# ParaCPI: A Parallel Graph Convolutional Network for Compound-Protein Interaction Prediction

Longxin Zhang, Wenliang Zeng, Jingsheng Chen, Jianguo Chen, and Keqin Li

***Abstract*—Identifying compound-protein interactions (CPIs) is critical in drug discovery, as accurate prediction of CPIs can remarkably reduce the time and cost of new drug development. The rapid growth of existing biological knowledge has opened up possibilities for leveraging known biological knowledge to predict unknown CPIs. However, existing CPI prediction models still fall short of meeting the needs of practical drug discovery applications. A novel parallel graph convolutional network model for CPI prediction (ParaCPI) is proposed in this study. This model constructs feature representation of compounds using a unique approach to predict unknown CPIs from known CPI data more effectively. Experiments are conducted on five public datasets, and the results are compared with current state-of-the-art (SOTA) models under three different experimental settings to evaluate the model's performance. In the three cold-start settings, ParaCPI achieves an average performance gain of 26.75%, 23.84%, and 14.68% in terms of area under the curve compared with the other SOTA models. In addition, the results of the experiments in the case study show ParaCPI's superior ability to predict unknown CPIs based on known data, with higher accuracy and stronger generalization compared with the SOTA models. Researchers can leverage ParaCPI to accelerate the drug discovery process.**

***Index Terms*—Cold-start settings, compound-protein interaction, drug discovery, parallel graph convolutional network.**

## I. Introduction

IDENTIFYING compound-protein interactions (CPIs) is a critical step in compound screening, lead discovery, and holds important application value in drug discovery and design. Accurate CPI prediction, also known as drug-target interaction (DTI) prediction, reduces the cost and time of drug discovery and enhances the success rate of new drug development. Although high-throughput screening [1] remains a reliable method for detecting interactions between compounds and proteins, building large compound screening libraries in practical applications is difficult [2]. In vitro tests to discover CPIs remain expensive, time consuming, and laborious, limiting their scalability [3]. Each new compound approved by the US Food and Drug Administration to enter the market has an average capital cost of billions of dollars during its development process, which typically lasts for more than a decade [4]. Over the past few decades, numerous biologists and pharmacologists have devoted considerable effort to CPI prediction. Today, researchers utilize efficient computational methods to analyze CPI data accumulated in past studies, aiming to improve the accuracy of CPI forecasts. Such computational methods have the potential to dramatically decrease the time and cost of experimental drug testing while providing scientists with reliable evidence to accelerate drug discovery by identifying more precise candidate targets.

Three existing computational methods for CPI prediction include molecular docking-based, machine learning-based, and deep learning-based approaches. The molecular docking-based computational approach involves mapping different conformations of a compound (conformations refer to the different arrangements of molecules in space) onto the 3D structure of a protein and then calculating the energy between them to predict their CPI. However, this method suffers from two drawbacks: (1) accurate modeling of 3D structures of proteins and compounds requires high computational costs, and obtaining 3D structures of some proteins is difficult [5], greatly limiting the application range of molecular docking; (2) the multistep process of molecular docking introduces potential errors at each step, leading to inaccurate final prediction results. The machine learning-based computational method directly utilizes compound and protein data to predict CPI, bypassing the need to study complex physicochemical properties as in the molecular docking method, thereby improving prediction accuracy [6]. For instance, Yamanishi et al. [7] proposed a supervised learning method called bipartite graph to infer the existence of CPI by synthesizing compound and protein information into the pharmacological space. Bleakley et al. [8] designed a bipartite local model method based on Ref. [7], which computes the similarity between compounds and proteins using kernel functions and employs a kernel-based support vector machine (SVM) for CPI prediction. To further enhance the accuracy of CPI prediction, Peng et al. [9] developed a prediction model named NormMulInf based on collaborative filtering theory, utilizing labeled and unlabeled interaction information. NormMulInf identifies similarity features by integrating biological information, such as the similarity of samples and the local correlation between

Manuscript received 13 August 2023; revised 3 April 2024; accepted 21 May 2024. Date of publication 24 May 2024; date of current version 9 October 2024. This work was supported in part by the Natural Science Foundation of Hunan Province under Grant 2023JJ50204, in part by the Scientific Research Foundation of Hunan Provincial Education Department, China under Grant 23B0560, in part by the National Key R&D Program of China under Grant 2018YFB1003401, and in part by the National Natural Science Foundation of China under Grant 61702178, Grant 62072172, Grant 62002110, and Grant 62372486. *(Corresponding authors: Longxin Zhang; Jianguo Chen.)*

Longxin Zhang, Wenliang Zeng, and Jingsheng Chen are with the College of Computer Science, Hunan University of Technology, Zhuzhou 412007, China (e-mail: longxinzhang@hut.edu.cn).

Jianguo Chen is with the School of Software Engineering, Sun Yat-Sen University, Zhuhai, Guangdong 519082, China (e-mail: chenjg33@mail.sysu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA.

Digital Object Identifier 10.1109/TCBB.2024.3404889

sample labels. These features are integrated into a robust principal component analysis model and solved using an enhanced Lagrange multiplier. In the same year, Zhang [10] developed an ensemble learning approach that integrates available heterogeneous data to predict CPIs using intrinsic associations of known interactions between compounds and proteins. This approach utilizes a stacked framework and employs an SVM classifier as a meta-learner, resulting in enhanced prediction outcomes. In response to the prohibitive costs and labor-intensive nature of traditional wet experiments, Yang et al. [11] introduced the BioNet model. This model adopts a deep biological network architecture, utilizing a graphical encoder-decoder design to glean insights into intricate interactions among chemicals, genes, diseases, and biological pathways via a graph convolution process. Aiming at the problems of high false positive and low accuracy rates in current DTI prediction methods, DTI-CDF [12] fuses various features of compounds and proteins to predict CPI by cascading a random subforest composed of multiple random trees. DTI-CDF exploits a hierarchical strategy where the output of each random forest (RF) layer serves as the input for the next RF layer. It also incorporates a negative sampling strategy to enhance the diversity of the training dataset, thereby improving the prediction performance.

Although machine learning has shown good performance in CPI prediction, the increasing size of biological datasets and the need for prior knowledge have made traditional methods less capable of handling massive data. To address these limitations, the industry and academia have turned their attention to deep learning methods, which have gained widespread application in several fields. One early CPI prediction model based on deep learning is DeepDTA [13], which adopts a “Y”-type prediction framework using two convolutional neural networks (CNNs) branches to encode the features of compounds and proteins. The two feature sets are then fed into a fully connected layer (FCL) for drug-target binding affinity (DTA) prediction. The DTA prediction problem is a regression problem which predicts the binding affinity between the drug and the target. Although CNN-based models have shown success in CPI prediction, they represent compounds as strings, which is not the natural representation of compounds. This representation may lead to the loss of critical chemical structure information during the process of extracting compound features, thereby affecting prediction performance. To address this limitation, researchers have explored alternative approaches, such as SPP-CPI [14], which uses a distance matrix to represent the compound and employs a feature pyramid network to extract potential features of compounds. For proteins, SPP-CPI utilizes natural language processing (NLP) methods to obtain the semantic information of the amino acid sequence. Another approach, DrugVQA [15], employs 2D distance maps for proteins and molecular linear symbols for drugs, along with a visual question-answering model for CPI prediction. DeepConv-DTI [16] recognizes that traditional protein descriptors may not provide sufficient information for accurate CPI prediction, and they capture local residue patterns of generalized protein classes by convolving amino acid subsequences of varying lengths. The experimental results demonstrate that DeepConv-DTI effectively enriches the features of the original protein sequences. HoTS [17] constructs a binding region (BR) dataset by collecting protein sequences of CPI complexes and binding sites, and the model is pretrained on this dataset to improve the interpretability of CPI prediction models. The pretrained model is then used to predict the DTI dataset. The model performs well in BR prediction on independent test datasets and accurately predicts DTI without using 3D structural information. However, the improvement in prediction performance by these methods is not obvious. Graph neural network (GNNs) have been applied to CPI prediction to extract graphical structural information of compounds and overcome these challenges. GNN-based methods represent compounds as molecular graphs, with atomic nodes as graph nodes and bond pairs between atoms as edges, and employs GNN models to obtain embedding representations of the molecular graphs. For example, Tsubaki et al. [18] developed an end-to-end prediction model called CPI_prediction by integrating discrete symbolic data for compounds and proteins, combining GNN and CNN, and simultaneously employing an attention mechanism for efficient visualization of CPI. Yang et al. [19] proposed MGraphDTA, a deep multiscale neural network model for DTA prediction, based on chemical intuition to address this problem. MGraphDTA employs an ultradeep GNN with 27 graph convolutional network (GCN) layers, utilizing dense connections in the GNN to simultaneously capture both the local and global structure of the compound. To improve model interpretability, MGraphDTA introduces a novel visual interpretation method called gradient-weighted affinity activation mapping, revealing the global relationships between atoms in a molecule. However, this method of constructing full graph features with ultradeep GNNs leads to considerable information redundancy, resulting in unsatisfactory prediction performance. Additionally, MGraphDTA fails to account for the model’s generalizability. In CPI prediction, accurate representations of atomic features can substantially enhance prediction accuracy. The CGINet [20] model takes a unique approach by focusing on subgraphs to streamline the feature learning process. Initially, node embeddings are generated through the learning of binary association subgraphs. These embeddings are then transitioned to subgraphs encompassing multiple interactions. This methodology enables the model to acquire a more concentrated high-level representation of the target nodes, resulting in more efficient prediction of interactions between chemicals and genes. With the widespread application of transformers in NLP and computer vision, researchers have found that the self-attention mechanism of the transformer architecture effectively captures global dependencies in long text sequences. Building on this insight, Chen et al. [21] proposed a new CPI prediction model, TransformerCPI, based on the transformer architecture. The authors highlighted specific challenge faced by sequence-based CPI prediction models, including the use of inappropriate datasets, hidden ligand bias, and inappropriately split datasets, which can lead to an overestimation of the model’s prediction performance. TransformerCPI addresses these traps by constructing new datasets tailored specifically for CPI prediction and introducing strict label reversal experiments to evaluate the model’s ability to learn true underlying features. Nevertheless, the self-attention

