Interdisciplinary Sciences: Computational Life Sciences (2024) 16:361–377
https://doi.org/10.1007/s12539-024-00609-y

ORIGINAL RESEARCH ARTICLE

GraphsformerCPI: Graph Transformer for Compound–Protein Interaction Prediction

Jun Ma[1,2] · Zhili Zhao[1] · Tongfeng Li[1,3] · Yunwu Liu[1] · Jun Ma[1] · Ruisheng Zhang[1]

Received: 7 August 2023 / Revised: 1 January 2024 / Accepted: 8 January 2024 / Published online: 8 March 2024

Abstract

Accurately predicting compound–protein interactions (CPI) is a critical task in computer-aided drug design. In recent years, the exponential growth of compound activity and biomedical data has highlighted the need for efficient and interpretable prediction approaches. In this study, we propose GraphsformerCPI, an end-to-end deep learning framework that improves prediction performance and interpretability. GraphsformerCPI treats compounds and proteins as sequences of nodes with spatial structures, and leverages novel structure-enhanced self-attention mechanisms to integrate semantic and graph structural features within molecules for deep molecule representations. To capture the vital association between compound atoms and protein residues, we devise a dual-attention mechanism to effectively extract relational features through .cross-mapping. By extending the powerful learning capabilities of Transformers to spatial structures and extensively utilizing attention mechanisms, our model offers strong interpretability, a significant advantage over most black-box deep learning methods. To evaluate GraphsformerCPI, extensive experiments were conducted on benchmark datasets including human, *C. elegans*, Davis and KIBA datasets. We explored the impact of model depth and dropout rate on performance and compared our model against state-of-the-art baseline models. Our results demonstrate that GraphsformerCPI outperforms baseline models in classification datasets and achieves competitive performance in regression datasets. Specifically, on the human dataset, GraphsformerCPI achieves an average improvement of 1.6% in AUC, 0.5% in precision, and 5.3% in recall. On the KIBA dataset, the average improvement in Concordance index (CI) and mean squared error (MSE) is 3.3% and 7.2%, respectively. Molecular docking shows that our model provides novel insights into the intrinsic interactions and binding mechanisms. Our research holds practical significance in effectively predicting CPIs and binding affinities, identifying key atoms and residues, enhancing model interpretability.

Zhili Zhao, Tongfeng Li, Yunwu Liu and Jun Ma contributed equally to this work.

✉ Jun Ma
maj19@lzu.edu.com

✉ Ruisheng Zhang
zhangrs@lzu.edu.cn

Zhili Zhao
zhaozhl@lzu.edu.cn

Tongfeng Li
litf19@lzu.edu.cn

Yunwu Liu
liuyw19@lzu.edu.cn

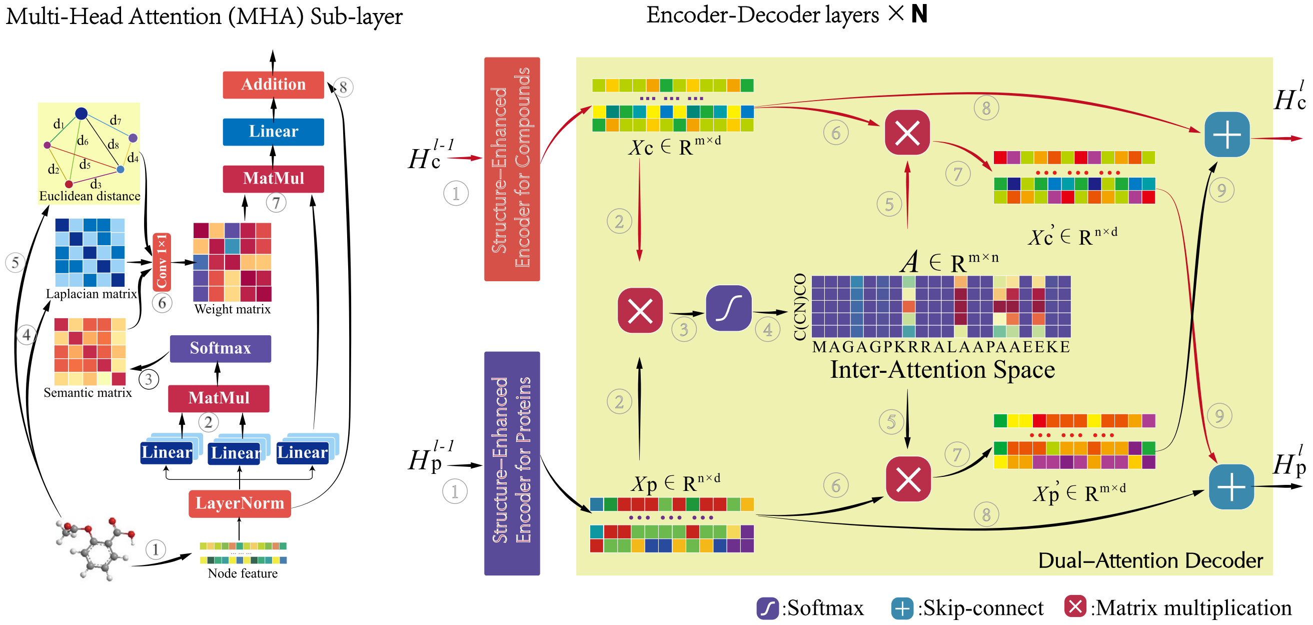
Jun Ma
junma@lzu.edu.cn

[1] School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China

[2] School of Information Engineering, Lanzhou University of Finance and Economics, Lanzhou 730020, China

[3] Computer College, Qinghai Normal University, Xi'ning 810016, China

Graphical abstract



Keywords CPI prediction · Deep learning · Molecular graph · Attention mechanism

1 Introduction

Predicting compound–protein interactions (CPIs) is a pivotal stage in drug discovery. It plays a crucial role in drug hit identification, lead discovery, comprehension of drug side effects, drug repositioning, polypharmacology, drug resistance, and other facets [1–3]. Accurate prediction of CPIs or drug–protein-binding affinity can significantly accelerate lead compound generation and optimization, facilitating the screening of potential candidate compounds [4].

