CPGL: Prediction of Compound-Protein Interaction by Integrating Graph Attention Network With Long Short-Term Memory Neural Network

Minghua Zhao, Min Yuan, Yaning Yang, and Steven X Xu

Abstract—Recent advancements of artificial intelligence based on deep learning algorithms have made it possible to computationally predict compound-protein interaction (CPI) without conducting laboratory experiments. In this manuscript, we integrated a graph attention network (GAT) for compounds and a long short-term memory neural network (LSTM) for proteins, used end-to-end representation learning for both compounds and proteins, and proposed a deep learning algorithm, CPGL (CPI with GAT and LSTM) to optimize the feature extraction from compounds and proteins and to improve the model robustness and generalizability. CPGL demonstrated an excellent predictive performance and outperforms recently reported deep learning models. Based on 3 public CPI datasets, C.elegans, Human and BindingDB, CPGL represented 1 - 5% improvement compared to existing deep-learning models. Our method also achieves excellent results on datasets with imbalanced positive and negative proportions constructed based on the C.elegans and Human datasets. More importantly, using 2 label reversal datasets, GPCR and Kinase, CPGL showed superior performance compared to other existing deep learning models. The AUC were substantially improved by 20% on the Kinase dataset, indicative of the robustness and generalizability of CPGL.

Index Terms—Deep learning, graph attention, LSTM, sequence analysis, compound-protein interaction

1 Introduction

Binding affinity is a parameter describing interaction of a drug with a target protein, and can be used to predict in vivo pharmacological effects (efficacy or/and safety) of drug candidates [1], [2], [3]. In drug discovery, potential drug candidates are often screened for binding affinity or compound-protein interaction (CPI). However, in vitro techniques measuring binding affinity in laboratories are usually expensive and time-consuming [4].

Recent advancements of artificial intelligence based on deep learning algorithms have made it possible to computationally predict CPI without conducting laboratory experiments [5]. Convolutional neural network (CNN) based models have been developed to extract features of compounds and proteins [6], [7], [8]. To improve the learning of the representation of drugs, graph neural networks (GNNs) and graph CNNs [9] (GCNs) have been proposed to extract structural features from the 2-D structure of compounds [10], [11], [12]. In addition, recurrent neural networks (RNN) were also used for feature extraction from compounds and proteins [13], [14], [15], [16].

Most existing methods use fix input features (i.e., binary values) for compound embedding, which however, often leads to poor performance for unbalanced datasets [12]. The 2D structure of small molecule compounds (i.e., atoms and chemical bonds) can be naturally viewed as a graph structure, where each node corresponds to an atom, and each edge corresponds to a chemical bond between two atoms. It is considered feasible to obtain the structural information of compounds by image recognition [17]. RD-Kit [18] was used to generate graph structures from SMILES of compounds due to its simplicity and accuracy in recovering graph structure. Bert-type method is also widely used in dealing with atomic embedding problem [19], [20]. Furthermore, coupling subgraph embeddings with end-to-end representation learning allows extracting input features adaptively during the training of the deep-learning model and has been shown to be able to achieve more robust performance [12].

CNN designed for image data usually does not work well for molecule graph data without a euclidean structure [21]. Although GCN has also been commonly utilized for atomic 2-D structures [10] and graph encoder-decoder architecture [22], [23], it cannot assign larger weights to important nodes as a non-parametric approach is used in GCN for weight allocation to neighboring nodes [24]. To

- *Minghua Zhao and Yaning Yang are with the Department of Statistics and Finance, University of Science and Technology of China, Hefei 230026, China. E-mail: zmh07@mail.ustc.edu.cn, ynyang@ustc.edu.cn.*
- *Min Yuan is with the School of Public Health Administration, Anhui Medical University, Hefei 230032, China. E-mail: myuan@ustc.edu.cn.*
- *Steven X Xu is with Genmab US, Inc., Princeton, NJ 08540 USA. E-mail: sxu@genmab.com.*

Manuscript received 24 April 2022; revised 11 October 2022; accepted 23 November 2022. Date of publication 29 November 2022; date of current version 5 June 2023.
The work of Minghua Zhao and Yaning Yang were supported by the National Science Foundation of China (NSFC) under Grant 11671375. The work of Min Yuan, was supported by the Natural Science Foundation of Anhui Province under Grant 2008085MA09.
(Corresponding authors: Yaning Yang and Steven X Xu.)
Digital Object Identifier no. 10.1109/TCBB.2022.3225296

solve this problem, graph attention network (GAT) is designed to process information from graph structures and can learn the strength of the connection (i.e., chemical bonds) between different nodes (i.e., atoms) at the same time. Therefore, GAT can markedly improve feature extraction for small molecule compounds.

Long-short term memory (LSTM) is a recurrent neural network designed to process sequence information [25], [26]. Compared with CNN, which has been widely used to process protein sequence information in previous methods [6], [7], [8], LSTM has been shown to better capture the large gap between the temporal information of input data and the relevant input [27]. That is, LSTM is capable of learning the relationship between words that are far apart in the sequence. This unique utility of LSTM may allow better extraction of spatial features of protein structure. Conversely, CNN based approaches may only process adjacent amino acids in the sequence for feature extraction.

The features of compounds and proteins are usually merged using concatenation [7], [15]. Tsubaki et.al. [12] proposed to use the importance weights of subsequences in a protein and a compound via an attention mechanism to fuse the features. In addition, two-sided attention mechanism incorporates the interaction of protein subsequences with compound substructures, where an attention weight is calculated for each the compound-protein substructure pair, and represents the degree of the binding between them [13]. The attention mechanism allows detecting the critical subsequences of the protein and substructures of the compound, thereby improving CPI prediction.