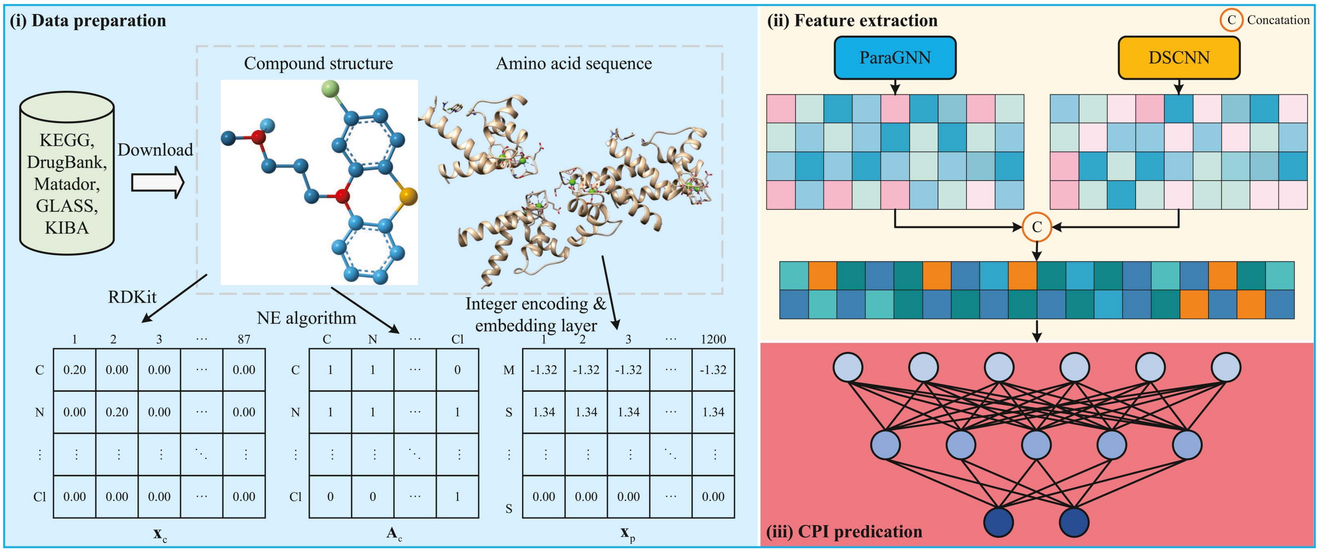


Fig. 1. Overall framework of ParaCPI.

mechanism in TransformerCPI significantly inflates the model parameters, leading to diminished efficiency during both training and testing phases. Similar to TransformerCPI, the MG-BERT [22] model integrates the local information propagation mechanism of GNNs with the advanced capabilities of BERT to enhance the efficiency of learning molecular graph structures. Through this approach, pretraining the MG-BERT model on a large-scale unlabeled dataset enables more effective extraction of contextual information from molecules. The atomic features produced by this method exhibit high sensitivity to context and are applicable to various intricate prediction tasks. Although all of the above models have achieved remarkable results in CPI prediction, further improvements in accuracy and generalization are needed to enhance the drug development process.

This study proposes a novel approach called parallel GNN for CPI prediction (ParaCPI) to address the above problems. We believe that the molecular graph representation of the compound remains the most efficient approach. Thus, ParaCPI uses the simplified molecular input line entry specification (SMILES) [23] of the compounds, and the amino acid sequences of the proteins are used as inputs to obtain their initial features. The parallel GNN (ParaGNN) and the deep separable CNN (DSCNN) are constructed in parallel to process the two initial feature sets and obtain the embedding representations of the compounds and proteins. ParaCPI concatenates the two sets of embedding representations to obtain potential features of compound-protein complexes (CPCs) and feeds them into a FCL for CPI prediction. The main contributions of this study are fourfold.

- A new neighborhood expanding (NE) algorithm is designed to generate an efficient expanded adjacency matrix (EAM) for aggregating the global feature structure of compounds.
- A unique GNN feature extraction network, ParaGNN, is constructed on the basis of EAM to extract compound features. Additionally, a more effective regularization strategy, half dropout (HD), is introduced to prevent the model from overfitting.
- A novel graph regularization block (GRB) is developed to adjust the model parameters and extract compound features that are more beneficial for predicting CPIs.
- Compared with the current state-of-the-art (SOTA) models, ParaCPI demonstrates strong competitive performance in terms of generalizability across various environments.

The rest of this study is organized as follows. Section II elaborates ParaCPI in detail, followed by the presentation of experimental results and discussions in Section III. Section IV concludes the study.

## II. Methods

Given a CPC composed of SMILES sequences and protein sequences, ParaCPI models the CPI problem by considering the global network structure of the compound and the local bioinformation of the CPC. Fig. 1 shows the overall framework of ParaCPI. In particular, ParaCPI is divided into three steps: (i) data preparation, which aims to initialize the biological information and obtain initial features of compounds and proteins; (ii) feature extraction, which aims to mine potential feature information in compounds and proteins to obtain features of the CPC; and (iii) CPI prediction, which feeds the CPC features into the FCL to predict the CPI. ParaCPI treats CPI prediction as a binary classification problem, defined as follows:

$$y = FCL\left([f_{\mathrm{c}}(\mathbf{x}_{\mathrm{c}}), f_{\mathrm{p}}(\mathbf{x}_{\mathrm{p}})]\right), \tag{1}$$

where $\mathbf{x}_{\mathrm{c}}$ and $\mathbf{x}_{\mathrm{p}}$ are the initial features of the compound and protein obtained after data processing. $f_{\mathrm{c}}(\cdot)$ and $f_{\mathrm{p}}(\cdot)$ denote the feature extraction functions of compounds and proteins, respectively. $FCL(\cdot)$ is the FCL. $y$ indicates whether an interaction occurs between the compound and the protein, with its value being 0 or 1. $y = 1$ indicates a positive interaction (i.e., yes), otherwise, it indicates no interaction.

TABLE I
ATOMIC FEATURES ON THE HUMAN AND C.ELEGANS DATASETS

| Name | Description (encoding method) | Dimension |
|---|---|---|
| Atom type | [B, Br, C, Ca, Cd, Cl, Co, Cr, Cu, F, Fe, Ge, H, Hg, I, In, K, Li, Mg, Mn, N, Na, Ni, O, P, Pd, Pt, S, Sb, Se, Si, Sn, Ti, Tl, V, Yb, Zn, Zr, other] (one-hot) | 44 |
| Degree | Number of covalent bonds [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] (one-hot) | 11 |
| Hydrogens | Number of connected hydrogens [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] (one-hot) | 11 |
| Implicit valence | Implicit valence of the atom [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] (one-hot) | 11 |
| Hybridization | [sp, sp2, sp3, sp3d, sp3d2, other] (one-hot) | 6 |
| Aromatic | Whether the atom is part of an aromatic system [0/1] (binary) | 1 |
| Chirality | Whether the atom is a chiral center [0/1] (binary) | 1 |
| Chirality type | [R, S] (one-hot) | 2 |

### A. Drug Preparation

*1) Compound Representation:* The sequence-based representation method greatly reduces the cost of expensive molecular docking simulations. In this study, compounds are represented using SMILES sequences, which are text symbols representing topological information based on chemical bond rules. ParaCPI captures the structural and topological information of molecules through SMILES sequences in a simple and intuitive manner. In addition, SMILES sequences are a standard representation that can be easily searched in chemical information systems and databases. ParaCPI utilizes the RDKit tool [24] to convert each compound into a molecular map, where the atoms and bonds of the compound are represented as vertices and edges in the map, respectively. Consequently, ParaCPI constructs a vector $\mathbf{x}_{\mathrm{c}}$ to represent the atomic features in the molecular map of a compound. These features include the type of atom, atomic degree, hydrogens, implicit valence, hybridization, aromaticity, chirality, and chirality type. For the Human and Caenorhabditis elegans (C.elegans) datasets, this approach represents the features of each compound in the dataset as a vector of size $1 \times 87$. The specific features are encoded, as shown in Table I.

When compounds are represented as molecular maps, GNNs can automatically extract potential chemical features by taking the structure of neighboring nodes into account and aggregating messages between the layers. GNNs typically adopt the adjacency matrix of the graph as the method of message passing and then obtain the full graph structure by continuously deepening the number of network layers. However, this approach cannot be stacked extremely deep, unlike CNNs. In cases where the number of network layers is insufficient, GNNs cannot fully leverage the subgraph structure information. ParaCPI introduces a new message passing method based on the adjacency matrix, named the NE algorithm, as shown in Algorithm 1.

For a $G = \langle V, E \rangle$, where $V$ denotes the set of nodes and $E$ denotes the set of edges. NE first constructs the adjacency matrix $\mathbf{A}$ based on the graph $G$ (lines 1-4). The $k_1$-th power adjacency matrix $\mathbf{A}^{k_1}$ and the $k_2$-th power adjacency matrix $\mathbf{A}^{k_2}$ of $\mathbf{A}$ are calculated (line 5). The EAM $\mathbf{A}_{\mathrm{c}}$ is calculated using (2) (line 6). The NE algorithm converts $\mathbf{A}_{\mathrm{c}}$ into a binary matrix to prevent repeated extraction of features from a node in the graph (lines 7-13). The time complexity of the NE algorithm is $O(N_V^2) = O(N_E) + O(N_V \times N_V) + O(k_1!) + O(k_2!)$, where $k_1$ and $k_2$ are positive integers no higher than 5. In this study, the SMILES representation O=C(C)Oc1 ccccc1C(=O)O of Aspirin is used as an example to construct the EAM, as shown in Fig. 2. The NE algorithm utilizes neighborhood expansion kernels with sizes of $k_1 = 4$ and $k_2 = 5$, which can cover all atoms of the aromatic hydrocarbon composed of C:4–C:9 in Aspirin. As depicted in Fig. 2, the EAM generated by the NE algorithm contains more node information. The binarization operation of the algorithm can improve the efficiency of GNN aggregation information while reducing the redundancy of the data, thereby obtaining