The explosive growth of compound activity and biomedical data has brought new challenges to the field of CPI prediction. Although classical virtual screening (VS) [5] is beneficial, it does not accelerate CPI prediction [6]. Standard high-throughput screening (HTS) [7] experimental methods enable rapid automated testing of thousands to millions of molecular samples. However, there are still limitations on the time and cost for screening large libraries [8–10]. Advances in processing speed and storage capacity of computing devices have made big data analysis possible, driving the rapid development of artificial intelligence and machine learning technologies [11]. Machine learning techniques have been widely utilized in drug discovery to predict interactions using large amounts of high-dimensional complex data. These techniques offer more promising lead candidates for experimental validation and thus reducing both time and cost [4].

In recent years, deep learning methods have triggered unprecedented advancements in many fields, such as computer vision [12], speech recognition [13], machine translation [14], and recommended systems [15]. Moreover, these methods have demonstrated impressive predictive capabilities in addressing computational biology problems [16]. Similar to traditional machine learning, the effectiveness of deep learning is influenced by the selected features. However, machine learning heavily relies on high-quality features manually extracted by domain experts. In contrast, deep learning models can automatically identify multidimensional low-level hidden features from structured data. Compounds and proteins can be considered structural data, which make them particularly suitable for processing by deep learning models.

Many methods for CPI prediction or drug-target-related studies are derived from natural language processing (NLP) [6]. Small molecule compounds, typically represented as SMILES strings [17], and proteins, often depicted in FASTA format [18], are naturally viewed as one-dimensional sequences with rich semantics. Consequently, various sequence-based models, including recurrent neural network (RNN) [19], long short-term memory (LSTM) [20], 1D convolutional neural network (1D CNN)

[21], Transformer [22], and their respective variations, have been employed to identify recurring patterns in a given sequence.

DeepDTA [23] consisted of two independent 1D CNN blocks that learned representations from SMILES strings and protein sequences, and computed the final output through a fully connected layer. This methodology has demonstrated promising potential for enhancing the predictive accuracy of CPIs. DeepCPI [24] employed latent semantic analysis and Word2Vec techniques to learn low-dimensional feature representations of compounds and proteins from large corpora of compounds and proteins. The model emphasized the extraction of comprehensive and meaningful features from complex molecular and protein structures, thus contributing to more accurate CPI prediction. DeepAffinity [25] leveraged RNNs and 1D CNNs to jointly encode molecular representations and obtained long-term, nonlinear dependencies between atoms and residues in compounds and proteins. It achieved significant advancements in CPI prediction. TransformerCPI [26] applied 1D CNN layers to encode protein sequences and utilized word embedding methods for encoding small molecule compounds. The model integrated an attention mechanism to highlight important interaction regions between protein sequences and compound atoms, which was a significant step forward in modeling the intricate relationship between compounds and proteins. DeepLSTM [27] extracted protein evolutionary features by position specific scoring matrix (PSSM) and Legendre moment (LM), correlating them with drug molecular substructure fingerprints to create a drug–target pair feature vector. Experimental results have demonstrated great advantages on feature expression and recognition.

Although sequence representation is simple and convenient, one-dimensional notations are inadequate for representing molecular topology. Two similar chemical structures may be represented by vastly different one-dimensional symbols, and even the same compound can be represented by different one-dimensional sequences. Consequently, critical information about compound and protein structures is lost, which impairs the predictive power of the model and hinders learning about functional relevance of the latent space [28, 29].

Graph neural networks (GNNs) are designed specifically to process graph structures and can simultaneously learn the strength of connections between different nodes. The effectiveness of GNN in predicting CPIs has been widely acknowledged due to its representation learning capability. Molecular graphs are used to represent small molecule compounds, where atoms and chemical bonds serve as nodes and edges, respectively. Similarly, proteins can be depicted as structural graphs, with nodes representing atoms or amino acids and edges representing their neighborhood relationships. Various graph-based neural networks, including graph convolutional network (GCN) [30] and their variants, are commonly used for automatic feature representation of compounds and proteins. These models effectively capture essential CPI information by embedding neighboring node features into the central node when learning feature representations.

GraphDTA [29] represented drugs as molecular graphs using four GNN variants (i.e., GCN, graph attention network (GAT), graph isomorphic network (GIN) and combined GAT-GCN), respectively. Protein features was obtained using a multilayer 1D CNN. The results showed that GNNs not only predict drug-target affinity better than non-deep learning models, but also outperform competing deep learning methods. CPI-CNN [31] employed GNNs with r-radius subgraphs and CNN to learn low-dimensional vector representations of molecular graphs and protein sequences, respectively. The approach achieved competitive performance with significant advantages especially on unbalanced datasets. Graph-CNN [32] employed two graph autoencoders, one for modeling molecular graph structures and the other for representing protein pockets. This approach enabled the extraction of features from both pocket graphs and 2D ligand graphs. Notably, Graph-CNN demonstrated its ability to accurately capture protein–ligand-binding interactions without relying on target-ligand complexes. Recently, DGraphDTA [33] utilized the structural information of compounds and proteins to construct molecule graphs and protein contact maps, respectively. By connecting these representations, DGraphDTA achieved reliable affinity prediction and exhibited robustness and generalizability when tested on benchmark datasets.