In this manuscript, we proposed CPGL (Compound-protein interaction prediction with graph neural network and long short-term memory neural network) to optimize the feature extraction from compounds and proteins by integrating the GAT for compound structures with LSTM for protein representation and fuse the feature of compounds and proteins through two-sided attention mechanism. We demonstrated that, via coupling with end-to-end learning, CPGL could improve the prediction of CPI on multiple datasets. More importantly, CPGL markedly improved the generalizability of model prediction, and could provide much greater prediction accuracy on label-reversal datasets compared to other recently published deep-learning algorithms (e.g., as GCN [9], TransformerCPI [10], GraphDTA [11], and CPI-GNN [12], etc).

2 Methods

The proposed CPGL method is mainly composed of two deep learning models: the GAT model for molecular graph and the bi-directional LSTM for protein sequence (see Fig. 1 A for an illustration of the CPGL method). We will first review these two models (GAT and LSTM) in the two subsequent subsections and describe the CPGL model in the compound-protein interaction prediction subsection. Training method of the proposed model is briefly described in the modeling training and validation subsection.

2.1 Graph Attention Network for Molecular Graph

We denote a molecular graph by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices representing the atoms in the molecular and $\mathcal{E}$ is the set of edges representing the chemical bonds in the molecular. Let v_i be the i-th atom in $\mathcal{V}$ and e_{ij} be the chemical bond between the i-th atom and the j-th atom. We use r-radius subgraph [28] to reinforce representation learning. More precisely, we denote $\mathcal{V}_i^r$ to be the subgraph centered at the i-th atom with the atoms linked with atom i by at most r chemical bonds and the chemical bonds linking them, and $\mathcal{E}_{ij}^r$ to be the subgraph centered at chemical bond e_{ij} with the atoms connected with e_{ij} by at most r-1 chemical bonds and the chemical bonds linking them.

As shown in Fig. 1 B, by using the toolkit RD-Kit [18], we convert SMILES to r-radius subgraphs sequence. After random initialization, we represent $\boldsymbol{v}_i^{(t)} \in \mathbb{R}^d$ as the $\mathcal{V}_i^r$ embedding at iteration t. Denote $\boldsymbol{A}$ as the adjacency matrix of the whole graph. If the ith atom is connected to the jth atom, $\boldsymbol{A}_{ij} = 1$, otherwise $\boldsymbol{A}_{ij} = 0$. We next introduce the transition function for atoms and chemical bonds.

Assume that there are n atoms in a molecule. For i, j=1,...,n, as shown in Fig. 1 C, we use the same notation in Transformer [29] to represent the three inputs of the graph attention network, namely, the keys, the values and the queries, and calculate the graph attention weights as follows:

$$\begin{aligned}
\boldsymbol{h}_i^{key(t)} &= f(\boldsymbol{W}_{key}\boldsymbol{v}_i^{(t)} + \boldsymbol{b}_{key});\\
\boldsymbol{h}_i^{query(t)} &= f(\boldsymbol{W}_{query}\boldsymbol{v}_i^{(t)} + \boldsymbol{b}_{query});\\
\boldsymbol{h}_i^{value(t)} &= f(\boldsymbol{W}_{value}\boldsymbol{v}_i^{(t)} + \boldsymbol{b}_{value});\\
\boldsymbol{\alpha}_{ij}^{(t)} &= \sigma(\boldsymbol{h}_i^{key(t)T}\boldsymbol{h}_i^{query(t)}),
\end{aligned} \tag{1}$$

where f, σ are non-linear activation functions such as ReLU [30] or tanh function, $\boldsymbol{W}_{key}, \boldsymbol{W}_{query}, \boldsymbol{W}_{value} \in \mathbb{R}^{d\times d}$ are the weight matrices, and $\boldsymbol{b}_{key}, \boldsymbol{b}_{query}, \boldsymbol{b}_{value} \in \mathbb{R}^d$ are the bias vectors. The attention weights $\boldsymbol{\alpha}_{ij}^{(t)}$ are used to choose the direction of information enrichment. Inspired by electron cloud distribution of molecules, the density difference of electron cloud distribution between atoms is described through the attention matrix.

Then we update the whole subgraph embedding vector sequence as follows:

$$\boldsymbol{v}_i^{(t+1)} = \sigma\left(\boldsymbol{v}_i^{(t)} + \sum_{j:\boldsymbol{A}_{ij}=1} (\boldsymbol{\alpha}_{ij}^{(t)}\boldsymbol{h}_i^{value(t)})\right). \tag{2}$$

Thus, with the transition function (1) and (2) of graph attention network, atoms can continuously obtain information about the surrounding atoms. After several iterations, atoms can obtain more global information. As a result, we get compound vector sequence $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ as the output of the molecule network.

2.2 Bi-Directional LSTM for Protein Sequence

The LSTM model is a deep neural network designed to capture long-term dependence structure in natural language context. Since amino acids that are close in space may be far away in terms of text sequence representation, we choose to capture the spatial structure by applying the LSTM to process amino acid sequence information. Since the direction of an amino acid sequence is irrelevant in compound-protein interaction, we use the bi-directional LSTM model in our study.