**Algorithm 1:** NE.

```
Input: Compound molecular map $G = (V, E)$,
  neighbourhood expansion kernel size $k_1$, $k_2$.
Output: EAM $\mathbf{A}_{\mathrm{c}}$.
1: for $(v_i, v_j) \in E$ do
2:    $\mathbf{A}[v_{\mathrm{i}}, v_j] \leftarrow 1$;
3:    $\mathbf{A}[v_j, v_i] \leftarrow \mathbf{A}[v_i, v_j]$;
4: end for
5: Calculate the $k_1$-th and $k_2$-th powers $\mathbf{A}^{k_1}$, $\mathbf{A}^{k_2}$ of the
   adjacency matrix $\mathbf{A}$;
6: $\mathbf{A}_{\mathrm{c}} \leftarrow \mathbf{A}^{k_1} + \mathbf{A}^{k_2}$;
7: for $(n \leftarrow 0; n \leq len(V) - 1; n++)$ do
8:    for $(m \leftarrow 0; m \leq len(V) - 1; m++)$ do
9:       if $\mathbf{A}_{\mathrm{c}}[n, m]! = 0$ then
10:          $\mathbf{A}_{\mathrm{c}}[n, m] \leftarrow 1$;
11:      end if
12:   end for
13: end for
14: return EAM $\mathbf{A}_{\mathrm{c}}$.
```

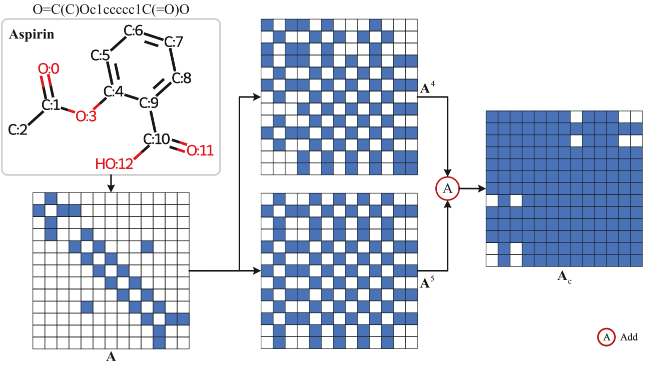


Fig. 2. NE algorithm in Aspirin.

more effective feature representation.

$$\mathbf{A}_{\mathrm{c}} = \mathbf{A}^{k_1} + \mathbf{A}^{k_2}. \tag{2}$$

*2) Protein Representation:* ParaCPI represents proteins using amino acid sequences, where each character denotes an amino acid. In this study, a vocabulary is created to map each character to an integer (i.e., glycine G for 6, histidine H for 9, and leucine L for 12). In this way, each protein can be converted to an integer sequence. Given that the amino acid sequence length of different proteins is not consistent, ParaCPI sets the maximum length of the protein sequence to 1,200 to ensure that at least 80% of the proteins can be covered. After receiving the integer encoding form of the protein, ParaCPI adopts an embedding layer to embed each amino acid into a 128-dimensional space to obtain the semantic features of different amino acids.

### B. Feature Extraction

*1) ParaGNN Module:* Inspired by CNNs, some previous work has improved the feature extraction performance of CPI prediction models by increasing the number of GCN layers. This approach improves the accuracy of CPI prediction, but this process may cause the feature expression of some nodes to be consistent. This section illustrates this drawback with the example of the graph $G = \langle V, E \rangle$, where each node $v_i$ in the node set $V$ has an initial feature vector $\mathbf{x}_i^0$. In a GNN model consisting of $l$ layers of GCNs, each layer of the GCN will update node features, and the update function is shown in (3).

$$\mathbf{x}_i^{(l+1)} = \sigma\left(\sum\nolimits_{j \in N_i} \frac{1}{n_{i,j}} \mathbf{W}^l \mathbf{x}_j^{(l)}\right), \tag{3}$$

where $\mathbf{x}_j^{(l)}$ denotes the feature vector of node $v_i$ at layer $l$. $N_i$ is the set of neighboring nodes of the node $v_i$. $\mathbf{W}^l$ is the weight matrix at the layer $l$. $n_{i,j}$ is the normalization factor of the edge between nodes $v_i$ and $v_j$, and $\sigma(\cdot)$ denotes the activation function. From the spatial perspective, the node features updated by the GCN at each layer can be regarded as the weighted summation of the neighbor node feature vector $\mathbf{x}_j^{(l)}$ of the node $v_i$, and the feature vector $\mathbf{x}_j^{(l+1)}$ of $v_i$ at the layer $l+1$ can be obtained through linear and nonlinear transformations. At each layer of the GNN, the node feature vectors are influenced by their neighbor nodes, evolving into higher dimensional embedding representations. However, the aggregation radius (the distance from the farthest node to the central node) of the node grows as $l$ increases. When the aggregation radius reaches a certain threshold, each node will aggregate the node feature information of the whole graph, as shown in Fig. 3.

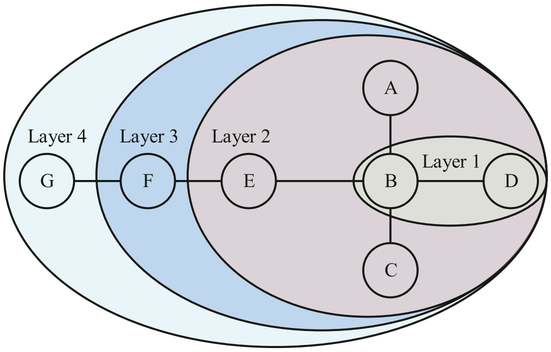


Fig. 3. GCN stacking process. With node D as the central node for graph convolution, each node will aggregate the full graph information when the GCN layer is larger than 4. At this time, the diversity of the node's local network structure is then lost, which will be detrimental to learning the node's own features.

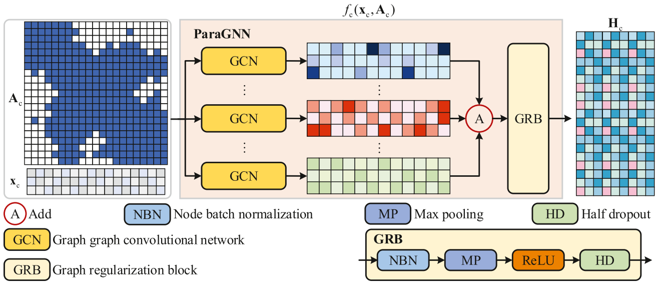

$f_c(\mathbf{x}_c, \mathbf{A}_c)$
ParaGNN
$\mathbf{A}_c$
$\mathbf{x}_c$
GCN
GCN
GCN
A
GRB
$\mathbf{H}_c$
A Add
NBN Node batch normalization
MP Max pooling
HD Half dropout
GCN Graph graph convolutional network
GRB Graph regularization block
GRB
NBN
MP
ReLU
HD

Fig. 4. ParaGNN module.

Therefore, we present a ParaGNN module composed of multiple parallel GCNs, which is based on EAM to learn the embedded representation of compounds. The structure of the ParaGNN module is shown in Fig. 4, which consists of multiple parallel GCNs and a graph regularization block (GRB). GRB includes node batch normalization (NBN) [19], max pooling (MP), the rectified linear unit (ReLU) activation function, and half dropout (HD). The calculation procedure is shown in (4).

$$\mathbf{H}_{\mathrm{c}} = f_{\mathrm{c}}(\mathbf{x}_{\mathrm{c}}, \mathbf{A}_{\mathrm{c}}) = \mathrm{GRB}\left(\sum_{c=1}^{n}\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{x}_{\mathrm{c}}\mathbf{W}^{c}\right)\right), \tag{4}$$

where $\mathbf{W}^c$ is the GCN layer of the $c$-th channel, and $n$ indicates the number of parallel GCNs. $\tilde{\mathbf{A}}$ denotes the adjacency matrix with self-connection, which is calculated by $\tilde{\mathbf{A}} = \mathbf{A}_{\mathrm{c}} + \mathbf{I}$. $\tilde{\mathbf{D}}$ represents the degree matrix of $\tilde{\mathbf{A}}$, which is estimated as $\tilde{\mathbf{D}}_{i,i} = \sum_j \tilde{\mathbf{A}}_{i,j}$. $\mathbf{I}$ denotes the identity matrix of the same dimension as $\mathbf{A}_{\mathrm{c}}$. $\mathrm{GRB}(\cdot) = \mathrm{HD}(\sigma(\mathrm{MP}(\mathrm{NBN}(\cdot))))$. The input to the ParaGNN module consists of the initial features $\mathbf{x}_{\mathrm{c}}$ of the compound and the EAM $\mathbf{A}_{\mathrm{c}}$. The EAM aggregates the feature information of almost all nodes in the molecular graph in each GCN layer by comparing the adjacency matrix of Aspirin with the EAM in Fig. 2. Obviously, this is contrary to the problem to be dealt with in this study, but the NE algorithm can avoid this phenomenon only by adjusting the size of the neighborhood expansion kernel. In addition, ParaGNN uses parallel GCN modules to ensure that the weight matrix of each GCN is independent of each other to extract multidimensional features of the atomic nodes. A unit matrix is added to the computation of $\tilde{\mathbf{A}}$. When the neighborhood expansion kernel is even, the diagonal values of $\tilde{\mathbf{A}}$ are both 2; that is, the GCN will aggregate the feature information of the central node twice. As shown in Fig. 4, partial nodes still aggregate the feature information of all nodes in the graph. ParaGNN designs the GRB

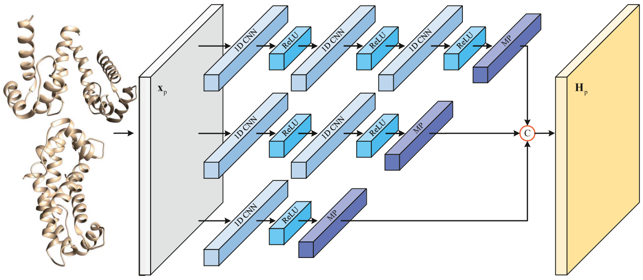


Fig. 5. DSCNN module.

to address the problem of overfitting caused by this phenomenon. First, considering that traditional batch normalization ignores the differences between nodes within a layer, GRB learns the complex structure of the data better by standardizing the features of each node within a single batch. Second, ParaGNN's parallel GCN can obtain molecular graph features from different dimensions, so GRB takes MP processing of NBN output to obtain effective feature information on different dimensions. The GRB then uses the ReLU activation function for nonlinear mapping, enabling ParaGNN to learn nonlinear patterns in the compound. Third, GRB adopts HD to clear the weights of some neurons in ParaGNN to force ParaGNN to learn different features, reduce overfitting, and ultimately improve the generalization ability of the model. HD involves early dropout (ED) and late dropout (LD). Specifically, ED denotes using dropout before an iteration and then disabling dropout for the rest of the training process. LD denotes not using dropout until a certain iteration but using dropout during the rest of the training process.

