 are plotted. The ratios of positive and negative labels are about 1:1.44, 1:46.5, and 1:605 under these three cross-validation settings, respectively.

(B–E) Validating MONN on an additional test set derived from the PDB (Wang et al., 2005).

(B) The distribution of AUC scores for all the compound-protein pairs.

(legend continued on next page)

Under different prediction tasks and cross-validation settings, all the four types of neural attentions achieved average AUC and enrichment scores around 0.5 and 1, respectively, which were close to the scores of random predictions (Figures 4 and S3). These results suggested that, although the attention-highlighted regions and the real binding sites displayed accordance in some cases (Tsubaki et al., 2019; Gao et al., 2018; Karimi et al., 2019), they only showed poor correlation in a comprehensive test on a large-scale dataset. Thus, it seems not possible to derive the accurate predictions of non-covalent interactions between compounds and proteins from the attention-based models trained using only binding affinity labels (i.e., without pairwise interaction labels).

MONN Successfully Predicts Pairwise Non-covalent Interactions with Extra Supervision

Based on the above observation that neural attentions cannot automatically capture the non-covalent interactions between compounds and proteins, we speculated that extra supervision information can be used to guide our model to capture such local interactions. Instead of using attention mechanisms, MONN uses an individual module (i.e., the pairwise interaction prediction module) to learn the pairwise non-covalent interactions from given labels (STAR Methods). Meanwhile, through marginalizing the predicted pairwise interaction matrix, the predicted interaction sites in either compounds or proteins can also be derived.

The cross-validation settings and the metrics for evaluating the pairwise non-covalent interaction prediction results of our model were the same as described in the previous section and STAR Methods. As shown in Figure 4A, our model achieved average AUC scores of 0.837, 0.763, and 0.821 and average enrichment scores of 1.63, 10.8, and 11.3 under the three application settings (i.e., new-compound, new-protein, and both-new settings), respectively. Note that the values of the enrichment scores were not comparable among these three settings, due to the different ratios of positive-negative labels (STAR Methods). A more comprehensive comparison test (Figure S1) on our model and different neural attentions was performed for different prediction goals, cross-validation settings, and clustering thresholds, which showed that the predictions of MONN are effective and robust (average AUC scores decreased less than 5% with the clustering threshold increasing from 0.3 to 0.6). These results suggested that while the neural attentions have difficulty in interpreting the non-covalent interactions, MONN is able to accurately predict such interactions between compounds and proteins under different cross-validation settings.

The distributions of AUC scores of compound-protein pairs achieved by MONN for predicting the pairwise interactions under the three cross-validation settings are shown in Figure S2A. The results indicated that different compound-protein pairs indeed can have distinct performance. To further explore the potential factors affecting the performance of MONN on individual samples, we also examined the relationships between the achieved AUC scores and various properties of the test compound-protein pairs. It seemed that the occurrence of the same compound in training data did not affect the prediction performance much (Figure S2B), but the prediction performance was obviously affected by whether the same protein occurred in training data or not (Figure S2C). Also, molecular weights, logP values of compounds and sequence lengths of proteins may slightly influence the prediction performance (Figures S2D–S2F). In addition, MONN may perform better on certain protein families (e.g., kinases) than on others (Figures S2G–S2I).

To further examine the generalization ability of our model, we also validated MONN on an additional independent dataset containing pairwise non-covalent interactions between compounds and proteins. As our training data (i.e., the benchmark dataset derived from the PDBbind v2018 [Wang et al., 2005, 2004]) included all the high-quality structures of compound-protein complexes released in the PDB (Berman et al., 2000) before 2018, we also constructed an additional test dataset by collecting all the compound-protein complexes from the PDB with the release date from Jan 1st, 2018 to March 31st, 2019 (STAR Methods). In this extra test, MONN achieved average AUC 0.859 and average enrichment score 112.47 in predicting pairwise interactions of compound-protein pairs on this additional dataset (Figure 4B).

To visualize the prediction results of our model, we selected three representative compound-protein pairs ranked around 10%, 50%, and 90% in terms of the AUC scores and plotted the corresponding true labels and the predicted interaction sites in the compound structures and protein sequences (Figures 4C–4E). The example pair ranked around top 10% was a tyrosine kinase inhibitor binding to TYK2 (Figure 4C, PDB ID: 6DBK) (Fensome et al., 2018). In this example pair, top 40% of the predicted interaction sites (atoms) in the compound covered all the true interaction sites, and the high prediction scores also appeared around the true interaction sites along the protein sequence. The example pair ranked around the median prediction score contained a compound binding to KRAS (Figure 4D, PDB ID: 6FA1) (Quevedo et al., 2018). The predicted interaction sites of the compound had several overlaps with true interaction sites (5/8 recall) but also with several false positives. For example, the positively charged group in the compound was predicted as an interaction site, which is actually located outside the binding pocket. The predicted interaction sites (residues) of the protein had several overlaps with the true labels, but also with a number of false positives. The example pair ranked around 90% was a ligand binding to rhodopsin (Figure 4E, PDB ID: 6FK7) (Mattle et al., 2018). The deviation of the predicted interaction sites from true labels in this example was probably due to the scarcity of training data to support the prediction. All these visualization results demonstrated that the accuracies of MONN predictions were consistent with their corresponding rankings in AUC scores. Overall, the above comprehensive validation tests supported the strong predictive power of MONN.

(C–E) Three example pairs ranked around 10% (C), 50% (D), and 90% (E) in terms of AUC scores for the pairwise interaction prediction. We show the numbers of occurrences of the same pair, the same compound, and the same protein in training data, as well as the AUC scores and the corresponding ranks for the predicted pairwise interactions, the interaction sites (atoms) in compounds, and interaction sites (residues) in proteins. In the compound structures, true labels and top 40% predicted interaction sites are marked in red using RDKit (Landrum, 2006). In the protein sequences, the true labels and the MONN-predicted scores for individual positions are plotted.

CellPress
OPEN ACCESS

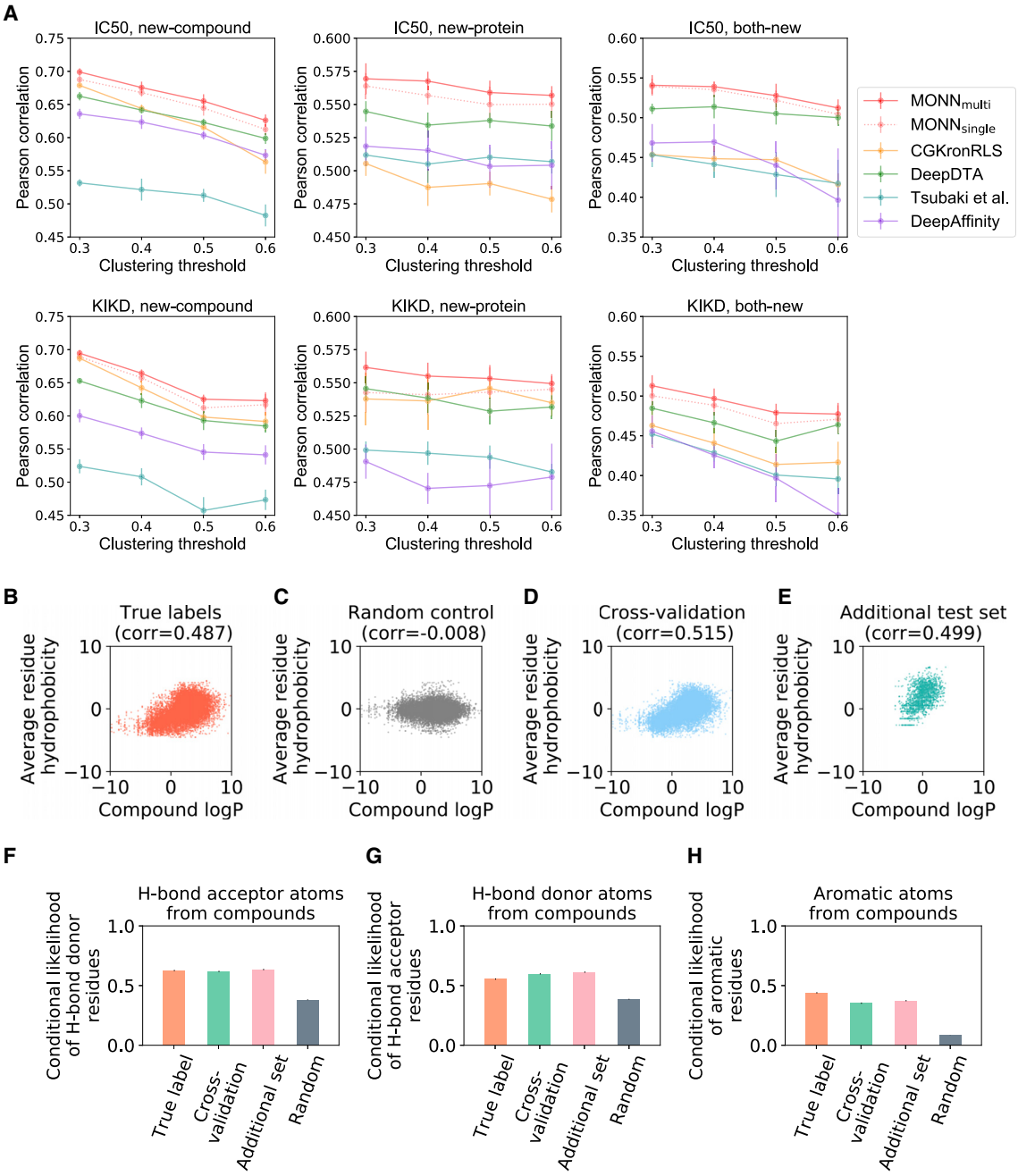


Figure 5. Evaluating MONN Using Binding Affinity Data and Known Chemical Rules

(A) Performance evaluation for MONN and baseline methods on binding affinity prediction, on both IC50 and KIKD datasets. Pearson correlations achieved by MONN with single (denoted as $MONN^{single}$) or multiple (denoted as $MONN^{multi}$) training objectives and four baseline methods, under three different cross-validation settings and four different clustering thresholds are shown. The mean values and standard deviations over 10 repeats of cross-validation are plotted.

(B–E) Correlations between the hydrophobicity scores of the compounds and the corresponding interaction sites (residues) in the proteins.

(B) The interaction residues were derived from the pairwise interaction labels of the benchmark dataset.

(C) The interaction residues of the proteins were derived from the randomly selected residues from the protein sequences. Here, the number of selected residues was the same as the number of true interaction sites in each protein sequence.

(D) The interaction residues of the proteins were predicted by MONN. Using a 9-fold cross-validation on the benchmark dataset under the both-new setting with a clustering threshold 0.3, the interaction residues were derived from the predicted pairwise interaction matrices of the test samples for each fold.

(E) The interaction residues were predicted for the compound-protein pairs in the additional test dataset, while the model was trained using the benchmark dataset.

Table 1. Performance Evaluation of Different Prediction Approaches on the BindingDB Dataset

Method	RMSE	Pearson Correlation
DeepAffinity (Single Model)	0.74	0.84
DeepAffinity (Parameter Ensemble)	0.73	0.84
DeepAffinity (Parameter + NN Ensemble)	0.71	0.86
DeepDTA (Single Model)	0.782	0.848
DeepDTA (Ensemble of 30 Models)	0.686	0.886
MONN (Single Model)	0.764	0.858
MONN (Ensemble of 30 Models)	0.658	0.895

The RMSE and Pearson correlation of DeepAffinity are adopted from the original paper (Karimi et al., 2019), in which "parameter ensemble" means averaging the predictions over the last 10 epochs, and "parameter + NN ensemble" means averaging predictions over the last 10 epochs of three networks with different hyper-parameter settings (i.e., averaging over 30 predictions).