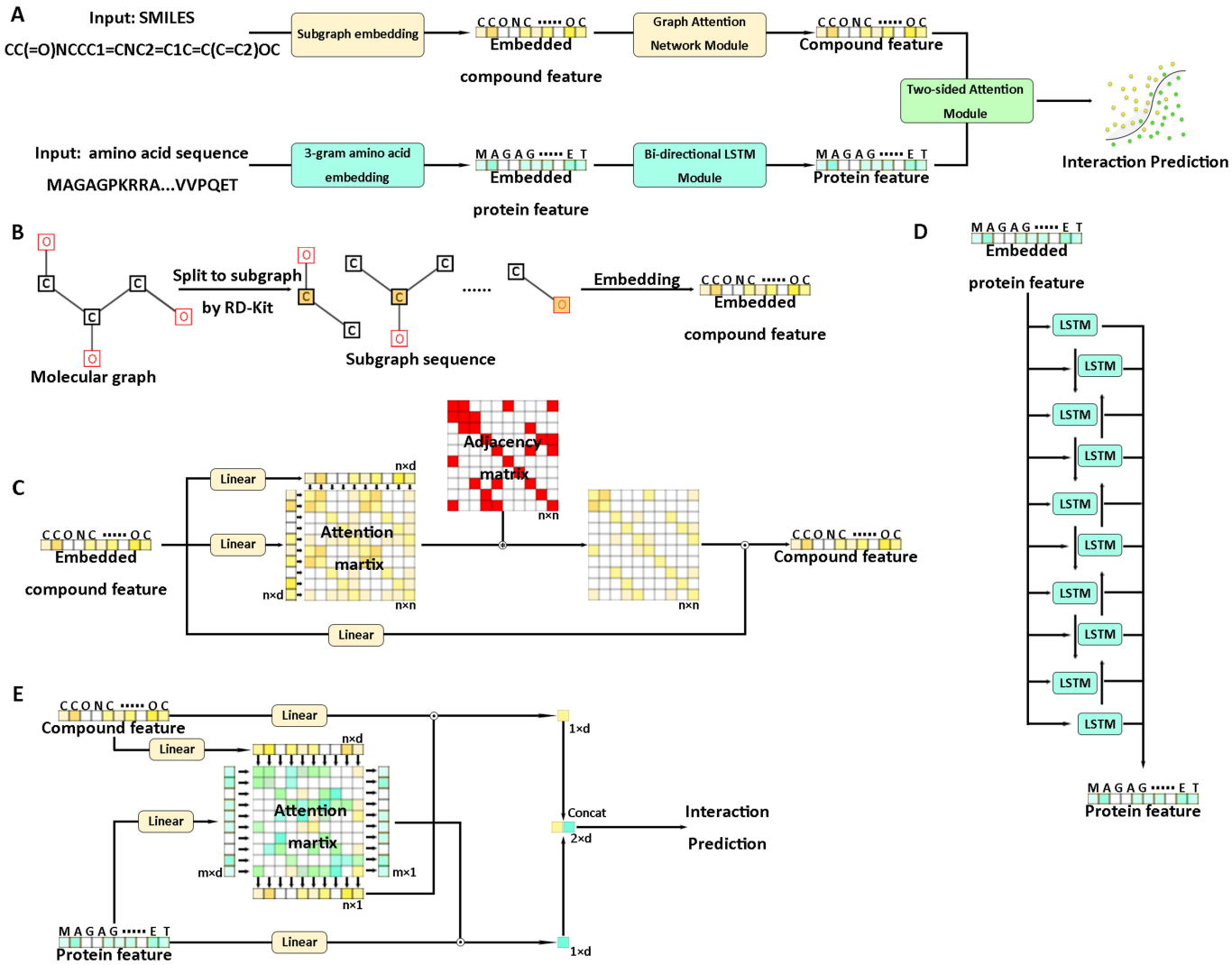


Fig. 1. (A) An overview of the CPGL. SMILES of small-molecular compounds are used as input and converted into subgraph sequences by RD-Kit, while for proteins, amino acid sequences are used as input and protein sequences are split based on k-gram amino acids. After the embedding layer, the compound and protein vectors, which are obtained using a GAT and a LSTM, respectively, are concatenated through a two-sided attention mechanism, and used as the input for a classifier to predict the compound-protein interaction. (B) The subgraph embedding module for SMILES of compounds. (C) The graph attention module for subgraph sequences to extract compound features. (D) The bi-directional LSTM module for k-gram amino acid sequences to extract protein features. (E) The two-sided attention mechanism for compound features and protein features. (d is the vector dimension, n is the length of the atom and m is the length of the amino acid sequence).

Before applying the bi-directional LSTM to proteins, we first use word2vec algorithm to embed the amino acid sequence into real-valued vectors [31], [32], [33], [34]. We then split the amino acid sequences into overlapping pieces or "words" of k-gram amino acids [35]. In this study, we set the length of the word to be $k = 3$. For example, we split *MAAVRM...LDLK* into *"MAA,"" AAV,"..., "DLK"*. Given an k-gram amino acid sequence, we embed them with word2vec into a vector sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ as the input of the bi-directional LSTM, where m is the length of amino acid sequence and $\boldsymbol{x}_i \in \mathbb{R}^d$ is the embedding of the i-th word obtained by the pretraining approach word2vec [36], [37]. Using the embedded features of protein sequence as the input of the bi-directional LSTM, each LSTM cell encodesthe short-term and long-term dependencies observed up to that cell's input (Fig. 1 D). The final feature vector were obtained by concatenating the outputs of the LSTM layers. The output of the concatenation layer for protein is shown as $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_m$, where $\boldsymbol{s}_i \in \mathbb{R}^{2\,d}$.