*2) DSCNN Module:* ParaCPI uses integer encoding and an embedding layer to represent the protein as a 2D matrix. The matrix size of each protein is $1,200 \times 128$, where 1,200 is the default number of amino acids per protein in this study, and 128 is the matrix size of each amino acid. A DSCNN based on a 1D CNN is built by ParaCPI to extract the chemical information from proteins. The structure of the DSCNN module is depicted in Fig. 5. The output of each CNN layer is processed by the ReLU activation function to prevent overfitting of the model, and then effective features are extracted by the MP layer at the end of the channel. Finally, these multiscale features are connected into a new vector representing the protein features. DSCNN utilizes different numbers of CNNs to construct multiple channels to efficiently extract protein feature information at different scales. This method of multiscale feature extraction has a wide range of applications in the CV field, which can extract hidden features at different levels of images. For protein features, the complex chemical information between amino acids is also multilayered. Therefore, DSCNN can obtain the embedded representation of proteins more efficiently, thereby improving the prediction accuracy of CPI.

## C. CPI Prediction

As illustrated in Fig. 1, the protein and compound features obtained by the ParaGNN and DSCNN modules are concatenated as CPC features and used by ParaCPI for CPI prediction. Specifically, four FCLs are used to predict the CPI, and the length of the FCLs is (1,024, 1,024, 256, 2). The ReLU activation function is used to learn the nonlinear features of the first three layers of the FCL. A random dropout layer is implemented for the output of the activation function. Considering that the CPI prediction problem is a traditional binary classification problem, we adopt the cross-entropy loss function to train the ParaCPI model, and it is calculated using (5) [25].

TABLE II
HYPERPARAMETER ON DIFFERENT DATASETS

| Datasets | Learning rate | Dropout | Dimension | Epoch |
|---|---|---|---|---|
| Human and C.elegans | $5 \times 10^{-4}$ | 0.2 | 87 | 100 |
| DrugBank | $5 \times 10^{-4}$ | 0.2 | 53 | 100 |
| GPCR | $5 \times 10^{-4}$ | 0.2 | 58 | 300 |
| Kinase | $1 \times 10^{-4}$ | 0.1 | 53 | 30 |

$$\text{Loss} = -[y \log \hat{y} + (1-y) \log (1-\hat{y})], \tag{5}$$

where $y$ represents the true class labels, and $\hat{y}$ is the class labels predicted by the model. Finally, ParaCPI updates the network with a set of parameters by applying a backpropagation algorithm, which aims at minimizing the loss function.

# III. Results and Discussion

## A. Experimental Setups

*1) Implementation Details:* The ParaCPI model is implemented in a software environment with Python3.7 support provided by PyTorch 1.6.0. The experiments are conducted on a server with Ubuntu 18.04, Intel(R) Core(TM) i7-6850 K CPU @ 3.60 GHz, and four NVIDIA GeForce RTX 1080 Ti graphics cards. ParaCPI uses the Adam optimizer to train the model. In the experiment, the batch size is 512, and the number of parallel GCNs is 5. Appropriate hyperparameters are selected by grid search for different datasets. The partial hyperparameters for the different datasets are reported in Table II.

*2) Datasets:* Five datasets are chosen in three different environments to assess the performance of the designed ParaCPI model. The three experimental environments include warm-start, cold-start, and label reversal settings. The warm-start setting refers to the inclusion of data (compounds or proteins) from the training set that may also appear in the test set. Three types of subjects are involved in the CPI prediction problem, including compounds, proteins, and CPCs. Thus, three types of cold-start settings are used, including compound cold-start settings, protein cold-start settings, and compound-protein cold-start settings. The three cold-start settings indicate that the model needs to detect compounds, proteins, or CPCs that have not been seen during the training phase of the model while testing. The cold-start setting is more suitable for the practical application of the CPI prediction problem than the warm-start setting. The label reversal setting indicates that a compound in the training set belongs only to the positive or negative CPI, whereas the compound in the test set belongs to the opposite class of the sample set. The

TABLE III
SUMMARY OF DATASETS

| Datasets | Settings | Protein | Compounds | Interactions | Positive | Negative | Train | Validate | Test |
|---|---|---|---|---|---|---|---|---|---|
| Human | Warm-start | 850 | 1,052 | 6,728 | 3,369 | 3,359 | 5,382 | - | 1,346 |
| C.elegans |  | 2,504 | 1,434 | 7,786 | 4,000 | 3,786 | 6,228 | - | 1,558 |
| DrugBank | Cold-start | 4,294 | 6,655 | 35,022 | 17,511 | 17,511 | [D1, P1, DP1] | [D2, P2, DP2] | [D3, P3, DP3] |
| GPCR | Label reverse | 356 | 5,359 | 15,343 | 7,989 | 7,354 | 11,045 | 2,761 | 1,537 |
| Kinase |  | 229 | 1,644 | 111,237 | 23,190 | 88,047 | 73,241 | 18,311 | 19,685 |

five publicly selected datasets selected in the experiments include Human [18], C.elegans [18], DrugBank [26], GPCR [21], and Kinase [21]. Details of the datasets are illustrated in Table III. In this study, we use 5-fold cross-validation (CV) to validate the model effect on Human and C.elegans datasets. Due to the additional requirements of the cold-start and label reversal settings on the test data, the hold-out method combined with 5-fold CV is employed to validate model performance. This approach initially extracts data meeting specific criteria from the entire dataset to form the test set. Subsequently, 5-fold CV is applied to the remaining data, with the final step involving the evaluation of model metrics on the test set. To ensure fairness, this study repeated experiments three times with three different random seeds in the warm-start setting; in the cold-start setting, it repeated the experiments ten times with ten different random seeds. D1, D2 and D3 represent the number of training set, validation set and test set samples under the cold-start setting of the compound, respectively, with sizes of 22,578, 22,357 and 17,859. Similarly, P* and DP* (* is 1, 2, or 3) represent the number of samples in different sets of protein cold-start setting and compound-protein cold-start setting, respectively. P1, P2, P3, DP1, DP2 and DP3 are 5,646, 5,590, 4,465, 6,799, 7,075 and 1,430, respectively. The reason for the lower total sample number in the compound-protein cold-start setting compared to the other two cold-start settings is to ensure that the compounds or proteins in the test set have not appeared in the training or validation sets.

*3) Baselines:* Several classical machine learning models [27] with six SOTA computational models used for CPI prediction are compared in the experiment, including GraphDTA [28], DeepConv-DTI [16], TransformerCPI [21], MGraphDTA [19], CPGL [29], and MSF-DTA [30], to verify the effectiveness of ParaCPI. The details of these baseline models are given below.

- *Classical machine learning models [27]:* This includes k-nearest neighbors (KNN), RF, L2-regularized logistic regression (L2), and SVM. The authors established a set of highly reliable negative samples of CPI by computer screening methods and used classical machine learning classifiers to predict CPI.
- *GraphDTA [28]:* Compound characteristics are obtained by combining the molecular diagram structure of the compound and GCN to predict the unknown CPI. The model in the original study is used to predict DTA, but it can also be applied to predict CPI by modifying the output dimension and loss function of the model classifier.
- *CPGL [29]:* This model comprises a long-short term memory network for proteins and a graph attention network for compounds to learn potential feature representations, thereby predicting CPI in an end-to-end manner.
- *MSF-DTA [30]:* This model learns the feature representation of proteins in a novel manner by collecting information from protein interaction networks and sequence similarity networks, which are concatenated with compound features extracted by GCN and fed into FCL for CPI prediction.

*4) Evaluation Metrics:* In the experiment, the following six metrics are used to evaluate the performance of the ParaCPI and SOTA models: accuracy, precision, recall, area under the curve (AUC), area under precision-recall (AUPR), and F1-score.

*a) Accuracy:* The accuracy indicates the proportion of CPIs predicted correctly in all predicted outcomes. High accuracy indicates better predictive performance of the model, which is estimated by

$$\text{Accuracy} = \frac{TN + TP}{FP + TP + FN + TN}, \tag{6}$$

where true negatives ($TN$) and false negatives ($FN$) represent the number of correct and incorrect noninteractive CPCs, respectively.

*b) Precision:* The precision refers to the proportion of true positive examples of CPIs among CPIs predicted to be positive samples, and it can be calculated by

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{7}$$

In the CPI prediction problem, the goal is to find potential interactions, that is, positive samples. The precision reflects the predictive ability of the model on the positive samples. High accuracy indicates that the model has a strong prediction ability.

*c) Recall:* The recall is the proportion of CPIs predicted as positive samples in the true positive examples' CPIs, which is estimated by

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{8}$$

Recall reflects the coverage of positive samples predicted by the model on the true positive samples. A high recall rate indicates good predictive performance of the model.

*d) AUC:* The AUC denotes the area surrounded by the ROC and the coordinate axes. The vertical axis represents the TPR and the horizontal axis of the ROC shows the false positive rate (FPR). The FPR is calculated using (9), and the TPR is computed using the same equation as recall.

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{9}$$

TABLE IV
COMPARISON RESULTS OF PARACPI AND BASELINES ON THE HUMAN DATASET

| Model | Precision | Recall | AUC |
|---|---|---|---|
| KNN [27] | 0.927±0.005 | 0.798±0.012 | 0.862±0.008 |
| RF [27] | 0.897±0.003 | 0.861±0.003 | 0.940±0.004 |
| L2 [27] | 0.913±0.007 | 0.867±0.008 | 0.911±0.010 |
| SVM [27] | 0.910±0.023 | 0.939±0.025 | 0.910±0.023 |
| GraphDTA [28] | 0.883±0.040 | 0.912±0.040 | 0.960±0.005 |
| DeepConv-DTI [16] | 0.942±0.002 | 0.953±0.003 | 0.970±0.003 |
| TransformerCPI [21] | 0.916±0.006 | 0.925±0.006 | 0.973±0.002 |
| MGraphDTA [19] | 0.955±0.005 | 0.956±0.003 | 0.983±0.003 |
| CPGL [29] | 0.915±0.010 | **0.957±0.008** | 0.979±0.001 |
| MSF-DTA [30] | 0.935±0.005 | 0.930±0.015 | 0.987±0.003 |
| **ParaCPI (ours)** | **0.966±0.002** | **0.957±0.003** | **0.991±0.001** |

The bold indicates the best result.