MONN Successfully Predicts Binding Affinities with Single- and Multi-objective Learning

In this section, we examined the affinity prediction performance of MONN and compared it to that of other state-of-the-art models. For the binding affinity prediction task, we separated our PDBbind-derived dataset into two subsets, named IC50 (which contained IC_{50} values) and KIKD (which contained both K_i and K_d values). The main reason for such a separation was that IC_{50} values are generally dependent on experimental conditions and thus often considered noisier than the measured K_i and K_d values. Here, the IC50 dataset with the new-compound setting and clustering threshold 0.3 was used for hyper-parameter calibration. More details about training and hyper-parameter selection can be found in STAR Methods.

We considered the following state-of-the-art baseline methods for comparison: the similarity-based kernel method CGKronRLS (Cichonska et al., 2017), and the deep-learning-based methods, including DeepDTA (Öztürk et al., 2018), the method by Tsubaki et al. (Tsubaki et al., 2019) and DeepAffinity (Karimi et al., 2019). As in the previous sections, MONN and these baseline methods were evaluated under three different settings of clustering-based cross-validation (i.e., new-compound, new-protein, and both-new), in terms of Pearson correlation (Figure 5) and root mean squared error (RMSE, Figure S3). To investigate whether involving the extra supervision from the pairwise interaction labels can help predict the binding affinities, we mainly tested MONN under two conditions: one was a single objective model, denoted as $MONN_{single}$, which used only the affinity labels as supervision information, while the other was a multi-objective model, denoted as $MONN_{multi}$, which considered both pairwise interactions and binding affinities into the training objectives.

Our tests showed that both $MONN_{single}$ and $MONN_{multi}$ outperformed other baseline methods in all the three cross-validation settings with different clustering thresholds, on both IC50 and KIKD datasets (Figure 5). In particular, compared with the baseline methods, the multi-objective model ($MONN_{multi}$) achieved an increase in Pearson correlation by up to 3.6% (average 2.3%). In addition, the multi-objective model performed slightly better than the single objective one, which indicated that incorporating extra supervision information from pairwise interaction labels can further improve the binding affinity prediction task.

Since compound-protein complexes generally have limited structural availability, we further tested our model on a large-scale structure-free CPI dataset. To our best knowledge, among the baseline methods, only DeepAffinity has been evaluated previously on a large dataset with more than 260,000 training samples and more than 110,000 test samples, with the IC_{50} values derived from the BindingDB database (Gilson et al., 2016). We followed the same experimental settings as in DeepAffinity and also tested MONN and DeepDTA on the same dataset. The method by Tsubaki et al. and CGKronRLS are not suitable for this test mainly due to their limited scalability in processing such a large dataset. To make a fair comparison, we also evaluated an ensemble version (i.e., averaging predictions from several single models) of MONN on this BindingDB dataset, as in the DeepAffinity paper (Tsubaki et al., 2019) (details can be found in STAR Methods). The performances of DeepAffinity, DeepDTA, and MONN were evaluated in terms of RMSE and Pearson correlation, as listed in Table 1. When evaluating the single models, MONN achieved the best Pearson correlation (0.858). For all the methods, their performances can be largely improved through using the ensemble-based models. Among them, the ensemble version of MONN achieved the best performance (RMSE 0.658 and Pearson correlation 0.895). This comparison result suggested that, MONN can achieve better performance than the state-of-the-art baseline methods even when the structural data is not available.

To make a direct comparison between MONN and existing structure-based CPI prediction methods (including molecular docking and deep-learning-based models [Koes et al., 2013; Wallach et al., 2015; Ragoza et al., 2017; Gonczarek et al., 2018; Torng and Altman, 2019; Lim et al., 2019]), we also evaluated our model on the DUD-E dataset (Mysinger et al., 2012), which was widely used as a benchmark dataset for evaluating structure-based CPI prediction tasks. Among existing structure-based methods, Smina is a molecular docking method (Koes et al., 2013), and AtomNet, and the methods by Lim et al., Gonczarek et al., Torng et al., and Ragoza et al. are deep-learning-based methods dealing with structural input information. Although the method by Gonczarek et al. is structure independent, we still included it in the comparison as it was claimed to outperform most of the structure-based methods (Gonczarek et al., 2018). The DUD-E dataset contains 22,805 active compounds and 1,411,214 decoys (i.e., inactive compounds) for in total 102 proteins. Using the same training-test splitting strategy as in Wallach et al. (2015), we evaluate MONN under the "new-protein" condition, with 72 proteins as training data and the rest 30 proteins as test data. As shown in Table 2, the average AUC score over the 30 test proteins achieved by MONN was higher than those of the structure-based methods.

(F–H) Conditional likelihood scores (whose definition can also be found in the main text) measuring the preference of specific residue types given the properties of interaction sites (atoms) of the compounds, including hydrogen-bond acceptor atoms (F), hydrogen-bond donor atoms (G), and aromatic atoms (H). For each given type of atoms from the compounds, we considered the interaction residues for four different situations, as described in (B–E).

Table 2. Performance Evaluation of Different Prediction Approaches on the DUD-E Dataset

Method	Average AUC
Smina (Koes et al., 2013)	0.7
AtomNet (Wallach et al., 2015)	0.855
Ragoza et al. (Ragoza et al., 2017)	0.868
Torng et al. (Torng and Altman, 2019)	0.886
Gonczarek et al. (Gonczarek et al., 2018)	0.904
Lim et al. (Lim et al., 2019)	0.968
MONN	0.974

The performances of previous methods were directly obtained from the original papers (Wallach et al., 2015; Ragoza et al., 2017; Gonczarek et al., 2018; Torng and Altman, 2019; Lim et al., 2019). Note that the train-test split schemes of some methods were slightly different: Lim et al. used 72 proteins as training data and 25 as test data (Lim et al., 2019); Toring et al. and Ragoza et al. used 4-fold and 3-fold cross-validation strategies to evaluate their models, respectively (Torng and Altman, 2019; Ragoza et al., 2017).

We also examined the running time of our model. One of the structure-based models, AtomNet, requires about a week training on 6 Nvidia-K10 GPUs, as stated in the original paper (Wallach et al., 2015). In our test, MONN was able to fit this DUD-E dataset in about 20 h on one GeForce GTX 1080Ti GPU.

MONN Captures the Global Molecular Property

From the perspective of chemical properties, the size, shape, and hydrophobicity of a protein-binding pocket are essential for its interaction with a compound (Volkamer et al., 2012). Information about the size and shape of a binding pocket is usually hard to derive only based on its raw sequence, so we mainly examined the hydrophobicity of the potential binding residues predicted by MONN, through calculating the correlation between the hydrophobicity scores of the entire compounds and the average hydrophobicity scores of the predicted interaction sites (residues) in the proteins. Here, the hydrophobicity of the compound was measured by the logP value calculated by RDKit (Landrum, 2006), which is defined as the log ratio of the solubility of the compound in organic solvent (e.g., 1-octanol) against water (Wildman and Crippen, 1999). The hydrophobicity of the (predicted) interaction sites of a protein is defined as the average hydrophobicity score over the corresponding side chains (Lehninger et al., 2005). Here, the predicted interaction sites of the protein were selected from the top scored atom-residue pairs in the predicted pairwise interaction matrix $\boldsymbol{P}$, according to a cut-off value of mean ($\boldsymbol{P}$) + 3 × std ($\boldsymbol{P}$), where std(•) stands for the standard deviation. Next, the residues involved in the selected top atom-residue pairs were used for the downstream analysis.

The true interaction sites of proteins derived from the solved structures in the benchmark dataset showed a certain level of correlation (Pearson correlation 0.487) in hydrophobicity with their ligands (Figure 5B). As a control, no significant correlation was observed from randomly chosen residues (Figure 5C). The interaction sites of proteins predicted by MONN had similar correlations in hydrophobicity scores with their ligands (0.515 for cross-validation and 0.499 for the additional test dataset, Figures 5D and 5E), close to that of true labels. Note that the correlation achieved by the prediction result was slightly higher than that of true labels, this was probably because the information about hydrophobicity was somewhat over-represented in the predicted interaction sites selected according to the current threshold (i.e., 3 times of standard deviation above mean). If we used a stricter threshold, e.g., 4 or 5 times of standard deviation above mean, the resulting correlations (0.489 and 0.471, respectively) become closer to or lower than that of true labels, which suggested that MONN may focus more on other features under stricter thresholds. These results indicated that the predictions of MONN can also well reflect the relationships between compounds and proteins in terms of the global molecular property (i.e., the hydrophobicity).

MONN Captures the Chemical Rules of Non-covalent Interactions

The rules of non-covalent interactions and information of interaction types between compounds and proteins are not explicitly incorporated into MONN. Nevertheless, we examined whether MONN can automatically capture such chemical rules. Among the three most common non-covalent interaction types (i.e., hydrophobic interactions, hydrogen bonds, and π-stackings) between proteins and their ligands, we chose to analyze the preference of interaction partners for the atoms that can form hydrogen bonds or π-stackings. Hydrophobic interactions were not considered here, as hydrophobic carbons exist in all the 20 types of residues.

To characterize the preference of residues with a specific property under a given atom type of their interaction partners, we first define the conditional likelihood score p (residue property = x|atom property = y) = (number of residues $\in S(x)$ that interact with the atoms of property y)/(total number of residues interacting with the atoms of property y), where $S(x)$ represents the set of residues whose side chains contain at least one kind of elements satisfying the property x. To be more specific, S ("H-bond donor") = {H, K, N, Q, R, S, T, W, Y}, in which each residue has at least one hydrogen-bond donor in its side chain. Similarly, S ("H-bond acceptor") = {D, E, H, N, Q, S, T, Y}, and S ("aromatic") = {Y, W, F}. The corresponding properties of atoms from the compounds were calculated using RDKit (Landrum, 2006). Here, we calculated the conditional likelihood scores under different situations, in which the pairwise interactions referred in the above definition were obtained from either true labels, MONN predictions, or random choices (used as control).

A hydrogen bond is generally formed between a hydrogen donor group and an acceptor group. When the atoms from compounds are hydrogen-bond acceptors, the conditional likelihood of hydrogen-bond donor residues as their interaction partners (0.63, calculated using true labels) was much higher than the control residues (0.38, calculated using the randomly chosen residues, Figure 5F). The conditional likelihood scores calculated using MONN-predicted interaction sites were also relatively high (0.62 for cross-validation and 0.64 for the additional test, Figure 5F). Similarly, the hydrogen-bond acceptor residues from the MONN prediction results also had significantly higher conditional likelihood scores than the random control when their interaction partners were the hydrogen-bond donor atoms from the compounds (Figure 5G).

The π-stacking interactions generally occur between aromatic rings. There are three amino acids containing aromatic rings, i.e., phenylalanine, tryptophan and tyrosine. They generally had

higher conditional likelihood scores when their interaction partners are aromatic atoms from the compounds (0.44 calculated from true labels compared with 0.09 from random control, Figure 5H). In the MONN prediction results, the three aromatic residues also had higher conditional likelihood scores (0.35 for cross-validation and 0.37 for the additional test set, Figure 5H) than that from random control, which thus provided another evidence to support the reasonableness of the MONN prediction results.