2.3 Two-Sided Attention Mechanism

Given compound vector sequence $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n)$ and protein vector sequence $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_m)$, as shown in Fig. 1 E, we introduce the attention matrix between compound vector sequence and protein vector sequence as follows:

$$\begin{aligned} \boldsymbol{h}_i &= f(\boldsymbol{W}_c \boldsymbol{v}_i + \boldsymbol{b}_c); \\ \boldsymbol{g}_j &= f(\boldsymbol{W}_p \boldsymbol{s}_j + \boldsymbol{b}_p); \\ \boldsymbol{\gamma}_{ij} &= \sigma(\boldsymbol{h}_i^T \boldsymbol{g}_j), \end{aligned} \tag{3}$$

where $\boldsymbol{W}_c \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}_p \in \mathbb{R}^{d \times 2\,d}$ are the weight matrices and $\boldsymbol{b}_c$, $\boldsymbol{b}_p \in \mathbb{R}^d$ are the bias vectors. The weight $\boldsymbol{\gamma}_{ij}$ represents attention, i.e., the degree of interaction between a compound subgraph and a protein subsequence. The weighted

TABLE 1
Hyper-Parameters of CPGL

Hyper-parameter Value	order
Radius(r)	0,1,2
Vector dimension(d)	16,32,64,128
Number of GAT layer	3
Number of output layer	3
Regularization	1e-5,1e-6,1e-7
Batch size	1
n-folds validation	5

sum of $\boldsymbol{h}_i$ and $\boldsymbol{g}_j$ is obtained using the attention weights

$$\boldsymbol{y}_c = \sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{\gamma_{ij}}\boldsymbol{h_i} = \sum_{i=1}^{n}\boldsymbol{\gamma_{i+}}\boldsymbol{h_i};$$
$$\boldsymbol{y}_p = \sum_{j=1}^{m}\sum_{i=1}^{n}\boldsymbol{\gamma_{ij}}\boldsymbol{g_j} = \sum_{j=1}^{m}\boldsymbol{\gamma_{+j}}\boldsymbol{g_j}. \quad (4)$$

The two-sided attention mechanism can give different weights to different parts of the compound and protein, which allows us to pay more attention to the parts that have a greater impact in the binding. This allows us to model the interaction between the compound and the protein instead of obtaining a simple summation.

2.4 Compound-Protein Interaction Prediction

To predict the interaction, we concatenate $\boldsymbol{y}_c$ and $\boldsymbol{y}_p$ and obtain an full-connected network as follows:

$$\boldsymbol{y} = \boldsymbol{W}[\boldsymbol{y}_c; \boldsymbol{y}_p] + \boldsymbol{b}, \quad (5)$$

where $\boldsymbol{W} \in \mathbb{R}^{2\times 2\,d}$ is the weight matrix, $\boldsymbol{b} \in \mathbb{R}^2$ is the bias vector. Finally, we apply a softmax function on the final output vector $\boldsymbol{y} = [y_0, y_1]$ to compute the probability of class of interaction (1 for interaction, 0 for no interaction) as follows:

$$p_t = \frac{exp(y_t)}{exp(y_0) + exp(y_1)}, \quad (6)$$

where $t \in \{0, 1\}$ is the binary label and p_t is the probability output from softmax function for interaction class t.

2.5 Modeling Training and Validation

To evaluate the performance of the algorithm and increase its stability, we used cross-validation. We selected a part of the data as the unseen test set, and the rest was used as the training set for 5-fold cross-validation, where one fold is used as the validation set and the rest is used as the training set. The final model was the average of all these models. All the settings and hyper-parameters of GPCL are summarized in Table 1.

With known compound-protein pairs and the interaction in the training dataset, we use the $L2$-penalized cross-entropy as loss function

$$l(\theta) = -\sum_{i=1}^{N} log(p_{t_i}) + \lambda||\theta||_2^2, \quad (7)$$

where θ is the set of all parameters in our model, N is the number of compound-protein pairs in the training dataset, t_i is the interaction of the i-th pair, and λ is the hyper-parameter of $L2$ penalty. We used backpropagation algorithm to train our model.

GPCL was implemented with pytorch 1.7.1 and we used the LookAhead [38] optimizer combined with RAdam [39], which does not suffer from the divergence problems of Adam optimizer without the learning rate warmup [10].

3 Materials

3.1 Public Datasets

We used three public datasets, for human, C.elegans [40] and BindingDB [41] to compare the performance of our model with other machine learning and deep learning-based approaches. Positive samples with CPI of human dataset and C.elegans dataset were obtained from two manually curated databases: DrugBank 4.1 [42] and Matador [43]. Validated negative samples of compound-protein pairs are also included in the two dataset [40]. In total, 3,369 positive interactions among 1,052 unique compounds and 852 unique proteins are present in the human dataset, while 4,000 positive interactions among 1,434 unique compounds and 2,504 unique proteins are included in the C.elegans dataset. Since most negative protein-compound pairs that have no CPI have not been experimentally validated or reported in the literature, we used the validated negative samples screened by a computer program in Liu. et.al 2015 [40]. The rationale behind this method is that proteins that are dissimilar to the known target of a given compound are unlikely to be the target of the compound. BindingDB dataset contains 39,747 positive examples and 31,218 negative examples with 49,745 unique compounds and 812 unique proteins from a public database [41].