TABLE V
COMPARISON RESULTS OF PARACPI AND BASELINES ON THE C.ELEGANS DATASET

| Model | Precision | Recall | AUC |
|---|---|---|---|
| KNN [27] | 0.801±0.010 | 0.827±0.009 | 0.858±0.008 |
| RF [27] | 0.821±0.006 | 0.844±0.007 | 0.902±0.007 |
| L2 [27] | 0.890±0.003 | 0.877±0.004 | 0.892±0.004 |
| SVM [27] | 0.785±0.002 | 0.818±0.005 | 0.894±0.003 |
| GraphDTA [28] | 0.927±0.015 | 0.912±0.023 | 0.974±0.004 |
| DeepConv-DTI [16] | 0.947±0.002 | 0.950±0.001 | 0.979±0.003 |
| TransformerCPI [21] | 0.957±0.006 | 0.953±0.005 | 0.988±0.002 |
| MGraphDTA [19] | 0.980±0.004 | **0.967±0.005** | 0.991±0.001 |
| CPGL [29] | 0.956±0.003 | 0.957±0.005 | 0.990±0.001 |
| MSF-DTA [30] | 0.965±0.003 | 0.961±0.004 | 0.984±0.002 |
| **ParaCPI (ours)** | **0.980±0.002** | 0.962±0.008 | **0.995±0.001** |

The bold values indicate the best results.

The value of AUC ranges from 0 to 1, and a value close to 1 indicates excellent classifier performance.

*e) AUPR:* AUPR represents the area surrounded by the PRC curve and the horizontal axis, where the horizontal axis of the PRC curve is recall and the vertical axis is precision. The larger the value of AUPR, the stronger the classification capability of the model. The value range of AUPR is the same as that of AUC, with a value closer to 1 indicating better model performance.

*f) F1-score:* The F1-score is a statistical measure used to evaluate the accuracy of a classification model, especially when there is an uneven distribution of class categories. It represents the harmonic mean of precision and recall, providing a comprehensive metric for assessing the model's overall performance. The F1-score is calculated as follows:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

The F1-score ranges from 0 to 1, where 0 signifies the worst performance and 1 indicates perfect performance. A higher F1-score signifies a better balance between precision and recall, which is crucial in practice, as improving precision often leads to a reduction in recall, and vice versa.

## B. Comparison Experiments

*1) Comparison in Warm-Start Setting:* In the warm-start setting, we evaluate the predictive performance of ParaCPI on the Human and C.elegans datasets. The results obtained on the Human and C.elegans datasets are shown in Tables IV and V. Error band analysis plots of the results obtained from the 5-fold CV method are plotted to further demonstrate the robustness of ParaCPI, as shown in Figs. 6 and 7. For the Human dataset, ParaCPI achieves precision, recall, and AUC scores of 0.966, 0.957, and 0.991, respectively, surpassing other models in all three evaluation metrics. It demonstrates average performance gains of 5.08%, 5.52%, and 4.82%, respectively. As depicted in Fig. 6, CPGL achieves comparable recall to ParaCPI, but ParaCPI outperforms CPGL by 5.52% and 1.21% in precision and AUC scores, respectively. Notably, ParaCPI surpasses the current SOTA model MGraphDTA by 1.10% in precision. Additionally, it exceeds MSF-DTA, which currently holds the highest AUC score, in all three metrics, with performance gains of 3.27%, 2.90%, and 1.01%, respectively. In Fig. 7, ParaCPI exhibits superior AUC and accuracy compared to SOTA models on the C.elegans dataset. While ParaCPI's recall is 0.005 lower than MGraphDTA, it surpasses MGraphDTA by 0.38% in terms of AUC. Moreover, ParaCPI achieves performance gains of

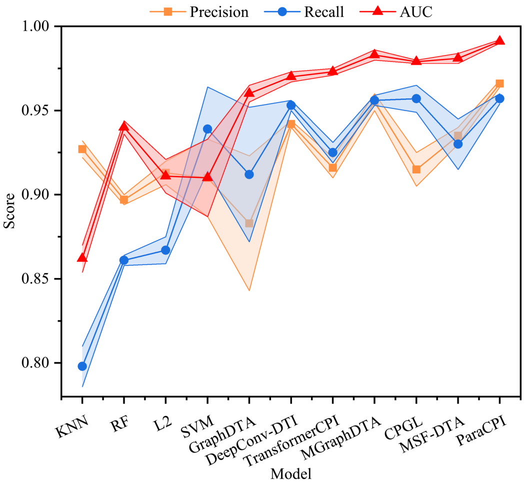


Fig. 6. Comparison results of ParaCPI and baselines on the Human dataset.

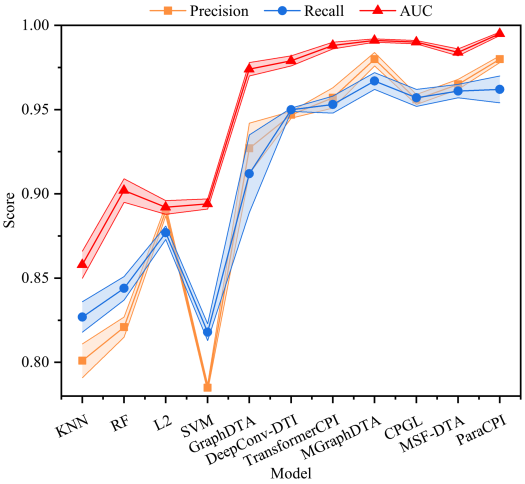


Fig. 7. Comparison results of ParaCPI and baselines on the C.elegans dataset.

TABLE VI
Comparison Results of ParaCPI and Baselines in the Cold-Start Setting

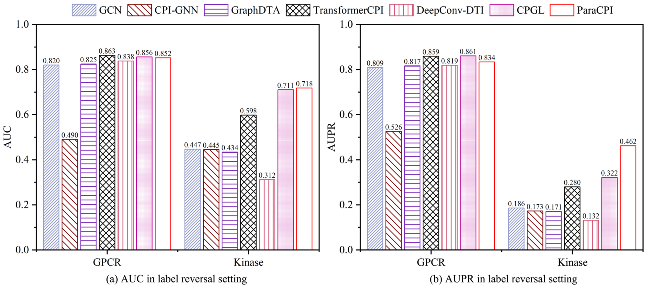
| Settings | Methods | Accuracy | Precision | Recall | AUC | AUPR |
|---|---|---|---|---|---|---|
| Compound cold-start | GNN-CPI [18] | 0.618±0.035 | 0.649±0.049 | 0.522±0.064 | 0.662±0.040 | 0.692±0.035 |
| | GNN-PT [31] | 0.615±0.008 | 0.675±0.013 | 0.436±0.036 | 0.655±0.011 | 0.684±0.019 |
| | GraphDTA [28] | 0.577±0.003 | 0.635±0.023 | 0.361±0.023 | 0.616±0.021 | 0.621±0.045 |
| | DeepConv-DTI [16] | 0.658±0.022 | 0.710±0.025 | 0.535±0.016 | 0.704±0.023 | 0.728±0.019 |
| | TransformerCPI [21] | 0.648±0.011 | 0.685±0.021 | 0.541±0.022 | 0.702±0.023 | 0.670±0.022 |
| | MolTrans [32] | 0.662±0.011 | 0.732±0.012 | 0.584±0.023 | 0.726±0.022 | 0.745±0.023 |
| | HyperAttentionDTI [26] | 0.718±0.011 | 0.774±0.023 | 0.612±0.011 | 0.785±0.011 | 0.785±0.013 |
| | ParaCPI (ours) | **0.791±0.006** | **0.778±0.020** | **0.778±0.019** | **0.871±0.003** | **0.830±0.005** |
| Protein cold-start | GNN-CPI [18] | 0.655±0.059 | 0.673±0.057 | 0.613±0.065 | 0.716±0.065 | 0.725±0.076 |
| | GNN-PT [31] | 0.713±0.021 | 0.751±0.026 | 0.641±0.063 | 0.777±0.025 | 0.783±0.016 |
| | GraphDTA [28] | 0.731±0.004 | 0.737±0.012 | 0.716±0.047 | 0.801±0.022 | 0.799±0.011 |
| | DeepConv-DTI [16] | 0.719±0.031 | 0.731±0.031 | 0.699±0.024 | 0.788±0.021 | 0.797±0.025 |
| | TransformerCPI [21] | 0.735±0.032 | 0.730±0.031 | 0.742±0.035 | 0.799±0.028 | 0.796±0.023 |
| | MolTrans [32] | 0.728±0.024 | 0.755±0.021 | 0.673±0.023 | 0.794±0.021 | 0.804±0.019 |
| | HyperAttentionDTI [26] | 0.743±0.015 | 0.765±0.017 | 0.698±0.021 | 0.818±0.014 | 0.834±0.011 |
| | ParaCPI (ours) | **0.927±0.004** | **0.921±0.011** | **0.939±0.004** | **0.970±0.003** | **0.946±0.005** |
| Compound-protein cold-start | GNN-CPI [18] | 0.557±0.023 | 0.586±0.027 | 0.418±0.078 | 0.589±0.030 | 0.607±0.015 |
| | GNN-PT [31] | 0.544±0.016 | 0.589±0.047 | 0.313±0.037 | 0.580±0.021 | 0.590±0.037 |
| | GraphDTA [28] | 0.544±0.003 | 0.586±0.027 | 0.307±0.018 | 0.579±0.025 | 0.579±0.023 |
| | DeepConv-DTI [16] | 0.585±0.033 | 0.631±0.022 | 0.415±0.025 | 0.638±0.022 | 0.622±0.021 |
| | TransformerCPI [21] | 0.611±0.022 | 0.669±0.023 | 0.467±0.024 | 0.657±0.027 | 0.664±0.025 |
| | MolTrans [32] | 0.623±0.021 | 0.705±0.024 | 0.450±0.026 | 0.685±0.022 | 0.706±0.025 |
| | HyperAttentionDTI [26] | 0.647±0.021 | **0.752±0.019** | 0.455±0.023 | 0.723±0.018 | 0.741±0.023 |
| | ParaCPI (ours) | **0.664±0.012** | 0.715±0.033 | **0.481±0.052** | **0.724±0.012** | **0.720±0.012** |

The bold values indicate the best results.