Currently, Transformer has demonstrated remarkable capabilities in feature learning. Its success mainly attributed to its attention mechanism, which allows it to capture implicit semantic correlations. These correlations represent association of semantic information between tokens, such as words, phrases, and other linguistic units, and are measured by the possibility of their co-occurrence in the context. In fact, the advantage of the attention mechanism can be extended from sequences to molecules. Essentially, any compound or protein can be regarded as a sequence of nodes with a specific order and spatial arrangement. If these sequences are considered as sentences, then the nodes are the tokens that form the sentence. Just like the semantic correlations between tokens, there inevitably exists a similar semantic correlation between nodes. In this context, the attention mechanism describes how nodes in molecular sequences are related to each other based on their node attributes.

For example, in the case of compounds, carbon (C) atoms generally tend to form multiple covalent bonds with C, hydrogen (H), oxygen (O), and nitrogen (N) atoms, which indicates that C atoms are semantically more related to them than to other atoms. As determined by the properties of the atoms themselves, the similarity between C and C atoms is higher than that between C and S atoms. Therefore, after the attention mechanism learning, when the atoms are projected to the word vector space, the distance between C and C atoms is closer than that between C and sulfur (S) atoms, indicating that the semantic correlation between the former is stronger. Similarly, there are also semantic correlations among amino acids. This prevalent semantic correlation reflect the importance of nodes in the sequence. Transformers can capture these correlations to generate feature vectors, also known as semantic features, that describe the semantic information of nodes.

However, it is insufficient to rely solely on the semantic information of nodes in molecules for predicting CPI, which would ignore important structural features. Structural features of compounds and proteins include but are not limited to adjacencies, distances, and angles between nodes. To some extent, most GCN-based CPI methods exploit the adjacency of nodes within a molecule to aggregate features but neglect the semantic features of nodes.

The objective of this study is to develop an interpretable model, termed Graph Transformer CPI (GraphsformerCPI), which integrates the semantic and structural features of compounds and proteins for CPI prediction. Inspired by attention mechanisms and graph convolutional networks, we extend the application of attention mechanisms from sequences to molecules to deepen our understanding of the complex relationship between compounds and proteins and improve the accuracy of CPI prediction. To overcome the limitation of using only a single sequence or graph representation, this research aims to effectively encode the topological structure of molecules and capture the semantic correlations between nodes to bridging the gap between these two representations. By combining the strengths of attention mechanisms and the graph structural information, our GraphsformerCPI model offers a promising approach to predicting CPIs.

The main contributions of this research are summarized as follows:

- We present a novel framework that integrates attention mechanisms and structural features of molecular graphs to predict CPI by effectively combining the one-dimensional sequences and two-dimensional structures of compounds and proteins.
- The proposed model focuses on the relationship features between compounds and proteins, and utilizes a dual attention mechanism to achieve sufficient fusion of their node features.
- Utilizing structure-enhanced attention mechanism can improve the CPI prediction performance and provide the model with good interpretability, which is not possessed by many black-box deep learning models.

2 Materials and Methods

2.1 GraphsformerCPI Overview

The overall architecture of GraphsformerCPI is illustrated in Fig. 1. The model is a stack of multiple identical encoder–decoder layers, with each layer comprising two encoders and one decoder. The encoders employ self-attention mechanisms to autonomously learn the semantic features of compounds and proteins. Subsequently, these acquired semantic features are integrated with their respective graph features to establish comprehensive representations. The decoder computes the affinity probabilities between different tokens of compound atoms and protein residues, thereby identifying the global attention between atoms and residues. Ultimately, a predefined predictor is used to determine the relationship between compounds and proteins.

2.2 Structure-Enhanced Encoder

The native Transformer encoder serves as a module for processing sequential data, exhibiting remarkable performance in calculating the semantic similarity among internal nodes within a sequence. This capability is equally valuable in the study of compounds and proteins. However, compounds and proteins possess intricate spatial structural features that are vital for comprehending their function, properties, and interactions. In order to enable the encoder to capture structural features, we have improved the original encoder.

The structure-enhanced encoder consists of a multi-head self-attention mechanism (MHA) sub-layer and a position-wise feed-forward network (FFN) sub-layer. In the self-attention mechanism, symmetrically normalized Laplacian matrices and Euclidean distance are employed to capture spatial attention between nodes, as shown in Fig. 2. This enhancement enable comprehensive consideration of semantic and spatial structural features of compounds and proteins, thereby augmenting the model's understanding and predictive capabilities regarding their function and properties.

The Laplacian matrix plays a role similar to that in GCN. The average aggregation method assigns appropriate weights to nodes, where nodes with higher degrees transmit less information to each edge to prevent excessive influence on neighboring nodes. Euclidean distance is one of the important features in describing the molecular structure of compounds and proteins. Changes in the distance between nodes significantly impact the molecule's properties. In general, when the Euclidean distance between C_β atoms is less