Since the proportion of positive and negative samples in real data is often unbalanced, we also synthesized unbalanced datasets from the human and C.elegans datasets to evaluate the robustness of the model. The ratio of positive to negative samples was set to be 1,3 and 5, respectively, and the training, validating and testing sets were randomly partitioned [12]. The training and testing data sets of BindingDB were carefully designed so that they have no common ligands or protein in CPI pairs. Therefore, BindingDB dataset can assess models' generalization ability to unknown ligands and proteins.

3.2 Label Reversal Datasets

Label reversal datasets were proposed to evaluate the robustness and generalizability of deep learning models [10], where a ligand in the training set appears only in one class of interaction (either positive or negative interaction pairs), whereas, in the test set, the same ligand appears only in the opposite class of interaction. Two label reversal datasets,GPCR and Kinase, created by Chen et al. 2020 were used in this analysis to further evaluate model performance. Table 2 summarizes all datasets we use.

4 Results and Discussion

4.1 Performance on the Balanced Public Datasets

We compared CPGL with existing machine learning models (i.e., K nearest neighbors (KNN), random forest (RF), L2-

TABLE 2
Summary of Datasets

	Proteins	Compounds	Interactions	Poisitive	Negative
Human	852	1,052	6,728	3,369	3,359
C.elegans	2,504	1,434	7,786	4,000	3,786
BindingDB	812	49,745	70,965	39,747	31,218
GPCR	356	5,359	15,343	7,989	7,354
Kinase	229	1,644	111,237	23,190	88,047

logistic (L2), and support vector machines (SVM)), and recently published deep learning-based models i.e., GCN [9], TransformerCPI [10], GraphDTA [11] (by tailoring the last layer to binary classification task), CPI-GNN [12], DrugVQA [16] on human and C.elegans (Tables 3 and 4) datasets. We followed the same training and evaluating strategies as CPI-GNN and repeated with ten different random seeds to evaluate CPGL. Area Under Receiver Operating Characteristic Curve (AUC), precision and recall of each model are compared.

For the C.elegans dataset, CPGL demonstrated an excellent performance, and outperformed all the studied existing machine learning and deep learning models in terms of all the 3 evaluation metrics (AUC, precision, and recall). The AUC of CPGL is 0.99, while the precision and recall of CPGL are both around 0.96. This represents approximately 0.6 - 0.8% improvement compared to the most recently published deep learning algorithm, TransformerCPI. CPGL demonstrates 2 - 5% improvement compared to other recently developed deep learning-based approaches.

In terms of the human dataset, CPGL improved the AUC by 1%, compared to TransformerCPI. Greater degree of improvements is observed compared to other deep learning algorithms. In general, for both human and C.elegans datasets, the deep learning models had better performance compared to conventional machine learning based approaches. It should be noted that models based on 3D structural information of protein are out of scope for this manuscript as such information is not available for these two datasets. In general, for both human and C.elegans datasets, the deep learning models had better performance compared to conventional machine learning based approaches.

Since the traditional methods such as KNN,L2,SVM and RF are generally not comparable to various deep learning methods, we only compare our model with other deep learning methods for BindingDb dataset. Area Under Precision Recall Curve (PRC) and AUC of each model are shown in Table 5 and CPGL outperformed all the deep learning models in terms of AUC and PRC. The AUC of CPGL is 0.985 and the PRC is 0.983. This represents approximately 3.6% improvement compared with TransfomerCPI. We conclude that CPGL is still stable even in the face of unknown proteins and compounds. It should be noted that models based on 3D structural information of protein are out of scope for this manuscript as such information is not available for these three datasets.

TABLE 3
Comparison Results of the Proposed Model and Baselines on Human Dataset.

Method	AUC	Precision	Recall
KNN	0.860	0.927	0.798
RF	0.940	0.897	0.861
L2	0.911	0.913	0.867
SVM	0.910	**0.966**	**0.969**
GraphDTA	0.960±0.005	0.882±0.040	0.912±0.040
GCN	0.956±0.004	0.862±0.006	0.928±0.010
CPI-GNN	0.970	0.918	0.923
DrugVQA	0.964±0.005	0.897±0.004	0.948±0.003
TransformerCPI	0.973±0 . 002	0.916±0.006	0.925±0.006
CPGL	**0.979±0.001**	0.915±0.010	0.957±0.008

The performances of previous methods were all directly obtain from the TransformerCPI [10]

TABLE 4
Comparison Results of the Proposed Model and Baselines on C.elegans Dataset

Method	AUC	Precision	Recall
KNN	0.858	0.801	0.827
RF	0.902	0.821	0.844
L2	0.892	0.890	0.877
SVM	0.894	0.785	0.818
GraphDTA	0.974±0.004	0.927±0.015	0.912±0.023
GCN	0.975±0.004	0.921±0.008	0.927±0.006
CPI-GNN	0.978	0.938	0.929
TransformerCPI	0.988±0.002	0.952±0.006	0.953±0.005
CPGL	**0.990±0.001**	**0.956±0.003**	**0.957±0.005**

The performances of previous methods were all directly obtain from the TransformerCPI [10].

4.2 Performance on Unbalanced Public Datasets

Furthermore, we evaluated CPGL using unbalanced datasets (i.e., different ratios of positive to negative samples) which was synthesized based on public datasets, human and C.elegans, and compared CPGL with KNN, RF, L2,

TABLE 5
Comparison Results of the Proposed Model and Baselines on BindingDB Dataset

Method	AUC	PRC
GraphDTA	0.929	0.917
GCN	0.927	0.913
CPI_GNN	0.603	0.543
TransformerCPI	0.951	0.949
CPGL	0.985	0.983

The performances of previous methods were all directly obtain from the TransformerCPI [10].