9.20%, 6.50%, and 5.54% on the three metrics compared to other models.

*2) Comparison in Cold-Start Setting:* The effectiveness of deep learning on multiple problems relies heavily on the ability of models to be trained on massive datasets. Many existing models use the result of the warm-start setting to evaluate performance. However, the cold-start setting is more in line with the practical requirements for drug discovery than the warm-start setting. A major challenge in the cold-start setting is whether the model that performs well on the training set can achieve accurate predictions on the test set. Four SOTA models are presented in this section, including GNN-CPI [18], GNN-PT [31], MolTrans [32], and HyperAttentionDTI [26], to compare the effectiveness of ParaCPI in the cold-start setting. HyperAttentionDTI does not directly connect the features extracted from the two modules but uses an attention mechanism to obtain the key information and then obtains the final features of the compounds and proteins through the MP layers. The two sets of features are concatenated and fed into the FCL to predict the CPI. The experimental results of ParaCPI with other compared models on DrugBank in the cold-start setting are reported in Table VI. The accuracy, precision, recall, AUC, and AUPR of ParaCPI are 0.791, 0.778, 0.778, 0.871, and 0.830, respectively, in the compound cold-start setting, which improves by 10.14%, 0.52%, 27.19%, 10.93%, and 5.73%, respectively, compared with the current best model HyperAttentionDTI. ParaCPI exhibits a remarkable improvement in the protein cold-start setting. The AUC of ParaCPI is 0.970, which gains an improvement of 18.61% over HyperAttentionDTI. Additionally, ParaCPI improves the other four evaluation metrics by approximately 10%-34% compared with HyperAttentionDTI. When CPCs that are not seen during model training are input during model testing, ParaCPI improves by 2.60%, 5.63% and 0.19% compared with HyperAttentionDTI in terms of accuracy, recall, and AUC, respectively. Table VI shows that ParaCPI is 0.037 and 0.021 lower than HyperAttentionDTI in terms of accuracy and recall in the compound-protein cold-start setting, respectively. The reason behind this phenomenon is that the attention mechanism adopted by HyperAttentionDTI after extracting the compound and protein features separately strengthens the connection between the two. However, it is this connection that causes the model to fail to achieve suitable performance in the other two cold start environments, whereas ParaCPI shows good performance in three cold-start settings. In particular, the average performance gains for the five performance metrics of ParaCPI are 29.45%, 25.40%, 38.98%, 23.69%, and 19.63% in the protein cold-start setting compared with the other seven models.

The experimental results in Table VI demonstrate that ParaCPI is superior to the SOTA models on the DrugBank dataset in evaluation metrics under the cold-start setting, with only a few exceptions. The reason for the excellent performance of ParaCPI in the cold-start setting is multifold. First, the NE algorithm is crucial for the expansion of node neighborhoods. Compared with the GNN-CPI, GNN-PT, and GraphDTA models without the NE algorithm, ParaCPI can construct compound feature representations with stronger generalization and higher efficiency based on known CPCs. For recall, ParaCPI increases by 49.12%, 78.53%, and 44.68% compared with GNN-CPI, GNN-PT, and GraphDTA, respectively. Second, the DSCNN exhibits the ability to extract sequence features without inferiority to the transformer framework. For the protein feature extraction module, the other seven compared models use FCL, transformer architecture or stacked CNN layers to extract protein features. From the experimental results of the protein cold start setting, ParaCPI achieves a more efficient protein feature extraction capability and can use these features to predict interactions



(a) AUC in label reversal setting

(b) AUPR in label reversal setting

Fig. 8. Comparison results of ParaCPI and baselines on the GPCR and Kinase datasets.

between compounds and unseen proteins. Third, the parallelization design of ParaGNN alleviates the deficiency of NE, and GRB effectively eases the problem of overfitting in the training process.

*3) Comparison in Label Reversal Setting:* The label reversal setting is designed to validate whether the model has hidden ligand bias issues, which was first reported in the DUD-E and MUV datasets [33]. The hidden ligand bias problem refers to the fact that the model predicts CPI primarily based on ligand patterns rather than on information about CPIs, which will lead to difficulties in matching between theoretical modeling and practical applications [21]. The experimental results of ParaCPI with other baseline models in the label reversal setting are illustrated in Fig. 8. Fig. 8 shows that ParaCPI does not have a remarkable performance advantage on the GPCR dataset. For AUC, ParaCPI is 0.011 lower than that of TransformerCPI. For AUPR, ParaCPI is 0.027 lower than that of CPGL. This phenomenon may arise from the size of the dataset. On the Kinase dataset with larger training data, the AUC and AUPR of ParaCPI reached 0.718 and 0.462, which increased by 0.98% and 43.48% compared with CPGL, respectively. The Kinase dataset is a category-imbalanced dataset compared with the GPCR dataset; that is, the AUPR can better reflect the prediction performance of the model. Although the prediction performance of ParaCPI on the GPCR dataset does not exceed that of the current SOTA model, ParaCPI does not suffer from severe hidden ligand bias compared with the GCN, CPI-GNN, and GraphDTA models, which is confirmed by experimental results on the Kinase dataset.

## *C. Ablation Study*

The core idea of ParaCPI depends on the design of ParaGNN combined with the NE algorithm. The following four variant models are developed using ParaCPI in the experiments to verify the effectiveness of each module in ParaCPI. CPI-NE: The NE algorithm is not used in the data processing process, but the features of other nodes are aggregated using the adjacency matrix of atomic nodes in the original drug molecule map. CPI-ParaGNN: The ParaGNN module is replaced with a five-layer stacked GCN module (the same number of GCN layers used by ParaGNN) based on CPI-NE, but the GRB is retained. CPI-GRB: This model removes GRB based on the CPI-ParaGNN. CPI-DSCNN: This model replaces the DSCNN module with six stacked CNN layers based on CPI-GRB.

As depicted in Table VII, when NE is not utilized, ParaCPI's accuracy, recall, AUC, and F1-score on the Human dataset decrease by 2.66%, 3.01%, 2.69%, and 2.46%, respectively. These reductions directly impact ParaCPI's predictive performance. CPI-ParaGNN surpasses CPI-NE across all evaluation metrics on both datasets. CPI-ParaGNN outperforms CPI-NE in terms of all evaluation metrics on both datasets. The reason for this phenomenon is that when EAM is not used, CPI-ParaGNN needs to use multiple layers of stacked GCNs to construct effective compound features. CPI-NE uses ParaGNN to extract compound features based on the adjacency matrix and cannot extract useful compound features. Despite replacing ParaGNN with a compound feature extraction module akin to GraphDTA, CPI-ParaGNN exhibits a 7.70% and 5.18% enhancement in accuracy over GraphDTA on the two datasets, respectively. This improvement is attributed to GRB, which successfully reduces the degree of overfitting of the model and enables the model to achieve more accurate prediction performance while stacking several layers of GCNs. When CPI-GRB removes GRB, the model tends to overfit due to the excessive number of GCN layers, leading to a decrease in the AUC of the model by 2.64% and 1.76% compared to CPI-ParaGNN on the two datasets, respectively. The structure of CPI-DSCNN is similar to that of GraphDTA. The distinction between the two lies in the depth of the network. CPI-DSCNN, having a deeper network, achieves AUCs of 0.934 and 0.959 on the two datasets, which are 2.78% and 1.56% lower than those of GraphDTA, respectively. Additionally, the F1-score of ParaCPI outperforms other variant models, exhibiting an average increase of 3.93% and 3.06% across the two datasets. The experimental results illustrate that ParaGNN combined with the NE algorithm can effectively extract compound features. GRB can also effectively adjust model parameters to prevent the model from overfitting. DSCNN has a more efficient protein feature extraction capability compared with the constantly stacked CNN module.

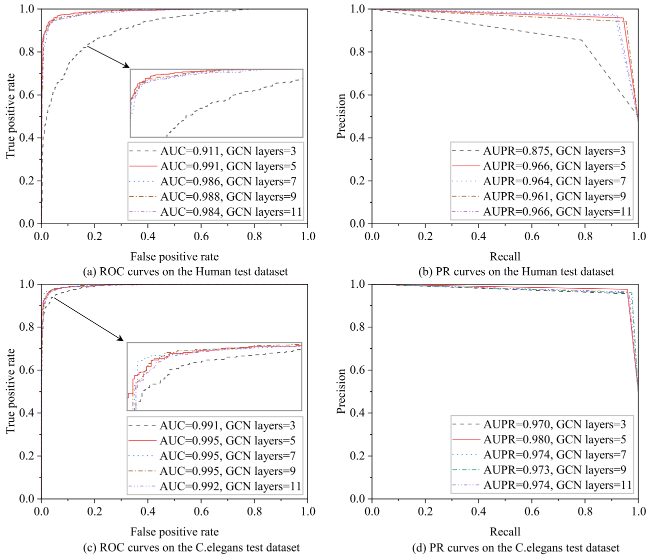
## *D. Performance of ParaCPI With Different Numbers of GCNs*

Given that the feature dimension of compounds is unknown to the model, determining the GCN number of ParaGNN is the key to extracting compound features. When the number of GCNs is extremely small, ParaGNN cannot extract efficient compound features for CPI prediction. When the number of GCNs is extremely large, the model may produce incorrect feature representations and affect the prediction performance. Fig. 9 reports the ROC and PR curves of ParaCPI under different numbers of GCNs on the Human and C.elegans datasets. Fig. 9 shows that the AUC and AUPR are low when the number of GCNs is 3. The AUC and AUPR reach the maximum when the number of GCN is 5. As the number of GCNs increases, the model's prediction performance exhibits a declining trend. Specifically, the AUC of ParaCPI decreases by 0.007 and 0.003 on the Human and C.elegans datasets, respectively, when the number of GCNs reaches 11. The AUC of ParaCPI under different GCN numbers still exceeds that of most current SOTA models.