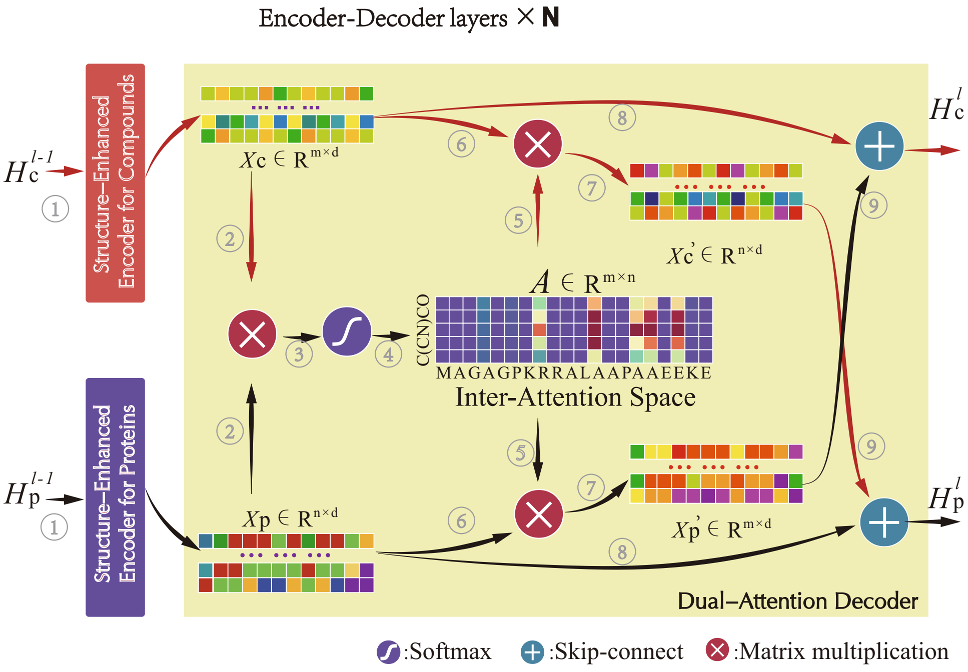


Fig. 1 Schematic overview of the lth encoder–decoder sub-layer in GraphsformerCPI. The GraphsformerCPI model is composed of N identical layers, each consisting of two encoders and one decoder. The hidden features H_c^{l-1} and H_p^{l-1} of the compounds and proteins learned in the previous layer are used as inputs to the current layer. The compound encoder learns structure-enhanced deep features of compounds by modeling and encoding different compound atoms using self-attention mechanism. Similarly, the protein encoder learns structure-enhanced deep features of the proteins by modeling and encoding residue information. The dual-attention decoder combines the features learned by the encoders to extract interactions between compound atoms and protein residues. This affinity probability is extracted through a inter-attention mechanism. Finally, the learned relational features of the decoder are skip-connected with the output features of the encoder, generating the output H_c^l and H_p^l of the layer. Hence, the GraphsformerCPI model can progressively learn complex features of compound–protein interactions layer by layer. Notes that A denotes the space of inter-attention between compound atoms and protein residues, m represents the number of atoms in the compound, n represents the number of residues in the protein, and d represents the dimension of the feature vectors

than 8$\AA$, it indicates that there is an interaction between two residues. The strength of this interaction increases as the distance between residues decreases, and vice versa.

To simplify the description of the relationship between interaction strength and Euclidean distance, we use the exponential function e^{-x} to transform the distance. This transformation projects smaller distances to larger interaction strengths, thus providing a more accurate depiction of interactions within molecular structures.

More specifically, let us consider compounds as an example. Assume that any compound consists of N atoms, the model performs h times attention function in parallel, Q_i, $K_i \in \mathbb{R}^{N\times d_K}$, and $V_i \in \mathbb{R}^{N\times d_V}$ are the projection matrices for the ith head, respectively, and d is the dimension of the feature vector. The symmetric normalized Laplacian matrix for the compound is denoted as $L_{sym} \in \mathbb{R}^{N\times N}$, and the Euclidean distance matrix is represented as $Dist \in \mathbb{R}^{N\times N}$. The formulation of the multi-head self-attention sub-layer in the enhanced encoder is as follows:

$$Q_i = QW_i^Q, \quad K_i = KW_i^K, \quad V_i = VW_i^V \tag{1}$$

$$A(Q_i, K_i) = stack(softmax(\frac{Q_iK_i^T}{\sqrt{d_K}}), L_{sym}, f(Dist)) \tag{2}$$

$$\hat{A}(Q_i, K_i) = A(Q_i, K_i)W^{(1\star 1)} \tag{3}$$

$$Attn(V_i) = \hat{A}(Q_i, K_i)V_i \tag{4}$$

$$MHA = [Attn(V_1), ..., Attn(V_h)]W^o \tag{5}$$

$$O_{MHA} = Q + MHA \tag{6}$$

where the projections are parameter matrices W_i^Q, $W_i^K \in \mathbb{R}^{d\times d_K}$, $W_i^V \in \mathbb{R}^{d\times d_V}$, $W^{(1\star 1)} \in \mathbb{R}^{3\times 1}$ and $W^O \in \mathbb{R}^{hd_v\times d}$, and [,] represents the concatenation. We calculate element-wise $f(x) = exp(-x)$ on the distance matrix.

Utilizing the input node features, the encoders compute semantic weights between nodes through a multi-head self-attention mechanism, generating a weight matrix that reflects the similarity of node features. The normalized semantic weight matrix, as illustrated in Eq. 2, is obtained

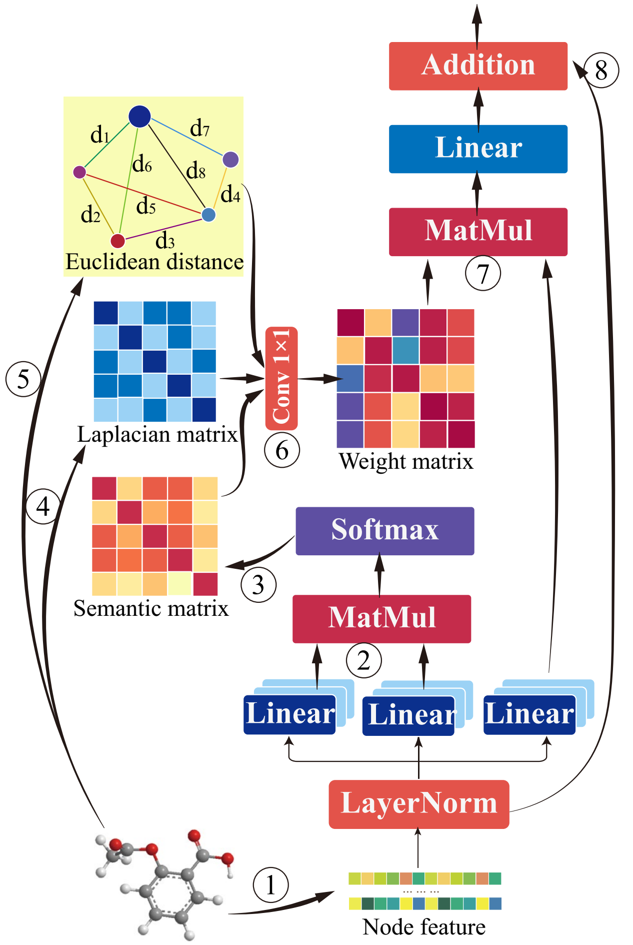


Fig. 2 Schematic overview of the multi-head attention (MHA) sublayer of the GraphsformerCPI