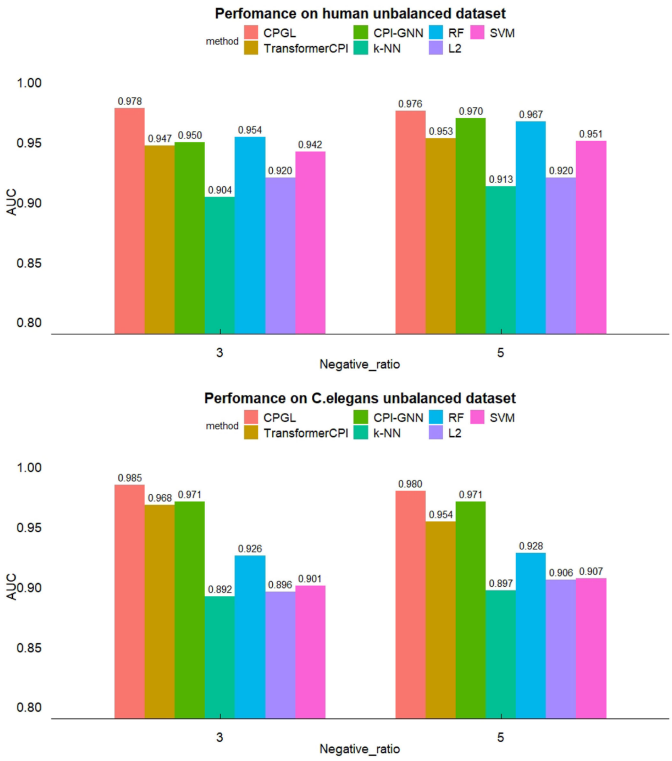


Fig. 2. The AUC scores of various methods on the human and C.elegans unbalanced datasets. Note that the AUC scores of CPI-GNN, KNN, RF, L2 and SVM are derived from CPI-GNN [12].

Fig. 3. Results of CPGL, TransformerCPI, CPI-GNN, GraphDTA and GCN on GPCR dataset and Kinase dataset. Note that the AUC and the PRC scores of TransformerCPI, CPI-GNN, GraphDTA and GCN are derived from CPI-GNN [10].

SVM, CPI-GNN and TransformerCPI using the unbalanced datasets. Fig. 2 showes the AUC scores on the human and C.elegans unbalanced datasets. It is apparent that CPGL achieved the best perdictive performance on all the synthesized data regardless of degree of unbalance. Compared to TransformerCPI and CPI-GNN, CPGL improved the AUC by 1.8 - 3.3% and 0.6 - 2.9%, respectively. The performance of traditional machine learning models such as SVM, L2, RF, and kNN was markedly lower compared to CPGL.

4.3 Performance on Label Reversal Datasets

We also used two label reversal datasets [10], GPCR and Kinase, to compare our proposed CPGL with CPI-GNN, GraphDTA, GCN and TrasformerCPI. As shown in Fig. 3, On the GPCR set, CPGL have achieved similar AUC (0.856 versus 0.863) but slightly better PRC (0.861 versus 0.859) compared to TransformerCPI and outperformed other method both in AUC and PRC. At the same time, our method has made substantial improvement in kinase dataset. Compared with TransformerCPI, the AUC and PRC of CPGL are increased by about 20% and 15%. Meanwhile, the AUC values of other methods are less than 0.5, which indicates that these methods have failed on the kinase dataset. It is also known that most ligands in Kinase dataset possess almost ten times more noninteraction pairs than interaction pairs [10], while most ligands in GPCR dataset has relatively balanced positive and negative samples. This showed that our method is not subject to the hidden ligand bias and

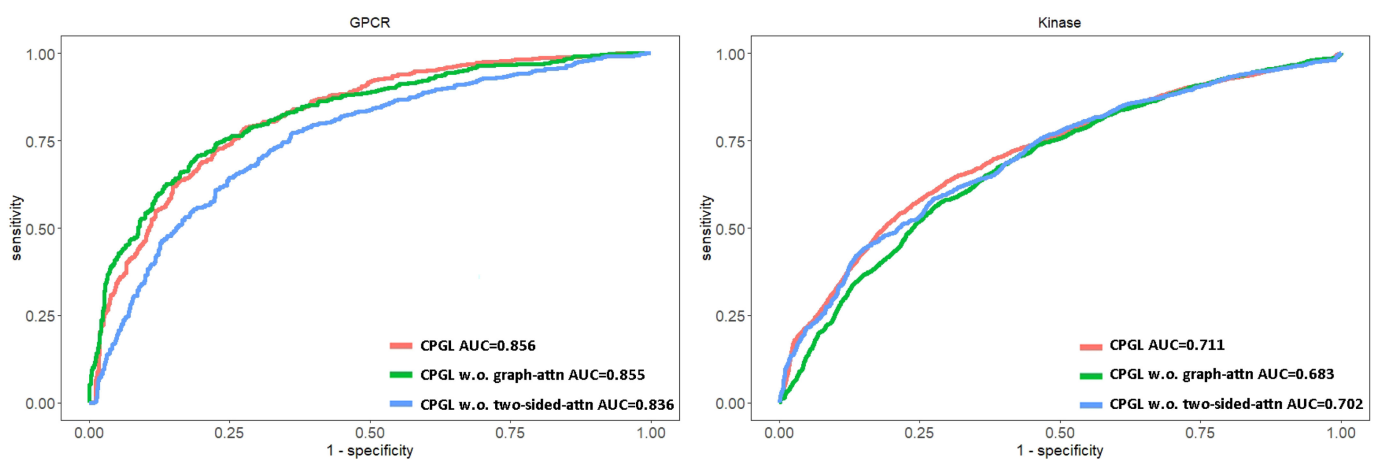


Fig. 4. ROC curves of CPGL for model ablation study.

possesses capability of learning interactions between proteins and compounds. Overall, the results on label reversal dataset show that CPGL is more robust under the label reversal datasets and unbalanced datasets.