TABLE VII
PERFORMANCE OF DIFFERENT VARIANT MODELS ON THE HUMAN AND C.ELEGANS DATASETS

| Model | Human | | | | C.elegans | | | |
|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | AUC | F1-score | Precision | Recall | AUC | F1-score |
| **ParaCPI** | **0.966±0.002** | **0.957±0.003** | **0.991±0.001** | **0.958±0.002** | **0.980±0.002** | **0.962±0.008** | **0.995±0.001** | **0.968±0.002** |
| CPI-NE | 0.941±0.013 | 0.929±0.004 | 0.965±0.001 | 0.935±0.008 | 0.971±0.009 | 0.940±0.019 | 0.980±0.003 | 0.955±0.010 |
| CPI-ParallelGCN | 0.951±0.007 | 0.949±0.004 | 0.971±0.002 | 0.950±0.005 | 0.975±0.007 | 0.951±0.005 | 0.983±0.001 | 0.963±0.008 |
| CPI-GRB | 0.891±0.006 | 0.924±0.005 | 0.946±0.004 | 0.907±0.009 | 0.923±0.003 | 0.917±0.007 | 0.966±0.001 | 0.920±0.016 |
| CPI-DSCNN | 0.876±0.011 | 0.914±0.013 | 0.934±0.002 | 0.895±0.015 | 0.919±0.019 | 0.920±0.004 | 0.959±0.002 | 0.919±0.012 |

The bold values indicate the best results.



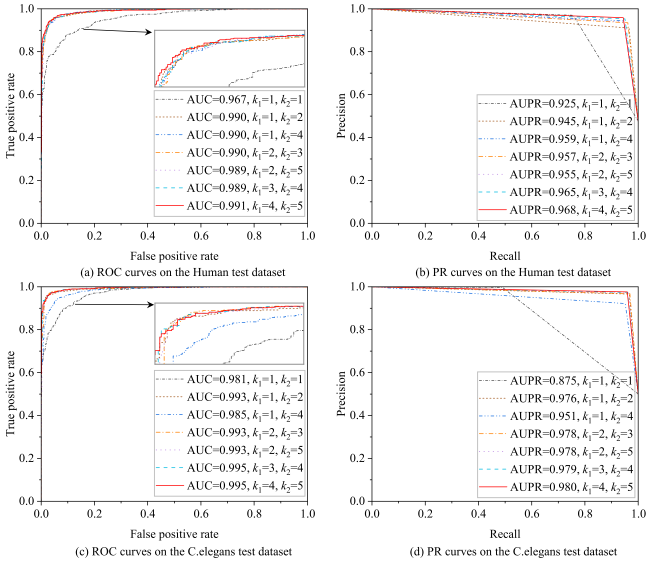
(a) ROC curves on the Human test dataset

(b) PR curves on the Human test dataset

(c) ROC curves on the C.elegans test dataset

(d) PR curves on the C.elegans test dataset

Fig. 9. The ROC curves and PR curves of ParaCPI with different numbers of GCNs on the Human and C.elegans test datasets.



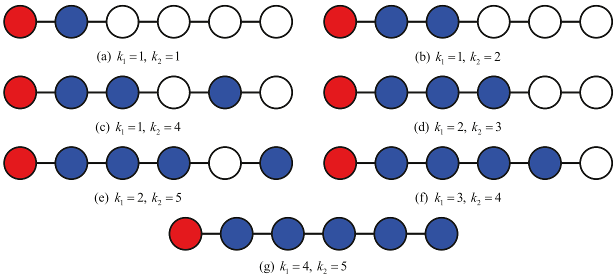
(a) ROC curves on the Human test dataset

(b) PR curves on the Human test dataset

(c) ROC curves on the C.elegans test dataset

(d) PR curves on the C.elegans test dataset

Fig. 10. The ROC curves and PR curves of ParaCPI with different neighborhood expansion kernel sizes on the Human and C.elegans test datasets.



(a) $k_1=1,\ k_2=1$

(b) $k_1=1,\ k_2=2$

(c) $k_1=1,\ k_2=4$

(d) $k_1=2,\ k_2=3$

(e) $k_1=2,\ k_2=5$

(f) $k_1=3,\ k_2=4$

(g) $k_1=4,\ k_2=5$

Fig. 11. Node association features are constructed by the NE algorithm at different k values. Red represents the starting node and the blue represents the neighborhood dilation node.

### *E. Performance of ParaCPI With Different Neighborhood Expansion Kernel Sizes*

The good feature extraction capability of ParaGNN is largely derived from the EAM constructed by the NE algorithm. The appropriate neighborhood expansion kernel size is the key to constructing an effective EAM. Fig. 10 shows the ROC and PR curves of ParaCPI on the Human and C.elegans datasets with different neighborhood expansion kernel sizes. Fig. 11 describes the EAM construction process to illustrate the differences between neighborhood nodes under different $k$ values. ParaCPI does not adopt the NE algorithm to construct the association features when $k_1 = 1,\ k_2 = 1$ due to the binarization operation in NE, which can be observed in Fig. 11(a). Fig. 10 shows that when $k_1 = 1,\ k_2 = 1$, ParaCPI has the smallest areas under the ROC and PR curves on the two datasets. When the NE algorithm is applied, the parity of the two neighborhood expansion kernels selected in this study is inconsistent. This occurs because when the size of the neighborhood extension kernels is either odd or even, it leads to the exclusion of either even-order or odd-order neighborhood node features of the central node. When $k_1 = 4,\ k_2 = 5$, the AUC and AUPR of ParaCPI using the NE algorithm on the Human dataset are 0.991 and 0.966, respectively. An interesting phenomenon that can be observed from Fig. 10 is that ParaCPI (i.e., Fig. 11(d)–(g)), considering third-order neighborhood node features, has better prediction performance. This is consistent with the view that most CPI prediction models without the NE algorithm set the number of GCN layers to 3. As can be observed in Fig. 10, the model performance of $k_1 = 3,\ k_2 = 4$ and $k_1 = 4,\ k_2 = 5$ is highly similar. On the basis of the results of Fig. 11(f)–(g) the degree of the selected starting node in this study is 1. When the degree of the selected starting node is greater than 1, the EAM constructed by $k_1 = 3,\ k_2 = 4$ or $k_1 = 4,\ k_2 = 5$ is the same in Fig. 11. This similarity in performance is because the degree of the atomic nodes in the molecular graph of the compound is mostly

TABLE VIII
PREDICTION RESULTS OF PARACPI ON THE DB00201 DATASET

| CID | PID | Pred | CID | PID | Pred |
|---|---|---|---|---|---|
| DB00201 | Q14432 | 0.992 | DB00201 | Q08493 | 0.647 |
| DB00201 | Q9HCR9 | 0.991 | DB00201 | O95263 | 0.328 |
| DB00201 | O60658 | 0.920 | DB00201 | Q08499 | 0.310 |
| DB00201 | P42338 | 0.795 | DB00201 | P35913 | 0.165 |
| DB00201 | Q13946 | 0.790 | DB00201 | P16499 | 0.126 |
| DB00201 | P27815 | 0.779 | DB00201 | Q14123 | 0.119 |
| DB00201 | Q9NP56 | 0.770 | DB00201 | Q9Y233 | 0.112 |
| DB00201 | Q13370 | 0.751 | DB00201 | P42336 | 0.041 |
| DB00201 | Q01064 | 0.714 | DB00201 | P51160 | 0.018 |
| DB00201 | O76074 | 0.688 | DB00201 | P29275 | 0.012 |

TABLE IX
PREDICTION RESULTS OF PARACPI ON THE Q99928 DATASET

| CID | PID | Pred | CID | PID | Pred |
|---|---|---|---|---|---|
| DB01589 | Q99928 | 0.997 | DB01595 | Q99928 | 0.971 |
| DB00897 | Q99928 | 0.996 | DB00475 | Q99928 | 0.971 |
| DB00404 | Q99928 | 0.996 | DB06579 | Q99928 | 0.959 |
| DB01215 | Q99928 | 0.996 | DB01588 | Q99928 | 0.954 |
| DB00801 | Q99928 | 0.994 | DB01068 | Q99928 | 0.930 |
| DB00546 | Q99928 | 0.991 | DB11901 | Q99928 | 0.922 |
| DB00334 | Q99928 | 0.990 | DB01567 | Q99928 | 0.878 |
| DB01559 | Q99928 | 0.983 | DB00543 | Q99928 | 0.847 |
| DB00252 | Q99928 | 0.982 | DB00690 | Q99928 | 0.847 |
| DB01558 | Q99928 | 0.972 | DB00829 | Q99928 | 0.835 |

greater than 1. Hence, the neighborhood expansion kernel is set to $k_1 = 4$, $k_2 = 5$ for ensuring the prediction ability of the model and enhancing the generalization ability of ParaCPI.

### F. Case Study

The compound DB00201 (Caffeine) and the protein Q99928 (gamma-aminobutyric acid receptor subunit gamma-3) are selected for the case studies to further evaluate the efficiency of ParaCPI on the top-ranking prediction. The DB00201 and Q99928 datasets are obtained from the DrugBank website for proteins or compounds that have been confirmed to interact with each other. ParaCPI is employed to predict the interaction probability of compounds and proteins on the DB00201 and Q99928 datasets (interaction exists when prediction >=0.5). The top 20 CPI pairs of predicted results from the two datasets are reported in Tables VIII and IX, where CID and PID denote compound ID and protein ID, respectively. Pred is the probability score of the interaction predicted by ParaCPI. Caffeine is a drug of the methylxanthine class used for various purposes, involving certain respiratory conditions in premature newborns, pain relief, and combatting drowsiness. Taking the first CPI in Table VIII as an example, DB00201 inhibits the activity of Q14432 (cGMP-inhibited 3’, 5’-cyclic phosphodiesterase A), thereby affecting a variety of cellular signaling and physiological effects, including effects on the biological clock, regulation of vasodilation, and cardiovascular systems. The function of Q99928 is to inhibit the ion channel activity of the extracellular ligand-gate. Taking the first CPI in Table VIII as an example, DB00201 inhibits the activity of Q14432 (cGMP-inhibited 3’, 5’-cyclic phosphodiesterase A), thereby affecting a variety of cellular signaling and physiological effects, including effects on the biological clock, regulation of vasodilation and cardiovascular systems. The function of Q99928 is to inhibit the ion channel activity of the extracellular ligand-gate. DB01589 (Quazepam) in Table IX is a long-acting benzodiazepine used to manage insomnia. Quazepam can enter the brain and interact with Q99928, thereby reducing neuronal excitability and nervous system activity. The prediction results from Tables VIII and IX show that ParaCPI has a wide range of application in CPI prediction problems. Especially when faced with unseen proteins, ParaCPI successfully predicts 20 of the first 20 CPIs with a prediction probability of over 80%. However, ParaCPI only successfully predicts 11 out of the first 20 CPIs in the face of previously unseen drugs, which may be due to the lack of integration of CPI features of the model, resulting in poor prediction performance.