by applying a *softmax* function and then stacked with the Laplacian matrix and the Euclidean distance matrix to form a three-dimensional feature weight matrix denotes as $A(Q_i, K_i)$. Each channel within this weight matrix is an independent feature space that captures the correlation of nodes in different spaces.

Since it is difficult to directly project node features into this high-dimensional feature space, a 1×1 convolution operation is employed to fuse the weights by channel, as shown in Eq. 3. This operation reduces the dimension of the $A(Q_i, K_i)$ matrix and introduces nonlinear transformations to generate a composite feature space denoted as $\hat{A}(Q_i, K_i)$.

Finally, as depicted in Eq. 6, the feature vectors are skip-connected with the original node feature vectors. It is important to note that the normalization process is not explicitly mentioned in the above equations. In practice, they are incorporated at the outset of the *MHA* and *FFN*, as depicted in Fig. 2, to improve the performance of GraphsformerCPI.

2.3 Dual Attention Decoder

Merely connecting the hidden features of compounds and proteins obtained by the encoder is insufficient for accurately predicting their interactions. This approach overlooks the essential relational features between compounds and proteins, which are crucial for comprehending their interactions. To overcome this limitation, we propose a decoder with dual attention.

The decoder leverages attention mechanisms to construct feature spaces for compound atoms and protein residues. By individually projecting compound atoms and protein residues into these feature spaces, we establish their inter attention. This attention reveals the intrinsic relationship between compound atoms and protein residues, enhancing the capacity of compounds and proteins to recognize each other. Through this dual attention mechanism, we achieve more accurate predictions of compound–protein interactions.

Let $X_C = [c_0^T, c_1^T, ..., c_{N-1}^T]^T \in \mathbb{R}^{N\times d}$ denote a sequence of compound vectors and $X_P = [p_0^T, p_1^T, ..., p_{M-1}^T]^T \in \mathbb{R}^{M\times d}$ denote a sequence of protein vectors generated by the l-th layer encoder, where N and M represent the number of atoms in the compound and the number of residues in the protein, respectively. Here, $c_i^T \in \mathbb{R}^{1\times d}$ and $p_i^T \in \mathbb{R}^{1\times d}$ represent the atomic features and the residue features, both possessing the same dimension d. The dual affine transformation of the decoder at the lth layer can be expressed as follows:

$$A(X_C, X_P) = softmax(\frac{X_C X_P^T}{\sqrt{d/h}}) \tag{7}$$

$$Attn(X_C) = A(X_C, X_P)X_P \tag{8}$$

$$Attn(X_P) = A(X_P, X_C)X_C \tag{9}$$

$$H_C^l = X_C + Attn(X_C) \tag{10}$$

$$H_P^l = X_P + Attn(X_P) \tag{11}$$

In Eq. 7, an attention space is created between atoms and residues. The dual attention decoder assigns different weights to various parts of the compound and protein. In Eq. 8, the protein feature sequences are projected into the attention space A, with greater attention given to the parts of the compound that have a stronger impact on binding. Similarly, in Eq. 9, the compound features are projected into the dual space of A, highlighting the parts of the proteins that have significant effects.

The dual attention decoder project node-level features of compounds and proteins into each other's feature space, respectively. This enables us to model compound–protein relationships and capture depth features related to the interaction, rather than representing such interactions through simple feature concatenation or summation. Additionally, skip connections are employed in the dual attention decoder to fuse learned relationship features with the hidden node features to prevent model overfitting and gradient vanishing, as shown in Eqs. 10 and 11.

2.4 Prediction module

Assuming that the embedding features of compounds and proteins generated by the last encoder–decoder layer are denoted as $H_C \in \mathbb{R}^{N\times d}$ and $H_P \in \mathbb{R}^{M\times d}$, respectively, where N represents the number of atoms in the compound, M represents the number of residues in the protein, and d represents the dimension of the embedding space.

The predictor calculates the node weights for compound atoms and protein residues, and then derives the graph-level feature vectors of compounds and proteins through weighted summation. Specifically, for the node weights of compound atoms, the calculation formula is:

$$\alpha_i^C = \sigma(h_i^C W^C) \tag{12}$$

Here, h_i^C represents the embedding feature of the i-th atom in the compound, $W^C \in \mathbb{R}^{d\times 1}$ is learnable parameter, and $\sigma(\cdot)$ represents the activation function. Similarly, for the node weights of protein residues, the calculation formula is:

$$\alpha_j^P = \sigma(h_j^P W^P) \tag{13}$$

Here, h_j^P represents the embedding feature of the j-th residue in the protein, $W^P \in \mathbb{R}^{d\times 1}$ is learnable parameter.

The graph-level feature vectors for compounds and proteins are calculated as follows:

$$v_C = \sum_{i=1}^{N} \alpha_i^C h_i^C \tag{14}$$

$$v_P = \sum_{j=1}^{M} \alpha_j^P h_j^P \tag{15}$$

Here, $v_C \in \mathbb{R}^d$ and $v_P \in \mathbb{R}^d$ represent the graph-level feature vectors for compounds and proteins, respectively.