4.4 Model Ablation Study

In order to assess if the two attention mechanisms we used in the model are necessary, we evaluated two reduced models by removing the graph attention (Fig. 1 C) or the two-sided attention component (Fig. 1 E) from CPGL respectively using the label reversal experiment. As shown in Fig. 4, although removing the component of graph attention had little impact on the model performance on the GPCR dataset, removing this component substantially decreased the prediction (AUC) by nearly 4% on kinase dataset. Taking out the two-sided attention component also produced suboptimal results, and the AUC decreased by 2.3% on GPCR dataset and 1.3% on kinase dataset. We argue that graph attention describes the strength of the bond between adjacent atoms, which includes not only the type of bond, but also the shift of the electron cloud caused by the structure of the compound. The two-sided attention mechanism learns the interactions between different atoms and different amino acids, which allows us to discover key parts of compounds and protein sequences. These all enhance the performance of our model. Overall, these results demonstrated that both graph attention and the two-sided attention components were necessary and enhanced the predictive performance of the model.

5 Conclusion and Future Work

We developed a deep-learning based algorithm, CPGL to optimize the feature extraction from compounds and proteins by integrating the GAT for compound structures with LSTM for protein representation. CPGL not only improved the prediction on multiple regular public datasets, but also substantially outperformed other reported deep-learning algorithms on unbalanced datasets and label-reversal datasets, indicative of the robustness and generalizability of CPGL.

With the emergence of AlphaFold [44], the 3D structures of proteins can be obtained by computational methods. Johansson-Akhe and Wallner [45] have proposed an approach to improve peptide-protein docking via AlphaFold-multimers. Wang et.al. [46] proposed to improve the accuracy of protein property prediction by combining protein sequence information and 3D structure information, which can correct the deviation between AlphaFold Protein Structure Database and experimentally obtained 3D structures. We believe that using AlphaFold predicted protein structure may further improve the predictive performance for compound-protein interaction. Future work is warranted.

Data and Code Availability

The data and source codes of CPGL are available on the GitHub repository at https://github.com/RobinDoyle/CPGL.

References

[1] M. J. Keiser et al., "Predicting new molecular targets for known drugs," *Nature*, vol. 462, pp. 175–181, 2009.

[2] E. Lounkine et al., "Large-scale prediction and testing of drug activity on side-effect targets," *Nature*, vol. 486, pp. 361–367, 2012.

[3] J. L. Medina-Franco, M. A. Giulianotti, G. S. Welmaker, and R. A. Houghten, "Shifting from the single to the multitarget paradigm in drug discovery," *Drug Discov. Today*, vol. 18, pp. 495–501, 2013.

[4] J. W. A. Scannell, H. BlanckleyBoldon, and B. Warrington, "Diagnosing the decline in pharmaceutical r&d efficiency," *Nature Rev. Drug. Discov.*, vol. 11, pp. 191–200, 2012.

[5] A. S. Rifaioglu et al., "Recent applications of deep learning and machine intelligence on in silico drug discovery: Methods, tools and databases," *Brief. Bioinf.*, vol. 20 no. 5, pp. 1878–1912, 2019.

[6] I. Lee, J. Keum, and H. Nam, "DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences," *PLoS Comput. Biol.*, vol. 15, 2019, Art. no. e1007129.

[7] H. Ozturk, A. Ozgur, and E. Ozkirimli, "DeepDTA: Deep drug-target binding affinity prediction," *Bioinformatics*, vol. 34, pp. i821–i829, 2018.

[8] H. Ozturk, E. Ozkirimli, and A. Ozgur, "WideDTA: Prediction of drug-target binding affinity," 2019, *arXiv:1902.04166*.

[9] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[10] L. Chen et al., "TransformerCPI: Improving compound-protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments," *Bioinformatics*, vol. 36 no. 16, pp. 4406–4414, 2020.

[11] T. Nguyen, H. Le, and S. Venkatesh, "GraphDTA: Prediction of drug-target binding affinity using graph convolutional networks," 2019.

[12] M. Tsubaki, K. Tomii, and J. Sese, "Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences," *Bioinformatics*, vol. 35, pp. 309–318, 2019.

[13] K. Abbasi et al., "DeepCDA: Deep cross-domain compound-protein affinity prediction through LSTM and convolutional neural networks," *Bioinformatics*, vol. 36 no. 17, pp. 4633–4642, 2020.

[14] K. Gao et al., "Interpretable drug target prediction using deep neural representation," In *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3371–3377.

[15] M. Karimi, D. Wu, Z. Wang, and Y. Shen, "DeepAffinity: Interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks," *Bioinformatics*, vol. 35, pp. 3329–3338, 2019.

[16] S. Zheng, Y. Li, S. Chen, J. Xu, and Y. Yang, "Predicting drug-protein interaction using quasi-visual question answering system," *Nature Mach. Intell.*, vol. 2, pp. 134–140, 2020.