## IV. CONCLUSION

In this study, a novel sequence-based CPI prediction model, ParaCPI, is developed to predict unknown CPIs for CPCs with known interactions. Different from existing models, ParaCPI uses a unique neighborhood feature construction method to extract the structural features of compounds using a novel approach. Additionally, a DSCNN module is integrated into ParaCPI to obtain potential feature representations of proteins. The experimental results show that ParaCPI performs better than SOTA models in three different settings. In particular, ParaCPI achieves AUCs of 0.871, 0.970, and 0.724 in three real-world application settings (cold-start settings), which are improvements of 26.75%, 23.84%, and 14.68%, respectively, on average, compared with the GNN-CPI, GNN-PT, GraphDTA, DeepConv-DTI, TransformerCPI, MolTrans, and HyperAttentionDTI models. Additionally, the case study in this study shows that ParaCPI can detect CPIs with considerable accuracy, as ParaCPI accurately predicts the presence of interactions in 31 out of 40 CPCs that had been confirmed by DrugBank. In the near future, we plan to capture feature representations of text sequences based on the transformer architecture, expand the cognitive ability of the model by adding more biological information, and adopt a self-attention mechanism to increase the interpretability of the model.

## CODE AND DATA AVAILABILITY

The source code for ParaCPI and the experiment dataset can be found at https://github.com/Zengwenliang0416/ParaCPI.

## ACKNOWLEDGMENTS

The authors would like to thank three anonymous reviewers for their suggestions to improve the manuscript.

## REFERENCES

[1] R. Macarron et al., “Impact of high-throughput screening in biomedical research,” *Nature Rev. Drug Discov.*, vol. 10, no. 3, pp. 188–195, 2011.

[2] O. Trott and A. J. Olson, “AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading,” *J. Comput. Chem.*, vol. 31, no. 2, pp. 455–461, 2010.

[3] Y.-C. Li et al., “PPAEDTI: Personalized propagation auto-encoder model for predicting drug-target interactions,” *IEEE J. Biomed. Health Inform.*, vol. 27, no. 1, pp. 573–582, Jan. 2023.

[4] M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, and F. Wang, “Graph convolutional networks for computational drug development and discovery,” *Brief. Bioinf.*, vol. 21, no. 3, pp. 919–935, 2020.

[5] L. Zhang, W. Zeng, J. Chen, J. Chen, and K. Li, “GDilatedDTA: Graph dilation convolution strategy for drug target binding affinity prediction,” *Biomed. Signal Process. Control*, vol. 92, 2024, Art. no. 106110.

[6] X. Chen et al., “Drug-target interaction prediction: Databases, web servers and computational models,” *Brief. Bioinf.*, vol. 17, no. 4, pp. 696–712, Aug. 2015.

[7] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, “Prediction of drug–target interaction networks from the integration of chemical and genomic spaces,” *Bioinformatics*, vol. 24, no. 13, pp. i232–i240, 2008.

[8] K. Bleakley and Y. Yamanishi, “Supervised prediction of drug–target interactions using bipartite local models,” *Bioinformatics*, vol. 25, no. 18, pp. 2397–2403, 2009.

[9] L. Peng, B. Liao, W. Zhu, Z. Li, and K. Li, “Predicting drug-target interactions with multi-information fusion,” *IEEE J. Biomed. Health Inform.*, vol. 21, no. 2, pp. 561–572, Mar. 2017.

[10] R. Zhang, “An ensemble learning approach for improving drug–target interactions prediction,” in *Proc. 4th Int. Conf. Comput. Eng. Netw.*, W. E. Wong, Ed., Springer International Publishing, Cham, Switzerland, 2015, pp. 433–442.

[11] 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, Nov. 2021, Art. no. bbab491.

[12] Y. Chu et al., “DTI-CDF: A cascade deep forest model towards the prediction of drug-target interactions based on hybrid features,” *Brief. Bioinf.*, vol. 22, no. 1, pp. 451–462, 2021.

[13] H. Öztürk, A. Özgür, and E. Ozkirimli, “DeepDTA: Deep drug–target binding affinity prediction,” *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018.

[14] Y. Qian, X. Li, Q. Zhang, and J. Zhang, “SPP-CPI: Predicting compound–protein interactions based on neural networks,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 1, pp. 40–47, Jan./Feb. 2022.

[15] 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, no. 2, pp. 134–140, 2020.

[16] 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, no. 6, 2019, Art. no. e1007129.

[17] I. Lee and H. Nam, “Sequence-based prediction of protein binding regions and drug–target interactions,” *J. Cheminformatics*, vol. 14, no. 1, pp. 1–15, 2022.

[18] 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, no. 2, pp. 309–318, 2019.

[19] Z. Yang, W. Zhong, L. Zhao, and C. Y.-C. Chen, “MGraphDTA: Deep multiscale graph neural network for explainable drug–target binding affinity prediction,” *Chem. Sci.*, vol. 13, no. 3, pp. 816–833, 2022.

[20] 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, no. 1, pp. 1–17, 2020.

[21] 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.

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

[23] D. Weininger, “SMILES, A chemical language and information system. 1. Introduction to methodology and encoding rules,” *J. Chem. Inf. Comput. Sci.*, vol. 28, no. 1, pp. 31–36, 1988.

[24] C. Li, W. Wei, J. Li, J. Yao, X. Zeng, and Z. Lv, “3DMol-Net: Learn 3D molecular representation using adaptive graph convolutional network based on rotation invariance,” *IEEE J. Biomed. Health Inform.*, vol. 26, no. 10, pp. 5044–5054, Oct. 2022.

[25] M. Liu et al., “A deep learning method for breast cancer classification in the pathology images,” *IEEE J. Biomed. Health Inform.*, vol. 26, no. 10, pp. 5025–5032, Oct. 2022.

[26] Q. Zhao, H. Zhao, K. Zheng, and J. Wang, “Hyperattentiondti: Improving drug–protein interaction prediction by sequence-based deep learning with attention mechanism,” *Bioinformatics*, vol. 38, no. 3, pp. 655–662, 2022.

[27] 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, no. 12, pp. i221–i229, 2015.

[28] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, “GraphDTA: Predicting drug–target binding affinity with graph neural networks,” *Bioinformatics*, vol. 37, no. 8, pp. 1140–1147, 2021.

[29] M. Zhao, M. Yuan, Y. Yang, and S. X. Xu, “CPGL: Prediction of compound-protein interaction by integrating graph attention network with long short-term memory neural network,” *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 3, pp. 1935–1942, May/Jun. 2023.

[30] W. Ma et al., “Predicting drug-target affinity by learning protein knowledge from biological networks,” *IEEE J. Biomed. Health Inform.*, vol. 27, no. 4, pp. 2128–2137, Apr. 2023.

[31] J. Wang, X. Li, and H. Zhang, “GNN-PT: Enhanced prediction of compound-protein interactions by integrating protein transformer,” 2020, *arXiv: 2009.00805*.

[32] K. Huang, C. Xiao, L. M. Glass, and J. Sun, “MolTrans: Molecular interaction transformer for drug–target interaction prediction,” *Bioinformatics*, vol. 37, no. 6, pp. 830–836, 2021.

[33] J. Sieg, F. Flachsenberg, and M. Rarey, “In need of bias control: Evaluating chemical data for machine learning in structure-based virtual screening,” *J. Chem. Inf. Model.*, vol. 59, no. 3, pp. 947–961, 2019.

**Longxin Zhang** (Member, IEEE) received the PhD degree in computer science from Hunan University, China, in 2015. He is currently an associate professor of computer science with the Hunan University of Technology. He was also a visiting scholar with the University of Florida from 2019 to 2020. His major research interests include bioinformatics, modeling and scheduling for distributed computing systems, distributed system reliability, parallel algorithms, cloud computing, edge computing, and deep learning. He has published more than 40 technique papers in international journals and conferences such as *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Sustainable Computing*, *Information Sciences*, *Neural Computing and Applications*, PPNA, ICA3PP, etc. He is the reviewers of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Industrial Informatics*, *ACM Transactions on Intelligent Systems and Technology*, *IEEE Internet of Things Journal*, ASC, INS, and so on.

**Wenliang Zeng** received the BEng degree from the Hunan University of Technology, Zhuzhou, China, in 2021. He is currently working toward the masters degree with the School of Computer Science and Technology, Hunan University of Technology. His research interests include bioinformatics, deep learning, and graph neural networks.

**Jingsheng Chen** received the BEng degree from the Hunan University of Technology, Zhuzhou, China, in 2017. He is currently working toward the master degree in computer science and technology with the Hunan University of Technology. Her research interests include object detection and computer vision.

**Jianguo Chen** (Member, IEEE) received the PhD degree in computer science and technology from Hunan University, China, in 2018. He is currently an associate professor with the School of Software Engineering, Sun Yat-Sen University. He was a research scientistic with the Institute for Infocomm Research, Agency for Science Technology and Research, Singapore from 2020 to 2021. He was a postdoctoral fellow with the University of Toronto, Canada, and Hunan University, China from 2018 to 2020. His major research interests include distributed computing, artificial intelligence, computer vision, and robot cluster control.

**Keqin Li** (Fellow, IEEE) is a SUNY distinguished professor of computer science with the State University of New York. He is also a National distinguished professor with Hunan University, China. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, Big Data computing, high performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, intelligent and soft computing. He has authored or coauthored more than 880 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world's top 5 most influential scientists in parallel and distributed computing in terms of both single-year impact and career-long impact based on a composite indicator of Scopus citation database. He has chaired many international conferences. He is currently an associate editor of the *ACM Computing Surveys* and *CCF Transactions on High Performance Computing*. He has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*. He is an AAIA fellow. He is also a member of Academia Europaea (Academician of the Academy of Europe).