Finally, to predict the relationship between compounds and proteins, we concatenate these two feature vectors and utilize a fully connected layer for prediction:

$$\hat{y} = \mathrm{softmax}([v_C, v_P]W_O) \tag{16}$$

Here, $W_O \in \mathbb{R}^{2d\times(2or1)}$ is learnable parameter, $[v_C, v_P]$ represents the concatenation of the feature vectors for compounds and proteins, and $\hat{y}$ represents the final prediction result, which is transformed into probability via the *softmax* function.

In the case of a classification task, GraphsformerCPI employs cross-entropy loss as its objective function and minimizes it in the following way:

$$L = \frac{1}{N}\sum_{i}^{N} -[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{17}$$

Where $y_i \in \{0, 1\}$ denotes the label and N represents the number of samples.

In the case of a regression task, the model employs mean squared error (MSE) as its objective function and minimizes it in the following manner:

$$L = \frac{1}{N}\sum_{i}^{N}(y_i - \hat{y}_i)^2 \tag{18}$$

2.5 Feature Representation

2.5.1 Compound Representation

In GraphsformerCPI, compounds are initially described by SMILES strings. We parse compound descriptors from SMILES and extract their raw features as vectors that can be processed by the encoder.

Specifically, the compound SMILES dataset is analogized to a "corpus" consisting of "sentences" (in this case, atoms) of varying lengths. We tokenize the compounds based on atoms, truncating compounds longer than a predefined length and padding compounds shorter than the predefined length. We encode the attributes of the atoms using one-hot encoding, obtaining 78-dimensional raw features of the compound $\mathcal{X}_C \in \mathbb{R}^{N\times 78}$. The selected atom attributes in this study are the same as those in GraphDTA and are listed in Table 1.

To incorporate spatial information, we construct molecular graphs with atoms as nodes and chemical bonds as edges. We utilize the RDKit [34] Python package to retrieve atomic attributes and spatial three-dimensional coordinates based on the

Table 1 The details of atom features

No	Feature	Description	Dimension
1	Atom symbol	One hot encoding of the atom type	44
2	Atom degree	The number of directly-bonded neighbors	11
3	Total Hs	The number of total Hs on the atom	11
4	Implicit Hs	The number of implicit Hs on the atom	11
5	Aromatic	Whether the atom is aromatic	1

adjacency relationship of nodes in the molecular graph. Subsequently, we calculate the Euclidean distance between atoms to capture molecular structure information. Each element of the Euclidean distance matrix is subjected to computation of the negative exponential power with base e, as described in Sect. 2.2. To account for the impact of structural features on nodes, we employ a symmetric normalized Laplacian matrix with self-loops in graph construction, similar to GCN. The formula for the normalized Laplacian is shown below:

$$L_{sym} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$$

where $\tilde{A}$ is the adjacency matrix with self-loops, and $\tilde{D}$ is the degree matrix with respect to $\tilde{A}$.

2.5.2 Protein Representation

Protein representation primarily involves generating residue identity descriptors and spatial structure features. By treating residues as nodes rather than atoms, we can effectively mitigate the computational challenges posed by excessive data. Similar to compound representation, the first step is to obtain protein graphs, also known as contact maps, which indirectly represent three-dimensional structural features with a two-dimensional residue matrix. Therefore, acquiring highly reliable protein structure data is essential.

Due to the limited number of known protein structures, many protein structures remain undetermined. Therefore, it is not always feasible to obtain reliable tertiary structures and there are obstacles to directly using existing protein structure data. We utilized the protein structures predicted by AlphaFold2 [35]. AlphaFold2 has achieved significant advancements in protein structure prediction, contributing to more accurate and efficient prediction of CPI. The AlphaFold database (https://alphafold.ebi.ac.uk/) provides protein structures for the human proteome and 20 other major organisms. The PDB files of proteins are downloaded, residue information is parsed, and Euclidean distance matrices and contact maps are constructed to represent structural information.

Suppose the protein sequence has a length of L, the predicted contact map M and Euclidean distance matrix $Dist$ both have L rows and L columns. The elements of the contact map matrix, M_{ij}, indicate whether the corresponding residue pair (i, j) is in contact. The elements of the Euclidean distance matrix, $Dist_{ij}$, represent the Euclidean distance between the corresponding residue pairs, as follows:

$$Dist_{ij} = \begin{cases} \delta_{ij} & \text{if } i \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

$$M_{ij} = \begin{cases} 1 & \text{if } \delta_{ij} < 8\mathring{A}, \\ 0 & \text{otherwise.} \end{cases}$$

where δ_{ij} denotes the Euclidean distance between the C_β atoms of residues i and j. Residues are considered to be in contact if $\delta_{ij} < 8\mathring{A}$. It is worth noting that both the distance matrix and contact map are non-negative, symmetric square matrices. The diagonal elements of the distance matrix are 0, while those of the contact map are 1.

After obtaining the Euclidean distance matrix and contact map, GraphsformerCPI extracts the physicochemical properties of amino acids as node features for residues. These properties are determined by the side chain R-groups of amino acids, including polarity, electrification, aromaticity, hydrophobicity, and more. To simplify their representation, we adopt the amino acid properties documented in previous literature [33, 36]. These selected properties are encoded using the one-hot mechanism and concatenated into a protein feature matrix $\mathcal{X}_P \in \mathbb{R}^{L\times 50}$, where L denotes the length of the protein sequence. Detailed features of residues are listed in Table 2.