[17] X. Zhang et al., "ABC-Net: A divide-and-conquer based deep learning architecture for SMILES recognition from molecular images," *Brief. Bioinf.*, vol. 23, no. 2, Mar. 2022, Art. no. bbac033.

[18] G. Landrum, RDKit documentation, Sep. 01, 2015. [Online]. Available: http://www.rdkit.org

[19] S. Wang, Y. Guo, Y. Wang, H. Sun, and J. Huang, "SMILES-BERT: Large scale unsupervised pre-training for molecular property prediction," in *Proc. 10th ACM Int. Conf. Bioinf., Comput. Biol. Health Inform.*, 2019, pp. 429–436.

[20] X. Zhang et al., "MG-BERT: Leveraging unsupervised atomic representation learning for molecular property prediction," *Brief. Bioinf.*, vol. 22, no. 6, Nov. 2021, Art. no. bbab152.

[21] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[22] W. Wang, X. Yang, C. Wu, and C. Yang, "CGINet: Graph convolutional network-based model for identifying chemical-gene interaction in an integrated multi-relational graph," *BMC Bioinf.*, vol. 21, 2020, Art. no. 544.

[23] X. Yang et al., "BioNet: A large-scale and heterogeneous biological network model for interaction prediction with graph convolution," *Brief. Bioinf.*, vol. 23, no. 1, Jan. 2022, Art. no. bbab491.

[24] Z. Wu et al., "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

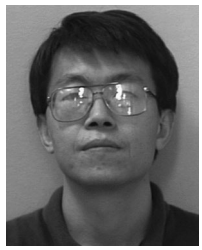
[25] A. Graves, "Long short-term memory," in *Supervised Sequence Labelling With Recurrent Neural Networks*, Berlin, Heidelberg: Springer, 2012, pp. 37–45.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9 no. 8, pp. 1735–1780, 1997.

[27] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31 no. 7, pp. 1235–1270, 2019.
[28] F. Costa and K. D. Grave, "Fast neighborhood subgraph pairwise distance kernel," in *Proc. Int. Conf. Mach. Learn.*, 2020, Art. no. e0141287.
[29] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[31] D. Kimothi, A. Soni, P. Biyani, and J. M. Hogan, "Distributed representations for biological sequence analysis," in, 2016, *arXiv:1609.05949*.
[32] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS One*, vol. 10, no. 11, 2015, Art. no. *e0141287*.
[33] C. Mazzaferro et al., "Predicting protein binding affinity with word embeddings and recurrent neural networks," 2017.
[34] K. K. Yang, Z. Wu, C. N. Bedbrook, and F. H. Arnold, "Learned protein embeddings for machine learning," *Bioinformatics*, vol. 34, pp. 2642–2648, 2018.
[35] Q.-W. Dong, X. Wang, and L. Lin, "Application of latent semantic analysis to protein remote homology detection," *Bioinformatics*, vol. 22, pp. 285–290, 2006.
[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[37] T. Mikolov, H. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Adv. Neural Inf. Process. Syst.*, vol. 26, pp. 3111–3119, 2013.
[38] M. R. Zhang, J. Lucas, G. Hinton, and J. Ba, "Lookahead optimizer: K steps forward, 1 step back," 2019, *arXiv:1907.08610*.
[39] L. Liu et al., "On the variance of the adaptive learning rate and beyond," 2019, *arXiv:1908.03265*.
[40] H. Liu, J. Sun, J. Guan, J. Zheng, and S. Zhou, "Improving compound-protein interaction prediction by building up highly credible negative samples," *Bioinformatics*, vol. 31, pp. i221–i229, 2015.
[41] M. K. Gilson et al., "BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology," *Nucleic Acids Res.*, vol. 44, pp. 1045–1053, 2016.
[42] D. S. Wishart et al., "DrugBank: A knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Res.*, vol. 36, pp. D901–D906, 2008.
[43] S. Gunther et al., "Supertarget and matador: Resources for exploring drug-target relationships," *Nucleic Acids Res*, vol. 36, pp. D919–D922, 2008.
[44] J. Jumper et al., "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, pp. 583–589, 2021.
[45] I. Johansson-Akhe and B. Wallner, "Improving peptide-protein docking with AlphaFold-Multimer using forced sampling," 2022.
[46] Z. Wang et al., "LM-GVP: An extensible sequence and structure informed deep learning framework for protein property prediction," *Sci. Rep.*, vol. 12, 2022, Art. no. 6832.

Minghua Zhao is currently working toward the PhD degree with the Department of Statistics and Finance, University of Science and Technology of China (Hefei). His main research interests are in the areas of deep learning and biostatistics.

Min Yuan received the PhD degree from the University of Science and Technology of China, Hefei. She is currently a professor with the School of Public Health Administration, Anhui Medical University (Hefei). Her main research interests include the areas of genome wide association study of Alzheimer's disease, longitudinal data analysis and statistical models and applications in public health and biomedicine.



Yaning Yang is currently a professor with the Department of Statistics and Finance, University of Science and Technology of China (Hefei). His main research interests include genome wide association study, longitudinal data analysis and deep learning.

Steven X Xu is a senior director of clinical pharmacology & quantitative science with Genmab Inc. He has more than 20 years' experience in drug development in oncology, immunology, cardiovascular/metabolism, neurology, and infectious diseases at Bristol-Myers Squibb, Johnson & Johnson, and Genmab Inc and extensive knowledge in clinical, clinical pharmacology, drug discovery, translation research, therapeutic antibody research, statistics, deep learning, and artificial intelligence.