In summary, the above results indicated that MONN can correctly capture the preferred interaction partners for different types of atoms in the compounds, according to the possibility of forming different kinds of non-covalent interactions.

DISCUSSION

Accurately predicting CPIs can greatly facilitate the drug discovery process. While several deep-learning-based tools have been proposed to predict binding affinities and improve virtual high-throughput screening, our approach MONN goes further to explore more about the mechanisms underlying CPIs. In this work, we demonstrated that MONN can successfully predict the pairwise non-covalent interaction matrices, which can also be used to infer the interaction sites in compounds and proteins. Comparison tests showed that MONN can outperform other state-of-the-art machine learning methods in predicting binding affinities. Besides, the structure-free input of MONN allows it to have a wider range of applications than those structure-dependent approaches. We also verified that the predictions of MONN are accordant with chemical rules, in terms of the correlation in hydrophobicity between interaction sites in compounds and proteins, and the preference of interaction partners for different atom types. All these results indicated that MONN can provide a powerful and useful tool to advance the drug development process.

MONN takes molecule graphs of compounds and protein sequences as input, which brings both advantages and limitations into our method. Structure-free inputs allow MONN to make predictions for proteins without known 3D structures. On the other hand, the sequence-only inputs may limit the amount of information directly conveyed into the model. Since most existing computer-aided drug design (CADD) tools rely on 3D structure data to predict binding poses, this kind of information is indeed useful for inferring the detailed binding mechanisms between proteins and compounds. For example, it would be beneficial if the region of binding pocket in a protein is already defined before predicting its interaction with compounds. Although the definition of "interaction sites" in our problem setting is not equivalent to that of binding pockets, we checked the percentage of our predicted interaction sites included in the regions of known binding pockets provided by PDBbind (Wang et al., 2005) (76.8%, 38.5%, and 33.4% for our model under new-compound, new-protein, and both-new settings, respectively, and 11.8% for random predictions as control). The results suggest that MONN may miss some of the true positions of binding pockets when it never sees the proteins in the training process. To better address the compound-protein interaction prediction task, a future method may face the problem of systematically integrating the abundant information extracted from high-quality 3D structures with the large-scale sequence data under the generalizable deep-learning-based frameworks.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- LEAD CONTACT AND MATERIALS AVAILABILITY
- METHOD DETAILS
  - Detailed Implementation of Individual Modules in MONN
  - Training
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - Construction of the Benchmark Dataset
  - Construction of the Additional Test Dataset for Validating the Pairwise Non-covalent Interaction Predictions
  - Evaluation of Different Types of Neural Attentions
  - Implementation of the Tested Neural Attentions
  - Clustering-based Cross Validation
  - Hyper-parameter Selection
  - Evaluation of MONN and Other Methods on the BindingDB-derived Dataset
- DATA AND CODE AVAILABILITY

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.cels.2020.03.002.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (61872216, 81630103, and 31900862) and the Turing AI Institute of Nanjing and the Zhongguancun Haihua Institute for Frontier Information Technology. The authors thank Mr. Tingzhong Tian and Dr. Hailin Hu for helpful discussions about this work and suggestions for the manuscript.

AUTHOR CONTRIBUTIONS

J.Z. and D.Z. conceived and supervised the research project. S.L. and F.W. developed the method. S.L., F.W., H.S., and T.J. conducted the analyses. All the authors contributed to the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: January 21, 2020
Revised: February 19, 2020
Accepted: March 5, 2020
Published: April 2, 2020

REFERENCES

Airola, A., and Pahikkala, T. (2018). Fast kronecker product kernel methods via generalized vec trick. IEEE Trans. Neural Netw. Learn. Syst. *29*, 3374–3387.

Alipanahi, B., Delong, A., Weirauch, M.T., and Frey, B.J. (2015). Predicting the sequence specificities of DNA-and rna-binding proteins by deep learning. Nat. Biotechnol. *33*, 831–838.

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000). The Protein Data Bank. Nucleic Acids Res *28*, 235–242.

Bickerton, G.R., Paolini, G.V., Besnard, J., Muresan, S., and Hopkins, A.L. (2012). Quantifying the chemical beauty of drugs. Nat. Chem. *4*, 90–98.

Chen, X., Yan, C.C., Zhang, X., Zhang, X., Dai, F., Yin, J., and Zhang, Y. (2016). Drug–target interaction prediction: databases, web servers and computational models. Brief. Bioinform. *17*, 696–712.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv https://arxiv.org/abs/1412.3555v1.

Cichonska, A., Ravikumar, B., Parri, E., Timonen, S., Pahikkala, T., Airola, A., Wennerberg, K., Rousu, J., and Aittokallio, T. (2017). Computational-experimental approach to drug-target interaction mapping: a case study on kinase inhibitors. PLoS Comput. Biol. *13*, e1005678.

Fensome, A., Ambler, C.M., Arnold, E., Banker, M.E., Brown, M.F., Chrencik, J., Clark, J.D., Dowty, M.E., Efremov, I.V., Flick, A., et al. (2018). Dual inhibition of TYK2 and JAK1 for the treatment of autoimmune diseases: discovery of ((s)-2, 2-difluorocyclopropyl)((1 r, 5 s)-3-(2-((1-methyl-1 h-pyrazol-4-yl) amino) pyrimidin-4-yl)-3, 8-diazabicyclo [3.2. 1] octan-8-yl) methanone (pf-06700841). J. Med. Chem. *61*, 8597–8612.

Gao, K.Y., Fokoue, A., Luo, H., Iyengar, A., Dey, S., and Zhang, P. (2018). Interpretable drug target prediction using deep neural representation. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), pp. 3371–3377.

Gilson, M.K., Liu, T., Baitaluk, M., Nicola, G., Hwang, L., and Chong, J. (2016). BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic Acids Res *44*, D1045–D1053.

Gonczarek, A., Tomczak, J.M., Zaręba, S., Kaczmar, J., Dąbrowski, P., and Walczak, M.J. (2018). Interaction prediction in structure-based virtual screening using deep learning. Comput. Biol. Med. *100*, 253–258.

Gower, J.C., and Ross, G.J.S. (1969). Minimum spanning trees and single linkage cluster analysis. Journal of the Royal Statistical Society: Series C (Applied Statistics) *18*, 54–64.

Henikoff, S., and Henikoff, J.G. (1992). Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. USA *89*, 10915–10919.

Inglese, J., and Auld, D.S. (2008). High throughput screening (HTS) techniques: applications in chemical biology. In Wiley Encyclopedia of Chemical Biology, T.P. Begley, ed. (American Cancer Society), pp. 1–15.

Ishiguro, K., Maeda, S.-i., and Koyama, M. (2019). Graph warp module: an auxiliary module for boosting the power of graph neural networks. arXiv https://arxiv.org/abs/1902.01020v4.

Karimi, M., Wu, D., Wang, Z., and Shen, Y. (2019). DeepAffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. Bioinformatics *35*, 3329–3338.

Koes, D.R., Baumgartner, M.P., and Camacho, C.J. (2013). Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. J. Chem. Inf. Model. *53*, 1893–1904.

Kola, I., and Landis, J. (2004). Can the pharmaceutical industry reduce attrition rates? Nat. Rev. Drug Discov. *3*, 711–715.

Landrum, G. (2006). RDKit: open-source cheminformatics. http://www.rdkit.org.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proc. IEEE *86*, 2278–2324.

Lehninger, A.L., Nelson, D.L., and Cox, M.M. (2005). Lehninger Principles of Biochemistry (MacMillan).

Lei, T., Jin, W., Barzilay, R., and Jaakkola, T. (2017). Deriving neural architectures from sequence and graph kernels. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 2024–2033.

Lim, J., Ryu, S., Park, K., Choe, Y.J., Ham, J., and Kim, W.Y. (2019). Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation. J. Chem. Inf. Model. *59*, 3981–3988.

Lipton, Z.C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. arXiv https://arxiv.org/abs/1506.00019v4.

Mattle, D., Kuhn, B., Aebi, J., Bedoucha, M., Kekilli, D., Grozinger, N., Alker, A., Rudolph, M.G., Schmid, G., Schertler, G.F.X., et al. (2018). Ligand channel in pharmacologically stabilized rhodopsin. Proc. Natl. Acad. Sci. USA *115*, 3640–3645.

Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., Wegner, J.K., Ceulemans, H., Clevert, D.A., and Hochreiter, S. (2018). Large-scale comparison of machine learning methods for drug target prediction on. ChEMBL. Chem. Sci. *9*, 5441–5451.

Mysinger, M.M., Carchia, M., Irwin, J.J., and Shoichet, B.K. (2012). Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. J. Med. Chem. *55*, 6582–6594.

Nam, H., Ha, J.-W., and Kim, J. (2017). Dual attention networks for multimodal reasoning and matching. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)), pp. 299–307.

Nogales, E., and Scheres, S.H. (2015). Cryo-EM: a unique tool for the visualization of macromolecular complexity. Mol. Cell *58*, 677–689.

Öztürk, H., Özgür, A., and Ozkirimli, E. (2018). DeepDTA: deep drug–target binding affinity prediction. Bioinformatics *34*, i821–i829.

Paul, S.M., Mytelka, D.S., Dunwiddie, C.T., Persinger, C.C., Munos, B.H., Lindborg, S.R., and Schacht, A.L. (2010). How to improve R&D productivity: the pharmaceutical industry's grand challenge. Nat. Rev. Drug Discov. *9*, 203–214.

Price, A.J., Howard, S., and Cons, B.D. (2017). Fragment-based drug discovery and its application to challenging drug targets. Essays Biochem *61*, 475–484.

Quevedo, C.E., Cruz-Migoni, A., Bery, N., Miller, A., Tanaka, T., Petch, D., Bataille, C.J.R., Lee, L.Y.W., Fallon, P.S., Tulmin, H., et al. (2018). Small molecule inhibitors of RAS-effector protein interactions derived using an intracellular antibody fragment. Nat. Commun. *9*, 3169.

Ragoza, M., Hochuli, J., Idrobo, E., Sunseri, J., and Koes, D.R. (2017). Protein–ligand scoring with convolutional neural networks. J. Chem. Inf. Model. *57*, 942–957.

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

Salentin, S., Schreiber, S., Haupt, V.J., Adasme, M.F., and Schroeder, M. (2015). PLIP: fully automated protein–ligand interaction profiler. Nucleic Acids Res *43*, W443–W447.

Salsbury, F.R., Jr. (2010). Molecular Dynamics simulations of protein dynamics and their relevance to drug discovery. Curr. Opin. Pharmacol. *10*, 738–744.

Santos, C.d., Tan, M., Xiang, B., and Zhou, B. (2016). Attentive pooling networks. arXiv https://arxiv.org/abs/1602.03609v1.

Sousa, S.F., Ribeiro, A.J., Coimbra, J.T., Neves, R.P., Martins, S.A., Moorthy, N.S., Fernandes, P.A., and Ramos, M.J. (2013). Protein-ligand docking in the new millennium-a retrospective of 10 years in the field. Curr. Med. Chem. *20*, 2296–2314.

Svergun, D.I., Petoukhov, M.V., and Koch, M.H. (2001). Determination of domain structure of proteins from X-ray solution scattering. Biophys. J. *80*, 2946–2953.

Torng, W., and Altman, R.B. (2019). Graph convolutional neural networks for predicting drug-target interactions. J. Chem. Inf. Model. *59*, 4131–4149.

Trott, O., and Olson, A.J. (2010). AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J. Comp. Chem. *31*, 455–461.

Tsubaki, M., Tomii, K., and Sese, J. (2019). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics *35*, 309–318.