Table 2 Details of residue features

No	Feature	Description	Dimension
1	Residue symbol	One-hot encoding of the residue symbol	21
2	Aliphatic	Aliphatic residue or not	1
3	Aromatic	Aromatic residue or not	1
4	Polar neutral	Whether it is a polar neutral residue	1
5	Polar positive charge	Whether it is a polar positive charge residue	1
6	Polar negative charge	Whether it is a polar negative charge residue	1
7	Molecular weight	Molecular weight of amino acids	1
8	pK_a	Negative log of the dissociation constant of –COOH group	1
9	pK_b	Negative log of the dissociation constant of $–NH_3$ group	1
10	pK_x	Negative log of the dissociation constants of other groups	1
11	pl	Isoelectric point	1
12	Hydrophobicity-1	Hydrophobicity of residue at pH 2	1
13	Hydrophobicity-2	Hydrophobicity of residue at pH 7	1
14	Meiler	Meiler feature	7
15	Kidera	Kidera feature	10

2.6 Datasets

The performance of the proposed GraphsformerCPI model is evaluated using four CPI datasets: human, *C. elegans* [31], Davis [37] and KIBA [38]. These datasets are commonly used as benchmark datasets for evaluating binding affinity prediction [23, 26, 29, 33, 39].

The human and *C. elegans* datasets are employed to assess the classification performance of the model. These datasets were created by Liu et al. [40]. The positive samples came from DrugBank 4.1 [41], Matador [42], and STITCH 4.0 [43]. To ensure high-confidence negative samples of compound-protein pairs, various compound and protein resources were integrated into a systematic screening framework. The prediction task is to determine whether a given compound and protein pair interact. The model predicts a binary label indicating interaction (1) or no interaction (0) between the compound and protein. To ensure a fairness in the comparison, a balanced dataset is used following the protocol described in reference [31]. The positive to negative sample ratio is maintained at 1:1. The balanced human dataset consists of 3369 positive interactions involving 1052 unique compounds and 852 unique proteins. The balanced *C. elegans* dataset comprises 4000 positive interactions with 1434 unique compounds and 2504 unique proteins.

The Davis and KIBA datasets are used to evaluate the regression performance of the proposed model. The Davis dataset, collected from clinical trials of kinase protein families and related inhibitors [37], includes 442 proteins, 68 compounds, and binding affinity values (dissociation constants, (K_d)) for protein–compound pairs [44]. The K_d values in the Davis dataset are transformed into logspace (pK_d) using the following formula:

$$pK_\mathrm{d} = -\log_{10}\left(\frac{K_\mathrm{d}}{10^9}\right)$$

The KIBA dataset was generated from several sources of kinase inhibitor bioactivities [38]. Different bioactivities, such as K_i, K_d and IC_{50}, were integrated by a method called KIBA. The dataset initially contained 467 targets, 52,498 drugs and 246,088 observations. After filtering by [44], the dataset consists 229 unique targets, 2111 unique drugs and 118,254 target–drug-binding affinities measured by KIBA values.

When evaluating the model performance, the human and *C. elegans* datasets are randomly shuffled and divided into six parts. One part is used as the test set, while the remaining portions serve as training and validation sets for five-fold cross-validation. For the Davis and KIBA datasets, all five training and test sets are utilized following the methodology described in reference [33].

3 Results and Discussion

3.1 Evaluation Metrics

Several evaluation metrics commonly used in the benchmark evaluation are employed in this study, including Concordance index (CI) [45], MSE, and Pearson correlation coefficient [46] for regression performance, as well as the area under the receiver operating characteristic curve (AUC), precision, and recall for classification performance.

The CI measures the extent to which the predicted CPI affinity values are ordered in a manner consistent with the true values, and it is calculated using the following formula:

$$CI = \frac{1}{Z}\sum_{\delta_i > \delta_j} h(p_i - p_j)$$

where p_i represents the predicted value of the larger affinity δ_i, p_j represents the predicted value of the smaller affinity δ_j, and Z is a normalization constant. The step function $h(x)$ is defined as:

$$h(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0.5 & \text{if } x = 0 \\ 0 & \text{if } x < 0 \end{cases}$$

MSE is a widely used measure of the disparity between predicted values and true values. A smaller mean square error indicates that the predicted values closely align with the true values. The formula for MSE is described by Eq. 18.

The Pearson correlation coefficient (r) is employed to quantify the linear relationship between a predicted value dataset and a true value dataset. A positive correlation implies that an increase in variable x corresponds to a corresponding increase in variable y, while a negative correlation suggests that an increase in x leads to a decrease in y. The Pearson correlation coefficient is given by:

$$r = \frac{\sum_i^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i^N (x_i - \bar{x})^2 \sum_i^N (y_i - \bar{y})^2}}$$

where $\bar{x}$ denotes the mean of the vector x_i and $\bar{y}$ denotes the mean of the vector y_i.

AUC is a critical metric for evaluating classifier performance and measures the overall performance of the classifier. It is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at various classification thresholds. AUC values ranges between 0 and 1, with values closer to 1 indicating superior classifier performance.

Precision measures the proportion with which the model can correctly identify positive interactions (True Positive) among all predicted positive interactions. A higher Precision

Table 3 The hyperparameter settings

Hyper-parameter	Setting
Epoch (classification)	500
Epoch (regression)	2000
Batch size (classification)	32
Batch size (regression)	64
Optimizer	Adam
Learning rate	0.001

score indicates that the model can more accurately identify positive interactions, reducing false positive predictions and increasing prediction reliability.

Recall measures the completeness with which the model can detect True Positive interactions among all interactions present in the validation dataset. A higher Recall score indicates that the model is better able to detect true positive interactions, reducing false negative predictions and enhancing model sensitivity.

3.2 Hyperparameter Settings

Setting and optimizing hyperparameters is a crucial and time-consuming aspect of training deep learning model. GraphsformerCPI utilizes a five-fold cross-validation strategy, and the model encompasses several pivotal parameters that require meticulous selection. While some of these parameters are designated empirically, others are established by referencing existing models. The ranges of hyperparameters utilized are detailed in Table 3.