UniProt Consortium (2019). UniProt: a worldwide hub of protein knowledge. Nucleic Acids Res *47*, D506–D515.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems, pp. 5998–6008.

Verdonk, M.L., Cole, J.C., Hartshorn, M.J., Murray, C.W., and Taylor, R.D. (2003). Improved protein–ligand docking using gold. Proteins *52*, 609–623.

Volkamer, A., Kuhn, D., Grombacher, T., Rippmann, F., and Rarey, M. (2012). Combining global and local measures for structure-based druggability predictions. J. Chem. Inf. Model. *52*, 360–372.

Wallach, I., Dzamba, M., and Heifets, A. (2015). AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. arXiv https://arxiv.org/abs/1510.02855v1.

Wan, F., Zhu, Y., Hu, H., Dai, A., Cai, X., Chen, L., Gong, H., Xia, T., Yang, D., Wang, M.W., and Zeng, J. (2019). DeepCPI: A deep learning-based framework for large-scale in silico drug screening. Genomics Proteomics Bioinformatics *17*, 478–495.

Wang, R., Fang, X., Lu, Y., and Wang, S. (2004). The PDBbind database: collection of binding affinities for protein- ligand complexes with known three-dimensional structures. J. Med. Chem. *47*, 2977–2980.

Wang, R., Fang, X., Lu, Y., Yang, C.Y., and Wang, S. (2005). The PDBbind database: methodologies and updates. J. Med. Chem. *48*, 4111–4119.

Wildman, S.A., and Crippen, G.M. (1999). Prediction of physicochemical parameters by atomic contributions. J. Chem. Inf. Comput. Sci. *39*, 868–873.

Wüthrich, K. (1989). Protein structure determination in solution by nuclear magnetic resonance spectroscopy. Science *243*, 45–50.

Zhao, M., Lee, W.P., Garrison, E.P., and Marth, G.T. (2013). SSW library: an SMID Smith-Waterman C/C++ library for use in genomic applications. PLoS One *8*, e82138.

CellPress
OPEN ACCESS

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and Algorithms		
MONN	This paper	https://github.com/lishuya17/MONN
CGKronRLS	Cichonska et al., 2017	https://github.com/ssalentin/plip
DeepDTA	Öztürk et al., 2018	https://github.com/hkmztrk/DeepDTA
DeepAffinity	Karimi et al., 2019	https://github.com/Shen-Lab/DeepAffinity
The method developed by Tsubaki et al.	Tsubaki et al., 2019	https://github.com/masashitsubaki/CPI_prediction
PLIP	Salentin et al., 2015	https://github.com/ssalentin/plip
Other		
PDBbind database	Wang et al., 2005	v2018, http://pdbbind.cn
Protein Data Bank	Berman et al., 2000	https://www.rcsb.org/
BindingDB database	Gilson et al., 2016	https://www.bindingdb.org/
DUD-E dataset	Mysinger et al., 2012	http://dude.docking.org/

LEAD CONTACT AND MATERIALS AVAILABILITY

Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Jianyang Zeng (zengjy321@tsinghua.edu.cn). This study did not generate new reagents.

METHOD DETAILS

Detailed Implementation of Individual Modules in MONN

The Graph Convolution Module

The graph convolution module (Figure 2B) takes the graph representation $G = \{V, E\}$ of a compound as input. More specifically, each node (i.e., atom) $v_i \in V$ is initially represented by a feature vector $\boldsymbol{v}_i^{init}$ of length 82, which is the concatenation of one-hot encodings representing the atom type, degree, explicit valence, implicit valence and aromaticity of the corresponding atom. Then, the initial atom features are transformed into $\mathbb{R}^{h_1}$ (h_1 is the size of hidden units) by a single-layer neural network:

$$\boldsymbol{v}_i^0 = f\left(\boldsymbol{W}_{init}\boldsymbol{v}_i^{init}\right), \quad \text{(Equation 1)}$$

where $f(\cdot)$ stands for the leaky ReLU activation function $f(x) = \max(0,x) + 0.1\min(0,x)$, and $\boldsymbol{W}_{init} \in \mathbb{R}^{h_1 \times 82}$. Note that for all the neural network layers described in this paper, unless otherwise stated, $f(\cdot)$ stands for the leaky ReLU activation function, $\boldsymbol{W}_x$ (x can be any subscript) stands for the learnable weight parameters, and the bias terms are omitted for clarity.

Each edge (i.e., chemical bond) $e_{i_1,i_2} \in E$ is represented by a feature vector $\boldsymbol{e}_{i_1,i_2}$ of length 6, which is the concatenation of one-hot encodings representing the bond type (i.e., single, double, triple or aromatic) and other properties, e.g., whether the bond is conjugated and whether it is in a ring.

The atom features are then processed by L iterations of graph convolution to produce a set of updated atom features $\{\boldsymbol{v}_i^L \in \mathbb{R}^{h_1}\}_{i=1}^{N_a}$ and a super node feature $\boldsymbol{s}^L \in \mathbb{R}^{h_1}$, which is an overall feature representation for the compound of interest. Note that the bond features are not updated during the whole process. In particular, at each iteration of graph convolution, the atom features are sequentially updated using both a basic message passing unit (Lei et al., 2017) and a graph warp unit (Ishiguro et al., 2019). The message passing unit executes the following two steps to extract the local features from the given graph: gathering information and updating information. During the first step (i.e., gathering information), each atom v_i gathers local information $\boldsymbol{t}_i^l$ from both its neighboring atoms and bonds, that is,

$$\boldsymbol{t}_i^l = \sum_{v_k \in Neighbor(v_i)} f\left(\boldsymbol{W}_{gather}^l\left[\boldsymbol{v}_k^{l-1}, \boldsymbol{e}_{i,k}\right]\right), \quad \text{(Equation 2)}$$

where $i = 1,2,\cdots,N_a$, $l = 1,2,\cdots,L$, $\boldsymbol{W}_{gather}^l \in \mathbb{R}^{h_1 \times (h_1+6)}$, $Neighbor(v_i)$ stands for the set of neighboring atoms of v_i, $\boldsymbol{v}_k^{l-1}$ represents the feature of atom v_k from the $(l-1)$-th layer, and $[\cdot,\cdot]$ stands for the concatenation operation. In the second step (i.e., updating

information), the gathered information and the atom features learned from the previous iteration are then processed to obtain the updated features $\{\boldsymbol{u}_i^l\}_{i=1}^{N_a}$ at each iteration l, that is,

$$\boldsymbol{u}_i^l = f\left(\boldsymbol{W}_{update}^l\left[\boldsymbol{t}_i^l, \boldsymbol{v}_i^{l-1}\right]\right), \quad \text{(Equation 3)}$$

where $i = 1, 2, \cdots, N_a$, $l = 1, 2, \cdots, L$, and $\boldsymbol{W}_{update}^l \in \mathbb{R}^{h_1 \times 2h_1}$.

The graph warp unit (Ishiguro et al., 2019) further improves the performance (the results of the corresponding ablation studies are shown in Figure S4) of graph convolution networks by introducing a super node s, which captures the global feature for the compound of interest. The extracted global feature will also be used in the affinity prediction module, as the properties of the whole compounds can generally contribute to their binding affinities. Before all the graph convolution iterations, the super node feature $\boldsymbol{s}^0 \in \mathbb{R}^{h_1}$ is initialized as the summation of all the atom features, that is,

$$\boldsymbol{s}^0 = \sum_{i=1}^{N_a} \boldsymbol{v}_i^0, \quad \text{(Equation 4)}$$

where $\boldsymbol{v}_i^0$ stands for the transformed initial feature of the i-th atom, which is described in STAR Methods of the main text.

Through information sharing between the super node and all the atoms, distant atoms in the graph can communicate effectively and efficiently through this super node, and thus a global feature can be extracted based on this technique (Ishiguro et al., 2019). In the lth iteration, the message passing unit is used to obtain the updated atom features $\{\boldsymbol{u}_i^l\}_{i=1}^{N_a}$, as described above. Accordingly, the graph warp unit first updates the super node feature by a single-layer neural network to obtain $\boldsymbol{u}_s^l$, that is,

$$\boldsymbol{u}_s^l = \tanh\left(\boldsymbol{W}_{super}^l \boldsymbol{s}^{l-1}\right), \quad \text{(Equation 5)}$$

where $l = 1, 2, \cdots, L$, $\tanh(\bullet)$ stands for the hyperbolic tangent activation function, $\boldsymbol{s}^{l-1}$ stands for the super node feature from the $(l-1)$-th iteration, and $\boldsymbol{W}_{super}^l \in \mathbb{R}^{h_1 \times h_1}$ denotes the learnable parameters.

Then, three steps are conducted to obtain the updated atom and super node features for each iteration.

Step 1: gathering information from the super node and the main nodes (atoms). The information ($\boldsymbol{u}_{s\to v}^l$) gathered from the super node is calculated by a single-layer neural network, that is,

$$\boldsymbol{u}_{s\to v}^l = \tanh\left(\boldsymbol{W}_{s\to v}^l \boldsymbol{s}^{l-1}\right), \quad \text{(Equation 6)}$$

where $l = 1, 2, \cdots, L$ and $\boldsymbol{W}_{s\to v}^l \in \mathbb{R}^{h_1 \times h_1}$.

To calculate the information gathered from each atom (main node), attention mechanism is used to weigh the contributions of individual atoms. As in the original version of graph warp unit (Ishiguro et al., 2019), a K-head attention mechanism is used to determine the contributions of features gathered from individual atoms, that is,

$$\boldsymbol{u}_{v\to s}^l = \tanh\left(\boldsymbol{W}_{v\to s}^l\left[\sum_{i=1}^{N_a}\alpha_{v,i}^{1,l}\boldsymbol{v}_i^{l-1}, \sum_{i=1}^{N_a}\alpha_{v,i}^{2,l}\boldsymbol{v}_i^{l-1}, \cdots, \sum_{i=1}^{N_a}\alpha_{v,i}^{K,l}\boldsymbol{v}_i^{l-1}\right]\right), \quad \text{(Equation 7)}$$

$$\alpha_{v,i}^{k,l} = \mathrm{softmax}\left(\boldsymbol{W}_{att}^{k,l}\boldsymbol{b}_{v,i}^{k,l}\right), \quad k = 1, 2, \cdots, K, i = 1, 2, \cdots, N_a, \quad \text{(Equation 8)}$$

$$\boldsymbol{b}_{v,i}^{k,l} = \tanh\left(\boldsymbol{W}_{vatt}^{k,l}\boldsymbol{v}_i^{l-1}\right) * \tanh\left(\boldsymbol{W}_{satt}^{k,l}\boldsymbol{s}^{l-1}\right), \quad k = 1, 2, \cdots, K, i = 1, 2, \cdots, N_a, \quad \text{(Equation 9)}$$

where $l = 1, 2, \cdots, L$, $\boldsymbol{W}_{v\to s}^l \in \mathbb{R}^{h_1 \times Kh_1}$, $\boldsymbol{W}_{att}^{k,l} \in \mathbb{R}^{1\times h_1}$, $\boldsymbol{W}_{vatt}^{k,l}$ and $\boldsymbol{W}_{satt}^{k,l} \in \mathbb{R}^{h_1 \times h_1}$, $\mathrm{softmax}(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$ stands for the softmax normalization function, $[\bullet, \bullet, \cdots, \bullet]$ denotes the concatenation operation, * denotes the element-wise multiplication, and K is the number of heads.

Step 2: calculating the passed information using warp gates. For the super node, an element-wise warp gate $\boldsymbol{g}_{v\to s}^l \in \mathbb{R}^{h_1}$ is used to combine the information from the super node itself $\boldsymbol{u}_s^l$ and the main nodes (atoms) $\boldsymbol{u}_{v\to s}^l$, that is,

$$\boldsymbol{g}_{v\to s}^l = \sigma\left(\boldsymbol{W}_{gate11}^l \boldsymbol{u}_{v\to s}^l + \boldsymbol{W}_{gate12}^l \boldsymbol{u}_s^l\right), \quad \text{(Equation 10)}$$

$$\boldsymbol{t}_{v\to s}^l = \left(\boldsymbol{1} - \boldsymbol{g}_{v\to s}^l\right) * \boldsymbol{u}_{v\to s}^l + \boldsymbol{g}_{v\to s}^l * \boldsymbol{u}_s^l, \quad \text{(Equation 11)}$$

where $l = 1, 2, \cdots, L$, $\boldsymbol{W}_{gate11}^l, \boldsymbol{W}_{gate12}^l \in \mathbb{R}^{h_1 \times h_1}$, $\sigma(\bullet)$ stands for the sigmoid activation function, $\boldsymbol{1}$ is an all-one vector of length h_1, and $\boldsymbol{t}_{v\to s}^l$ denotes the information passed to super node.

For each atom, similarly, an element-wise warp gate $\boldsymbol{g}_{s\to i}^l \in \mathbb{R}^{h_1}$ is used to combine the updated atom features $\boldsymbol{u}_i^l$ and information from the super node $\boldsymbol{u}_{s\to v}^l$, that is:

$$\boldsymbol{g}_{s\to i}^l = \sigma\left(\boldsymbol{W}_{gate21}^l \boldsymbol{u}_i^l + \boldsymbol{W}_{gate22}^l \boldsymbol{u}_{s\to v}^l\right), \quad \text{(Equation 12)}$$

$$\boldsymbol{t}^{l}_{s\to i} = \left(\mathbf{1} - \boldsymbol{g}^{l}_{s\to i}\right) * \boldsymbol{u}^{l}_{i} + \boldsymbol{g}^{l}_{s\to i} * \boldsymbol{u}^{l}_{s\to v}, \quad \text{(Equation 13)}$$

where $l = 1,2,\cdots,L$, $i = 1,2,\cdots,N_a$, $\boldsymbol{W}^{l}_{gate21}, \boldsymbol{W}^{l}_{gate22} \in \mathbb{R}^{h_1 \times h_1}$, and $\boldsymbol{t}^{l}_{s\to i}$ denotes the information passed to each atom.

Step 3: calculating the updated features using gated recurrent units (GRUs) (Chung et al., 2014). Here, two GRUs are used to determine the proportion of information updated at layer l for the atom and super node features, that is,

$$\boldsymbol{v}^{l}_{i} = \mathrm{GRU}_{v}\left(\boldsymbol{v}^{l-1}_{i}, \boldsymbol{t}^{l}_{s\to i}\right), i = 1,2,\cdots,N_a, \quad \text{(Equation 14)}$$

$$\boldsymbol{s}^{l} = \mathrm{GRU}_{s}\left(\boldsymbol{s}^{l-1}, \boldsymbol{t}^{l}_{v\to s}\right). \quad \text{(Equation 15)}$$

After completing L iterations of graph convolution, the final atom features $\{\boldsymbol{v}^{L}_{i} \in \mathbb{R}^{h_1}\}_{i=1}^{N_a}$ and the super node feature $\boldsymbol{s}^{L} \in \mathbb{R}^{h_1}$ are generated and then fed into the downstream modules. In the remaining part of this paper, we will drop the superscript L for clarity.

The CNN Module

The protein sequence is encoded using the BLOSUM62 matrix (Henikoff and Henikoff, 1992), that is, the initial feature of each residue is represented by the corresponding column of the BLOSUM62 matrix. The features of non-standard amino acids are zero-initialized. We use this encoding strategy instead of the commonly used one-hot encoding scheme for protein sequences, mainly because the BLOSUM62 matrix is a 20×20 matrix that has encoded the evolutionary relationships between amino acids, while the one-hot encoding scheme lacks such information. Then, the initial features are updated through typical 1-D convolution layers (LeCun et al., 1998) with a leaky ReLU activation function. Note that before being fed into each convolutional layer, the input is zero-padded to ensure that the number of the output features remains fixed. The specific architecture of the employed convolutional neural network is determined by three hyper-parameters, including the number of convolution layers, the number and the size of filters in each layer. In the end, we obtain the final output features $\{\boldsymbol{r}_j \in \mathbb{R}^{h_1}\}_{j=1}^{N_r}$ for all the residues along the protein sequence (Figure 2C), where h_1 stands for the number of output channels and N_r stands for the number of residues in the input protein sequence.

The Pairwise Interaction Prediction Module

To predict the pairwise interactions between a given compound-protein pair, the pairwise interaction prediction module (Figure 3A) uses the atom features $\{\boldsymbol{v}_i \in \mathbb{R}^{h_1}\}_{i=1}^{N_a}$ and the residue features $\{\boldsymbol{r}_j \in \mathbb{R}^{h_1}\}_{j=1}^{N_r}$ derived from the modules described above. The atom and residue features are first transformed into a compatible space by two single-layer neural networks separately. Then, the predicted probability of the interaction between an atom v_i and a residue r_j is derived based on the inner product between the transformed atom and residue features, normalized by a sigmoid function, that is,

$$P_{ij} = \sigma\left(f\left(\boldsymbol{W}_{atom}\boldsymbol{v}_i\right) \cdot f\left(\boldsymbol{W}_{residue}\boldsymbol{r}_j\right)\right), \quad \text{(Equation 16)}$$

where $i = 1,2,\cdots,N_a$, $j = 1,2,\cdots,N_r$, $\boldsymbol{W}_{atom}, \boldsymbol{W}_{residue} \in \mathbb{R}^{h_1 \times h_1}$, $\sigma(\cdot)$ represents the sigmoid function $\sigma(x) = \dfrac{1}{1+e^{-x}}$, and $\cdot$ denotes the inner product.

The Affinity Prediction Module

The affinity prediction module (Figure 3B) integrates information from not only the previously learned atom features $\{\boldsymbol{v}_i\}_{i=1}^{N_a}$, the super node feature $\boldsymbol{s}$ and the residue features $\{\boldsymbol{r}_j\}_{j=1}^{N_r}$, but also the predicted pairwise interaction matrix $\boldsymbol{P}$. Intuitively, $\boldsymbol{P}$ can be used to construct the links to share information between atom and residue features, which may thus provide additional useful information for predicting the binding affinity. Here, we describe how the binding affinity is predicted by our affinity prediction module.

First, the atom features $\{\boldsymbol{v}_i\}_{i=1}^{N_a}$ and the super node feature $\boldsymbol{s}$, which are originally constructed in the compound space, as well as the residue features $\{\boldsymbol{r}_j\}_{j=1}^{N_r}$, which are originally constructed in the protein space, are transformed into a compatible space for affinity prediction by single-layer neural networks, that is,

$$\boldsymbol{h}_{v,i} = f\left(\boldsymbol{W}_{v}\boldsymbol{v}_i\right), \quad \text{(Equation 17)}$$

$$\boldsymbol{h}_{s} = f\left(\boldsymbol{W}_{s}\boldsymbol{s}\right), \quad \text{(Equation 18)}$$

$$\boldsymbol{h}_{r,j} = f\left(\boldsymbol{W}_{r}\boldsymbol{r}_j\right), \quad \text{(Equation 19)}$$

where $\boldsymbol{W}_{v}, \boldsymbol{W}_{r}, \boldsymbol{W}_{s} \in \mathbb{R}^{h_2 \times h_1}$, and h_2 is the size of hidden units in the single-layer neural networks used in the affinity prediction module.

Next, we generate a fixed-size feature representation for each compound and each protein, from a list of transformed atom and residue features, using attention mechanism that has been widely used to enhance the performance of deep learning. In particular, the neural attention mechanism is introduced to weigh the contributions of features from individual atoms and residues, which has been proved to be more effective than simply averaging all the atom and residue features (the results of the corresponding ablation studies are shown in Figure S4). The dual attention network (DAN) (Nam et al., 2017) is a recently published method that can produce attentions for two given related entities (each with a list of features). For example, given an image with a sentence annotation, DAN generates a textual attention for the word features of the sentence and a visual attention for the spatial features of the image. Here, we

modify the original DAN framework by further exploiting the predicted pairwise interaction matrix to construct the direct links between atoms and residues. Information passing is thus enabled by gathering features of interaction partners through such links for each atom of the compound and each residue of the protein. The passed information is then incorporated into the calculation of compound and protein attentions by DAN. Next we will describe how to use the modified DAN framework to derive compound and protein attentions in the affinity prediction module to transform the atom and residue features into fixed-size vector representations.

Before all the DAN iterations, we first define the initial compound feature $\boldsymbol{h}_c^0 \in \mathbb{R}^{h_2}$, the initial protein feature $\boldsymbol{h}_p^0 \in \mathbb{R}^{h_2}$ and the initial memory vector $\boldsymbol{m}^0 \in \mathbb{R}^{h_2}$ (h_2 is the size of hidden units used in the affinity prediction module), that is,

$$\boldsymbol{h}_c^0 = \frac{1}{N_a}\sum_{i=0}^{N_a}\boldsymbol{h}_{v,i}, \tag{Equation 20}$$

$$\boldsymbol{h}_p^0 = \frac{1}{N_r}\sum_{j=0}^{N_r}\boldsymbol{h}_{r,j}, \tag{Equation 21}$$

$$\boldsymbol{m}^0 = \boldsymbol{h}_c^0 * \boldsymbol{h}_p^0, \tag{Equation 22}$$

where $\{\boldsymbol{h}_{v,i}\}_{i=1}^{N_a}$ and $\{\boldsymbol{h}_{r,j}\}_{j=1}^{N_r}$ are the transformed atom and residue features, as also described in the main text.

The compound feature, protein feature and the memory vector are then updated by D iterations of DAN. More specifically, at the dth iteration ($d = 1, 2, \cdots, D$), we first calculate the information shared between the atom features and the residue features, according to the predicted pairwise interaction matrix $\boldsymbol{P}$. Intuitively, each atom of the compound receives information from all the residues of the protein, weighed by the probabilities of local interactions between the atom-residue pairs, and vice versa for the residues. That is,

$$\boldsymbol{s}_{r\to v,i}^d = \sum_{j=1}^{N_r} P_{ij}\tanh\left(\boldsymbol{W}_{r\to v}^d\boldsymbol{h}_{r,j}\right), i = 1, 2, \cdots, N_a, \tag{Equation 23}$$

$$\boldsymbol{s}_{v\to r,j}^d = \sum_{i=1}^{N_a} P_{ij}\tanh\left(\boldsymbol{W}_{v\to r}^d\boldsymbol{h}_{v,i}\right), j = 1, 2, \cdots, N_r, \tag{Equation 24}$$

where $d = 1, 2, \cdots, D$, $\boldsymbol{W}_{r\to v}^d, \boldsymbol{W}_{v\to r}^d \in \mathbb{R}^{h_2\times h_2}$, $\{\boldsymbol{s}_{r\to v,i}^d\}_{i=1}^{N_a}$ are the information delivered from residues to atoms, $\{\boldsymbol{s}_{v\to r,j}^d\}_{j=1}^{N_r}$ are the information delivered from atoms to residues, and P_{ij} stands for the corresponding element in $\boldsymbol{P}$.

Next, the atom or residue features ($\{\boldsymbol{h}_{v,i}^d\}_{i=1}^{N_a}$ or $\{\boldsymbol{h}_{r,j}^d\}_{j=1}^{N_r}$), the memory vector from the previous iteration ($\boldsymbol{m}^{d-1}$) and the above derived shared information ($\{\boldsymbol{s}_{r\to v,i}^d\}_{i=1}^{N_a}$ or $\{\boldsymbol{s}_{v\to r,j}^d\}_{j=1}^{N_r}$) are combined to calculate the hidden states of the compound and protein attentions, that is,

$$\boldsymbol{b}_{v,i}^d = \tanh\left(\boldsymbol{W}_{vc}^d\boldsymbol{h}_{v,i}\right) * \tanh\left(\boldsymbol{W}_{mc}^d\boldsymbol{m}^{d-1}\right) * \boldsymbol{s}_{r\to v,i}^d, \tag{Equation 25}$$

$$\boldsymbol{b}_{r,j}^d = \tanh\left(\boldsymbol{W}_{rp}^d\boldsymbol{h}_{r,j}\right) * \tanh\left(\boldsymbol{W}_{mp}^d\boldsymbol{m}^{d-1}\right) * \boldsymbol{s}_{v\to j}^d, \tag{Equation 26}$$

where $d = 1, 2, \cdots, D$ and $\boldsymbol{W}_{vc}^d, \boldsymbol{W}_{mc}^d, \boldsymbol{W}_{rp}^d, \boldsymbol{W}_{mp}^d \in \mathbb{R}^{h_2\times h_2}$.

The compound and protein attentions are then calculated through two linear layers, normalized by the softmax function, that is,

$$\alpha_{v,i}^d = \mathrm{softmax}\left(\boldsymbol{W}_{vs}^d\boldsymbol{b}_{v,i}^d\right), \tag{Equation 27}$$

$$\alpha_{r,j}^d = \mathrm{softmax}\left(\boldsymbol{W}_{rs}^d\boldsymbol{b}_{r,j}^d\right), \tag{Equation 28}$$

where $d = 1, 2, \cdots, D$ and $\boldsymbol{W}_{vs}^d, \boldsymbol{W}_{rs}^d \in \mathbb{R}^{1\times h_2}$.

Finally, the fixed-size compound feature, protein feature and memory vector for the dth iteration is updated by current attentions:

$$\boldsymbol{h}_c^d = \sum_{i=0}^{N_a}\alpha_{v,i}^d\boldsymbol{h}_{v,i}, \tag{Equation 29}$$

$$\boldsymbol{h}_p^d = \sum_{j=0}^{N_r}\alpha_{r,j}^d\boldsymbol{h}_{r,j}, \tag{Equation 30}$$

$$\boldsymbol{m}^d = \mathrm{GRU}\left(\boldsymbol{m}^{d-1}, \boldsymbol{h}_c^d * \boldsymbol{h}_p^d\right), \tag{Equation 31}$$

where GRU stands for the gated recurrent unit (Chung et al., 2014).

After completing all the D iterations of updating the attentions, we obtain the fixed-size feature representations of the input compound graph and protein sequence (that is, $\boldsymbol{h}_c^D$ and $\boldsymbol{h}_p^D$, respectively).

Finally, $\boldsymbol{h}_c^D$ is concatenated with the transformed super node feature $\boldsymbol{h}_s$ to obtain a combined representation of the compound features (i.e., $[\boldsymbol{h}_c^D, \boldsymbol{h}_s]$). To fully exploit the relationship between this combined representation of the compound features and the representation of the protein features, we calculate their outer product, normalized by a leaky ReLU activation function f, and then followed by a linear regression layer to predict the binding affinity, that is,

$$a = \boldsymbol{W}_{\text{affinity}}\, f\left(\text{flatten}\left(\left[\boldsymbol{h}_c^D, \boldsymbol{h}_s\right] \otimes \boldsymbol{h}_p^D\right)\right), \quad \text{(Equation 32)}$$

where $\otimes$ denotes the outer product, flatten($\cdot$) reshapes the result of the outer product into a column vector of length $2h_2^2$, and $\boldsymbol{W}_{\text{affinity}} \in \mathbb{R}^{1\times 2h_2^2}$.

Training

For a training dataset with N samples (i.e., compound-protein pairs), we minimize the cross-entropy loss for pairwise non-covalent interaction prediction, which is defined as

$$L_P = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{N_a^{(n)}}\sum_{j=1}^{N_r^{(n)}} -\left(\widehat{P}_{ij}^{(n)} \log P_{ij}^{(n)} + \left(1-\widehat{P}_{ij}^{(n)}\right)\log\left(1-P_{ij}^{(n)}\right)\right), \quad \text{(Equation 33)}$$

where $P_{ij}^{(n)}$ and $\widehat{P}_{ij}^{(n)}$ stand for the predicted probability and the true binary label of the interaction between the i-th atom and the j-th residue in the nth sample, respectively, and $N_a^{(n)}$ and $N_r^{(n)}$ stand for the total number of atoms in the compound and the total number of residues in the protein in the nth sample, respectively.

For binding affinity prediction, the objective is to minimize the mean squared error, which is defined as

$$L_A = \frac{1}{N}\sum_{n=1}^{N}\left(a^{(n)} - \widehat{a}^{(n)}\right)^2, \quad \text{(Equation 34)}$$

where $a^{(n)}$ and $\widehat{a}^{(n)}$ stand for the predicted affinity and the true affinity label for the n-th sample, respectively.

In our multi-objective training process, we aim to minimize the combination of two losses to further enhance the binding affinity prediction, that is,

$$L = L_A + \lambda L_P, \quad \text{(Equation 35)}$$

where λ stands for a weight parameter controlling the contribution of L_p to the final affinity prediction. During the training process, we use a mini-batch stochastic gradient descent scheme to optimize the model parameters. For each training batch, compounds with different numbers of atoms and proteins with different numbers of residues are zero-padded to obtain the same input feature lengths. During the training process, the padded regions of features are masked so that they do not contribute to the calculation of the losses and gradients. MONN has about two million learnable parameters. A single MONN model can be trained within an hour on a Linux server with 48 logical CPU cores and one Nvidia Geforce GTX 1080Ti GPU.

QUANTIFICATION AND STATISTICAL ANALYSIS

Construction of the Benchmark Dataset

Our benchmark dataset was constructed mainly based on PDBbind (version 2018, the general set) (Wang et al., 2005; 2004), which contains a high-quality set of protein-ligand complexes with available structural data and corresponding binding affinities. Each complex was provided with an affinity value of certain measurement type (i.e., K_i, K_d, or IC_{50}).

For complexes in the PDBbind dataset, we obtained their 3D structures from the RCSB PDB (Berman et al., 2000) and then extracted the pairwise non-covalent interactions between compounds and proteins using PLIP (Salentin et al., 2015). After considering the atom types, distances and bond angles, PLIP recognized seven types of non-covalent interactions, including hydrogen bond, hydrophobic interaction, π-stacking, π-cation, salt bridge, water bridge and halogen bond.

There are in total 16,151 entries in the downloaded PDBbind dataset (Wang et al., 2005; 2004). We filtered these entries according to the following criteria: 1) each affinity value needs to be an accurate number, rather than a range or an approximation; 2) compounds need to have available and valid graph representations that can be processed by RDKit (Landrum, 2006); and 3) proteins can be successfully mapped to UniProt IDs with available sequence data (UniProt Consortium, 2019). Note that we used UniProt sequences instead of the original protein sequences directly extracted from the PDB structures, for the following reasons: First, the protein sequences in the PDB structures may be incomplete (e.g., only including some domains or lacking some flexible regions); Second, one protein may have different sequence variants in different PDB structures; Third, in a practical scenario, when we want to predict candidate ligands for a protein without known structure, it is generally more convenient to use its full-length primary sequence. In total, we obtained 13,306 compound-protein pairs satisfying these criteria.

Next, we calculated the pairwise interaction labels for the resulting 13,306 compound-protein pairs. The non-covalent interactions between the proteins and the corresponding ligands were extracted using the PLIP tool (Salentin et al., 2015) (https://github.com/

ssalentin/plip/). The ligand atoms involved in the non-covalent interactions were then mapped to the corresponding compound structures (downloaded from http://ligand-expo.rcsb.org/ld-download.html), which contain the unique names and indices for all the non-hydrogen atoms. For proteins, the residues involved in the non-covalent interactions were first mapped to the UniProt sequences using a sequence alignment tool (Zhao et al., 2013) (https://github.com/mengyao/Complete-Striped-Smith-Waterman-Library/). Then, we examined the mappings to control the quality of the generated interaction labels, and discarded those structures when the detected interactions cannot be correctly mapped into the molecular graphs of the compounds and the protein sequences. In addition, to further improve the mapping quality, we also filtered the complexes whose protein sequences in the PDB structures and the corresponding UniProt sequences had less than 90% matched residues. After the mapping process, we filled the pairwise interaction matrix according to the indices of the atoms and the residues involved in the non-covalent interactions to obtain the final interaction labels. After these procedures, we successfully constructed pairwise non-covalent interaction labels for about 95% of the compound protein pairs, resulting in 12,738 interaction matrices out of the 13,306 complex structures.

After constructing the benchmark dataset as described above, the performance of the pairwise interaction prediction was evaluated using all the available data. For binding affinity prediction, we further separated the compound-protein pairs according to the measurement types of binding affinities (i.e., K_i, K_d or IC_{50}), resulting in two affinity datasets, which were called the IC50 dataset and the KIKD dataset (which contained both K_i and K_d values), respectively. The reason for this separation was that the IC_{50} values usually depend on the experimental conditions and thus are often considered noisier than the K_i or K_d values. Here, we mainly used the IC50 dataset for hyper-parameter tuning for binding affinity prediction. For those repetitive records (defined as pairs with the same protein IDs and the same compound InChIs), we only kept the pairs with pairwise interaction labels and higher binding affinities. The raw affinity values were transformed into p(affinity) (i.e., $-\log_{10}(\text{affinity})$ [M]) to obtain the affinity labels. Finally, we obtained 5,340 and 6,689 unique pairs for the IC50 and KIKD datasets, respectively.

Construction of the Additional Test Dataset for Validating the Pairwise Non-covalent Interaction Predictions

We also downloaded the compound-protein complexes from the RCSB PDB database (Berman et al., 2000) to construct an additional test set for evaluating the pairwise non-covalent interaction prediction results of MONN. Since the PDBbind v2018 dataset, which was used as our training data, already contained the high-quality compound-protein complex structures with releasing date up to the end of 2017, here we downloaded structure data with date from January 2018 to March 2019 to avoid overlap between training and additional test datasets. Here three criteria were used to select the compound-protein complexes and control the quality of this additional dataset: (1) Each protein sequence can be mapped to a Uniprot sequence, with at least 90% matches in sequence alignment; (2) To remove ions, coenzymes and other crystallization assistant chemicals, we retained only those compound-protein pairs in which the quantitative estimation of drug-likeness (QED) scores (Bickerton et al., 2012) of the compounds are larger than 0.5; (3) Overlaps between training and test datasets were removed by discarding the test samples with both compound and protein similarities larger than 0.9 with any compound-protein pair in the training data. Then, the selected compound-protein complexes were processed using PLIP (Salentin et al., 2015) to extract the non-covalent interactions and construct the pairwise interaction labels using the same procedure as in the construction of the benchmark dataset.

Evaluation of Different Types of Neural Attentions

Evaluation Metrics

We used the average AUC scores and the average enrichment scores to evaluate the interpretability of neural attentions and prediction performance of MONN. Given a test dataset containing N samples, the average AUC score is defined as:

$$\text{average AUC score} = \frac{1}{N}\sum_{n=1}^{N}\text{AUC}(n), \qquad \text{(Equation 36)}$$

where AUC(n) stands for the area under the ROC curve calculated between the labels and the predictions of the nth sample.

The average enrichment score is defined as:

$$\text{average enrichment score} = \frac{1}{N}\sum_{n=1}^{N}\text{enrichment}(n), \qquad \text{(Equation 37)}$$

$$\text{enrichment}(n) = \frac{\text{precision}(n)}{\text{random\_precision}(n)}, \qquad \text{(Equation 38)}$$

where precision(n) stands for the precision score between the true labels and the binarized predictions (defined below) of the nth sample, and random_precision(n) stands for the expected precision of random predictions. Suppose that the positive-negative ratio of the whole dataset is $x_{pos} : x_{neg}$, and the length of prediction is l_{pred}. Then the binarization is realized by sorting the real-value predictions, and assigning 1 for top $\lceil l_{pred} \times x_{pos}/(x_{pos} + x_{neg}) \rceil$ predictions ($\lceil \cdot \rceil$ stands for the ceiling operation), and 0 for the rest. The random_precision(n) is calculated as random_precision(n) = $x_{pos}/(x_{pos} + x_{neg})$.

The upper limit of the average enrichment score is derived below:

$$\text{average enrichment score} = \frac{1}{N}\sum_{n=1}^{N}\text{enrichment}(n) \quad \text{(Equation 39)}$$

$$= \frac{1}{N}\sum_{n=1}^{N}\frac{\text{precision}(n)}{x_{pos}/(x_{pos}+x_{neg})}$$

$$\leq \frac{1}{N}\sum_{n=1}^{N}\frac{1}{x_{pos}/(x_{pos}+x_{neg})}$$

$$= 1+\frac{x_{neg}}{x_{pos}}.$$

Thus, with a relatively small positive-negative ratio (i.e., relatively large $\frac{x_{neg}}{x_{pos}}$), the upper limit of the average enrichment score is relatively high.

Implementation of the Tested Neural Attentions

We tested four types of neural attentions, by either using the original implementations or re-implementing and incorporating them into our MONN framework. For the protein attentions in the method by Tsubaki et al., we directly used their source code (Tsubaki et al., 2019). Since only the attention for proteins is generated, evaluations in terms of compound interaction sites and pairwise interactions are not applicable for the method by Tsubaki et al. For the bilinear attention in the method by Gao et al., since the source code was not released, we re-implemented it according to the descriptions provided from the original paper (Gao et al., 2018). For the separate and joint attentions of DeepAffinity (Karimi et al., 2019), their original implementations of attentions were mainly used to weigh short secondary protein structures (SPSs), rather than single residues. Thus, we re-implemented their attentions for testing them in our settings, which requires the protein attention to be calculated at residue resolution.

In our implementations, we used the transformed atom and residue features (denoted by $\{\boldsymbol{h}_{v,i}\}_{i=1}^{N_a}$ and $\{\boldsymbol{h}_{r,j}\}_{j=1}^{N_r}$, respectively) from the affinity prediction module of MONN (as described in STAR Methods) to calculate the compound and protein attentions according to the neural attention based methods mentioned above. After that, the resulting compound and protein attentions substitute the corresponding part (i.e., the DAN part) in our affinity prediction module, and then the models were trained according to the binding affinity labels. Note that our pairwise interaction prediction module was not used in this process. More details about how we implemented these neural attentions under our MONN framework are described below.

The Bilinear Attention of the Method by Gao et al.

Variables and parameters used only by this algorithm are marked with superscript $[G]$. The atom features and the residue features are combined to calculate a soft alignment matrix $\boldsymbol{P}^{[G]}$ of size $N_a \times N_r$:

$$P_{ij}^{[G]} = \tanh\left(\left(\boldsymbol{W}_v^{[G]}\boldsymbol{h}_{v,i}\right)^T\left(\boldsymbol{W}_r^{[G]}\boldsymbol{h}_{r,j}\right)\right), \quad \text{(Equation 40)}$$

where $\boldsymbol{W}_v^{[G]}, \boldsymbol{W}_r^{[G]} \in \mathbb{R}^{h_2 \times h_2}$ stands for the learnable weight parameters. Note that the bias terms for single-layer neural networks are also omitted for clarity in this section.

Then, the compound attentions $\{\alpha_{v,i}^{[G]}\}_{i=1}^{N_a}$ and protein attentions $\{\alpha_{r,j}^{[G]}\}_{j=1}^{N_r}$ are calculated using max-pooling over the soft alignment matrix $\boldsymbol{P}^{[G]}$, and then followed by a softmax normalization function, that is:

$$\alpha_{v,i}^{[G]} = \text{softmax}\left(\max_{j=1,2,\cdots,N_r} P_{ij}^{[G]}\right), \quad \text{(Equation 41)}$$

$$\alpha_{r,j}^{[G]} = \text{softmax}\left(\max_{i=1,2,\cdots,N_a} P_{ij}^{[G]}\right), \quad \text{(Equation 42)}$$

where $\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$ stands for the normalization function.

These attentions are then used for reducing the sizes of compound and protein features for predicting binding affinity values. To evaluate the interpretability, the compound attentions $\{\alpha_{v,i}^{[G]}\}_{i=1}^{N_a}$ and the protein attentions $\{\alpha_{r,j}^{[G]}\}_{j=1}^{N_r}$ are used as the predictions of interaction sites of compounds and proteins, respectively. The soft alignment matrix $\boldsymbol{P}^{[G]}$ is used as the predicted pairwise interaction matrix.

The Separate Attention of DeepAffinity

Variables and parameters used only by this algorithm are marked with superscript $[Ds]$. The separate attention of DeepAffinity (Karimi et al., 2019) calculates the soft self-attentions for the features of compounds and proteins, separately. In particular, the attentions for atom features of a compound ($\{\alpha_{v,i}^{[Ds]}\}_{i=1}^{N_a}$) are calculated by:

$$\boldsymbol{e}_{v,i}^{[Ds]} = \tanh\left(\boldsymbol{W}_{ev}^{[Ds]}\boldsymbol{h}_{v,i}\right), \quad \text{(Equation 43)}$$

$$\alpha_{v,i}^{[Ds]} = \mathrm{softmax}\left(\boldsymbol{W}_{av}^{[Ds]}\boldsymbol{e}_{v,i}^{[Ds]}\right), \quad \text{(Equation 44)}$$

where $\boldsymbol{W}_{ev}^{[Ds]} \in \mathbb{R}^{h_2 \times h_2}$ and $\boldsymbol{W}_{av}^{[Ds]} \in \mathbb{R}^{1 \times h_2}$ stand for the learnable weight parameters, and $\tanh(\cdot)$ stands for the hyperbolic tangent activation function.

Similarly, the attentions for residue features of a protein ($\{\alpha_{r,j}^{[Ds]}\}$) are calculated by:

$$\boldsymbol{e}_{r,j}^{[Ds]} = \tanh\left(\boldsymbol{W}_{er}^{[Ds]}\boldsymbol{h}_{r,j}\right), \quad \text{(Equation 45)}$$

$$\alpha_{r,j}^{[Ds]} = \mathrm{softmax}\left(\boldsymbol{W}_{ar}^{[Ds]}\boldsymbol{e}_{r,j}^{[Ds]}\right), \quad \text{(Equation 46)}$$

where $\boldsymbol{W}_{er}^{[Ds]} \in \mathbb{R}^{h_2 \times h_2}$ and $\boldsymbol{W}_{ar}^{[Ds]} \in \mathbb{R}^{1 \times h_2}$ stand for the learnable weight parameters.

The compound and protein attentions are then fed into the affinity prediction module of MONN. After trained by binding affinity labels, these attentions are used as the predictions of the interaction sites. Evaluation on pairwise interaction prediction is not applicable for this kind of attention, as the matchings between atoms and residues are not considered in this condition.

The Joint Attention of DeepAffinity

Variables and parameters used only by this algorithm are marked with superscript $[Dj]$. A pairwise interaction matrix $\boldsymbol{P}^{[Dj]}$ of size $N_a \times N_r$ is first calculated through a single layer neural network that combines both atom and residue features, that is,

$$P_{ij}^{[Dj]} = \tanh\left(\left(\boldsymbol{W}_{pv}^{[Dj]}\boldsymbol{h}_{v,i}\right)^T\left(\boldsymbol{W}_{pr}^{[Dj]}\boldsymbol{h}_{r,j}\right)\right), \quad \text{(Equation 47)}$$

where $\boldsymbol{W}_{pv}^{[Dj]}, \boldsymbol{W}_{pr}^{[Dj]} \in \mathbb{R}^{h_2 \times h_2}$ stand for the learnable weight parameters.

Then, a softmax function is used to normalize the pairwise interaction matrix over all the elements, to obtain a $N_a \times N_r$ attention matrix $\boldsymbol{A}^{[Dj]}$, that is,

$$A_{ij}^{[Dj]} = \frac{\exp\left(P_{ij}^{[Dj]}\right)}{\sum_{i=1}^{N_a}\sum_{j=1}^{N_r}\exp\left(P_{i,j}^{[Dj]}\right)}. \quad \text{(Equation 48)}$$

This normalized pairwise attention matrix $\boldsymbol{A}^{[Dj]}$ can be used in the evaluation of pairwise interaction prediction. In addition, through marginalizing $\boldsymbol{A}^{[Dj]}$, we can also derive the predictions of interaction sites in compounds or proteins, that is,

$$\alpha_{v,i}^{[Dj]} = \max_{j \in 1,2,\cdots,N_r} P_{i,j}^{[Dj]}, \quad \text{(Equation 49)}$$

$$\alpha_{r,j}^{[Dj]} = \max_{i \in 1,2,\cdots,N_a} P_{i,j}^{[Dj]}. \quad \text{(Equation 50)}$$

Since the original implementation of DeepAffinity with joint attention did not define the compound-wise/protein-wise attentions, here we modified our affinity prediction module, by replacing the outer product between compound and protein features with a combined feature, which is used in DeepAffinity:

$$\boldsymbol{b}_{i,j}^{[Dj]} = \tanh\left(\boldsymbol{W}_{bv}^{[Dj]}\boldsymbol{h}_{v,i} + \boldsymbol{W}_{br}^{[Dj]}\boldsymbol{h}_{r,j}\right), \quad \text{(Equation 51)}$$

$$\boldsymbol{h}^{[Dj]} = \sum_{i=1}^{N_a}\sum_{j=1}^{N_r} A_{ij}^{[Dj]}\boldsymbol{b}_{i,j}^{[Dj]}, \quad \text{(Equation 52)}$$

where $\boldsymbol{W}_{bv}^{[Dj]}, \boldsymbol{W}_{br}^{[Dj]} \in \mathbb{R}^{h_2 \times h_2}$ stand for the learnable weight parameters.

The final binding affinity is then predicted by:

$$a^{[Dj]} = \boldsymbol{W}_a^{[Dj]} f\left(\left[\boldsymbol{s}, f\left(\boldsymbol{h}^{[Dj]}\right)\right]\right), \quad \text{(Equation 53)}$$

where $f(\cdot)$ stand for the leaky ReLU activation function, $[\cdot,\cdot]$ stands for concatenation operation, and $\boldsymbol{s}$ represents the super node feature.

Clustering-based Cross Validation

Clustering

In the real datasets for compound-protein interaction prediction, there often exist highly similar compounds or proteins. To avoid the data redundancy problem caused by these similar compounds or proteins, we follow the same strategy as in (Mayr et al., 2018) and use a clustering-based cross validation strategy to evaluate the performance of our prediction model. The training-test splitting process in such a clustering-based cross validation scheme guarantees that the compounds (or proteins) within the same cluster, which share high similarities, are either all used in the training set, or all used in the test set. Note that an alternative approach to reduce data redundancy is to discard those high-similar data points. However, we argue that the clustering-based scheme would allow us to make better use of all available data. Here, we use the single-linkage clustering algorithm (Gower and Ross, 1969), which ensures that the minimal distance between any two clusters is above a given clustering threshold. The distance between a pair of compounds (c_i, c_j), is defined as

$$\text{Distance}(c_i, c_j) = 1 - \text{Jaccard}(\text{MF}(c_i), \text{MF}(c_j)), \quad \text{(Equation 54)}$$

where MF(•) stands for the Morgan fingerprints calculated by RDKit (Landrum, 2006) and Jaccard(•, •) denotes the Jaccard similarity.

The distance between a pair of proteins (p_i, p_j) is defined as

$$\text{Distance}(p_i, p_j) = 1 - \frac{SW(p_i, p_j)}{\sqrt{(\text{SW}(p_i, p_i)\text{SW}(p_j, p_j))}}, \quad \text{(Equation 55)}$$

where SW(•, •) stands for the Smith-Waterman alignment score calculated based on the SSW library (Zhao et al., 2013). The clustering threshold (which is a distance parameter used in the clustering algorithm) is defined as the mimimal distance between any compounds (proteins) from different clusters. The clustering threshold value used in our paper is selected from {0.3, 0.4, 0.5, 0.6}. We choose 0.3 as the lower limit of the threshold, because a distance smaller than 0.3 would not be separable enough to avoid the data redundancy problem, consistent with the previous study (Mayr et al., 2018). The upper limit of our clustering threshold is set to 0.6, because a higher threshold will lead to so large clusters that the splitting of training-test data would become highly imbalanced (i.e., too much training data and too little test data, or vice versa, Tables S2 and S3).

Cross Validation Settings

After generating the compound and protein clusters, three settings are considered during the cross validation process, i.e., the new-compound setting, the new-protein setting and the both-new setting. To explain these settings, we denote the training, validation and test sets by D_{train}, D_{valid} and D_{test}, respectively, and use (c_i, p_i) to represent the compound-protein pair of the i-th sample ($i = 1, 2, \cdots, N$).

In the new-compound setting, cross validation is performed on compound clusters, so that the compound-protein pairs with compounds from the same cluster cannot be shared across training, valid and test sets. That is, for any two compound-protein pairs (c_i, p_i) and (c_j, p_j) from different sets, c_i and c_j must come from different compound clusters.

In the new-protein setting, cross validation is performed on protein clusters, so that the compound-protein pairs with proteins from the same cluster cannot be shared across training, valid and test sets. That is, for any two compound-protein pairs (c_i, p_i) and (c_j, p_j) from different sets, p_i and p_j must come from different protein clusters.

In the both-new setting, both compound clusters and protein clusters cannot be shared across training, valid and test sets. That is, for any two compound-protein pairs (c_i, p_i) and (c_j, p_j) from different sets, c_i and c_j must come from different compound clusters, and p_i and p_j must come from different protein clusters as well.

For the new-compound and the new-protein settings, we use five-fold cross validation, and the train-valid-test splitting ratio is approximately 7 : 1 : 2. Note that here the ratio is an approximation, because the splitting is performed on clusters, and the number of data points among individual clusters is not necessarily evenly distributed. For the both-new setting, we randomly partition the pairs of compound-protein clusters into a 3×3 grid. Then, a nine-fold cross validation (Airola and Pahikkala, 2018) was conducted according to the following three steps: 1) select a grid as the test set; 2) discard the four grids that share compound or protein clusters with the selected one; 3) reorganize the remaining grids as a new 3×3 grid setting and randomly select one grid as the validation set, and the four grids that do not share any compound or protein cluster with the validation set are used as the training set. Such a cross-validation strategy results in an approximately 16 : 4 : 9 train-valid-test ratio.

Hyper-parameter Selection

Four baseline models were used in the performance comparison for the binding affinity prediction task, including CGKronRLS (Cichonska et al., 2017), DeepDTA (Öztürk et al., 2018), the method by Tsubaki et al. (Tsubaki et al., 2019) and DeepAffinity (Karimi et al., 2019). The method by Gao et al. was not included here, because its source code was not released, and the model requires additional input information (i.e., gene ontology terms of proteins) (Gao et al., 2018). For our model and all the baseline methods, each cross-validation setting (i.e., new-compound, new-protein or both-new) has a specific set of hyper-parameters. For MONN, the hyper-parameter selection was performed with both training objectives. The details of the hyper-parameter spaces for MONN and the baseline methods are provided below:

- For our model, the number of graph convolution iterations $L = 4$ and the number of DAN iterations $D = 2$ were determined using the same schemes as in the original papers (Lei et al., 2017; Nam et al., 2017). The hidden size h_1 is set to 128. Other hyper-parameters include the number of heads of the K-head attention used in the graph convolution module $K \in \{1, 2\}$, the number of CNN layers $L_{CNN} \in \{2, 4\}$, the kernel size of the CNN layers $S_{kernel} \in \{5, 7\}$, the hidden size of the affinity prediction module $h_2 \in \{64, 128\}$, and the ratio of pairwise loss $\lambda \in \{0, 0.1, 1\}$. A grid search was used to find the best combination of these four hyper-parameters. Although it is not practical to search all the hyper-parameter space thoroughly in the grid search process, we also tested the influence of the number of graph convolution iterations while fixing the other hyper-parameters after the grid search (Figure S5). The result suggested that the numbers of graph convolution iterations ranging from one to four achieved quite similar performance in both affinity prediction and pairwise interaction prediction tasks. Probably this was because the super node had already allowed sufficient information passing across remote nodes at a small number of iterations.
- For CGKronRLS (Airola and Pahikkala, 2018), the regularization parameter was chosen from $\{2^{-5}, 2^{-4}, \cdots, 2^{5}\}$.
- For DeepDTA (Öztürk et al., 2018), a grid search was conducted to select the best combination of different hyper-parameters, including the number of filters from $\{16, 32, 64, 128\}$, the length of sequence windows from $\{4, 8, 12\}$ and the length of SMILES windows from $\{4, 6, 8\}$. These ranges were adopted from the original paper (Öztürk et al., 2018).
- For the method by Tsubaki et al., according to the original paper (Tsubaki et al., 2019), the radius of compound subgraph and the length of the protein "ngram" (i.e., n-mer from the protein sequence) were selected from $\{(0, 1), (1, 2), (2, 3)\}$, the hidden dimension of ngram and atom embedding was selected from $\{5, 10, 20, 30\}$, and the number of layers of both CNN and GNN was selected from $\{2, 3, 4\}$. A grid search was performed to select the best combination from these ranges. Note that the method by Tsubaki et al. is originally a classification model. Here, we modified its last hidden layer by removing the activation function, and changed its loss function to the mean squared error (MSE) to perform the regression task.
- For DeepAffinity (Karimi et al., 2019), since the authors did not provide a specific hyper-parameter set and their model requires a pretraining step, we directly used their pretrained RNN-CNN models and then fine-tuned them with our data. For each cross-validation setting, we chose a better DeepAffinity model from those with either joint or separate attentions.

Apart from all the hyper-parameters mentioned above, all the methods have another hyper-parameter, i.e., the number of epochs (or iterations) during the training process. We used the RMSE as the evaluation metric from the validation set to select the best value of this hyper-parameter for all the methods. The maximum number of epochs for our method was set to 30. For DeepDTA, the method by Tsubaki et al. and DeepAffinity, we used their default maximum numbers of epochs (which is 100). For CGKronRLS, we set the maximum number of iterations to 500, as the performance no longer increased after 500 iterations.

For each cross-validation setting, the best hyper-parameters were selected based on the IC50 dataset with clustering threshold 0.3. The same parameters were used for other scenarios (i.e., other thresholds and the KIKD dataset) under the corresponding cross-validation setting. For pairwise interaction prediction, we also used the best hyper-parameters selected based on the affinity prediction results. We did not select the hyper-parameters according to the performance of the pairwise prediction task (that is, only single training objective of MONN was used), for the following two reasons. First, in MONN, those hyper-parameters in the affinity prediction module will not be optimized under this condition. Second, the baseline methods do not include a direct supervised optimization procedure for local interaction prediction. So for fair comparison, we did not specifically tune the hyper-parameters for pairwise interaction prediction for MONN.

Evaluation of MONN and Other Methods on the BindingDB-derived Dataset

We tested MONN and DeepDTA using the same dataset derived from BindingDB (Gilson et al., 2016) as in DeepAffinity (Karimi et al., 2019). The other two baseline methods, CGKronRLS (Airola and Pahikkala, 2018) and the method by Tsubaki et al. (Tsubaki et al., 2019), were not included in this test for the following reasons, respectively: CGKronRLS requires the input of compound and protein similarity matrices, and its space and time usage increases dramatically with the input data size (for example, loading the float32-format similarity matrix of all the 202,169 unique compounds from the BindingDB training set needs 149 Gb memory, and processing such a huge matrix by the CGKronRLS algorithm is nearly infeasible). Unlike other deep learning-based baseline methods that process batches of input samples, the implementation of the method by Tsubaki et al. allows only one sample to be processed at a time, so training this method on such a large dataset would be too time-consuming and nearly impractical.

The performance of DeepAffinity on this dataset was obtained from the original paper (Karimi et al., 2019). In addition to a "single model", the performance of several ensemble versions of DeepAffinity (i.e., averaging predictions over several single models) was also reported in (Karimi et al., 2019). In particular, one ensemble strategy was called "parameter ensemble", i.e., averaging the predictions over the last 10 epochs. The other ensemble strategy was called "parameter+NN ensemble", that is, averaging predictions over the last 10 epochs of three networks with different hyper-parameters (i.e., the sizes of the last fully-connected layers).

For MONN and DeepDTA (Öztürk et al., 2018), we used the same training and test sets as provided by DeepAffinity (Karimi et al., 2019), but dropped out a small number of samples (49 out of 263,583 training samples and 26 out of 113,168 test samples, about 0.02%), since these SMILES strings cannot be converted into a valid molecular graph by RDKit (Landrum, 2006). Followed the same strategy as used in DeepAffinity (Karimi et al., 2019), 10% of training data were used as the validation set. As we used this validation set to select the best number of epochs for MONN and DeepDTA, the "parameter ensemble" strategy is not suitable for these two models. So we directly use 30 ensemble models for MONN and DeepDTA (the same number of DeepAffinity predictions in their "parameter+NN ensemble" setting). That is, the predictions of 30 models were calculated and averaged as the ensemble prediction.

Here, we did not specifically optimize the hyper-parameters of MONN and DeepDTA over the BindingDB dataset, and directly used the hyper-parameter settings derived previously from the PDBbind-derived benchmark dataset.

DATA AND CODE AVAILABILITY

The benchmark dataset created in this work and the source code of the MONN model can be downloaded from https://github.com/lishuya17/MONN.