3.3 Impact of Model Depth

The foundational architecture and the depth of the model are key factors influencing prediction performance. Here, we explore the impact of varying the number of layers on the performance of GraphsformerCPI. It is important to note that simply increasing the number of layers does not guarantee improved performance. On the contrary, an excessive number of layers can lead to memory overflow and strain the computational capabilities of the device.

In our model, the number of layers is varied between 2 and 4. As shown in Table 4, GraphsformerCPI consistently demonstrates high classification performance when utilizing 2–4 layers, yielding average values exceeding 0.9 for all three metrics. Notably, setting the number of layers to 4 yields the most favorable outcomes, with AUC and recall rates of 0.990 and 0.979, respectively, on the human dataset, and AUC and recall rates of 0.989 and 0.959, respectively, on the *C. elegans* dataset. Furthermore, as illustrated in Table 5, the model also demonstrates satisfactory performance on the regression dataset, with an improvement in performance metrics such as MSE, Pearson correlation coefficient, and CI index as the number of layers increases to four.

It is apparent that increasing the number of layers in GraphsformerCPI enhances the model's performance. By incorporating more encoders, the model comprehensively and diversely learns compound and protein self-features, capturing deep features at the node-level for compounds and proteins. Moreover, an increasing the number of layers facilitates deeper interactions between compound and protein node-level features. These enriched features with diverse emphases are extensively fused through multi-layer interactions, improving the likelihood of accurately predicting compound–protein affinities.

However, there exists a trade-off. As observed in Table 4, a slight decrease in accuracy with an increase in the number of layers implies a potential for overfitting on small datasets. Therefore, a balance must be struck between the model's performance and the number of layers employed.

3.4 Impact of Dropout Rate

Figures 3 and 4 illustrate the impact of different dropout rates on the performance of GraphsformerCPI on different

Table 4 Performance of various module layers on the human and *C. elegans* datasets

Human				*C. elegans*			
No. of layers	AUC	Precision	Recall	No. of layers	AUC	Precision	Recall
2	0.914	0.890	0.942	2	0.936	0.936	0.933
3	0.966	0.970	0.960	3	0.947	0.948	0.942
4	0.990	0.952	0.979	4	0.989	0.935	0.959

Table 5 Performance of various module layers on the Davis and KIBA datasets

Davis				KIBA			
No. of layers	MSE	Pearson	CI	No. of layers	MSE	Pearson	CI
2	0.559	0.653	0.750	2	0.450	0.685	0.776
3	0.304	0.774	0.849	3	0.299	0.781	0.867
4	0.212	0.802	0.908	4	0.141	0.895	0.913

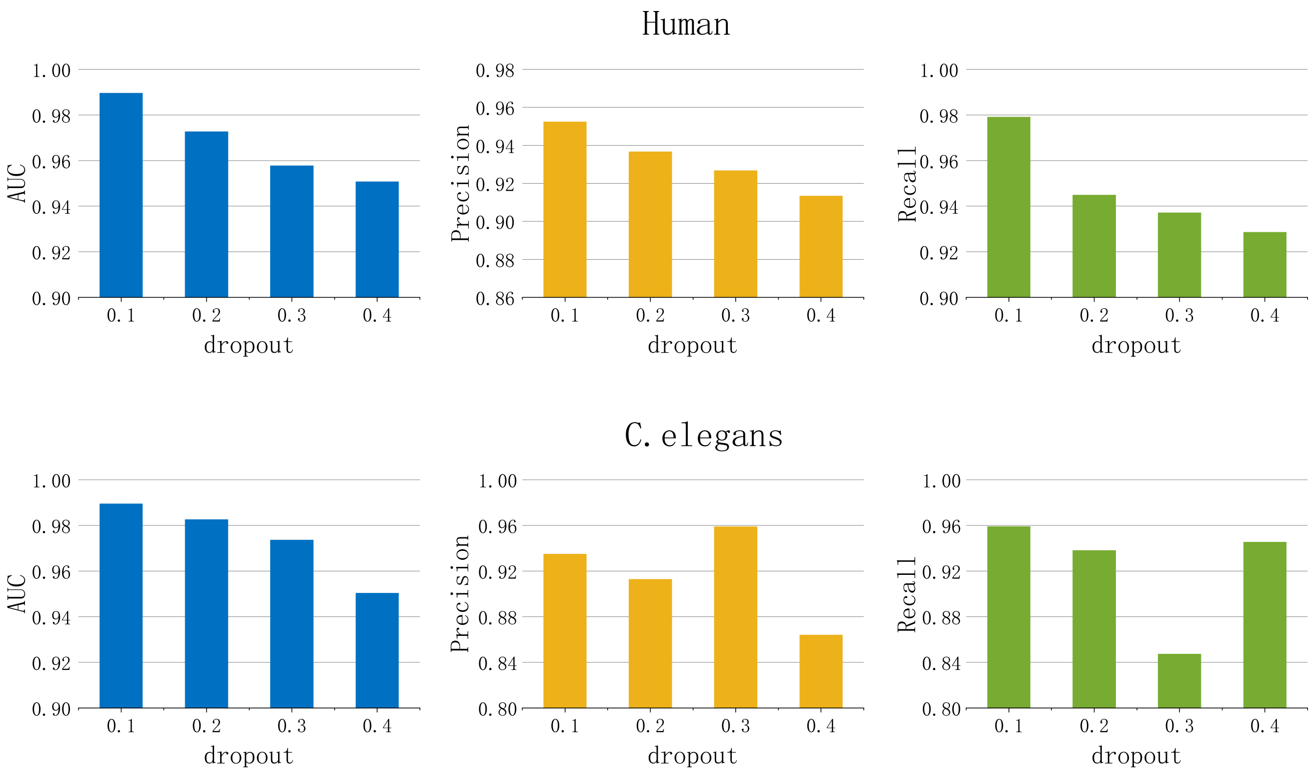


Fig. 3 Performance of various dropout rates on the human and *C. elegans* datasets

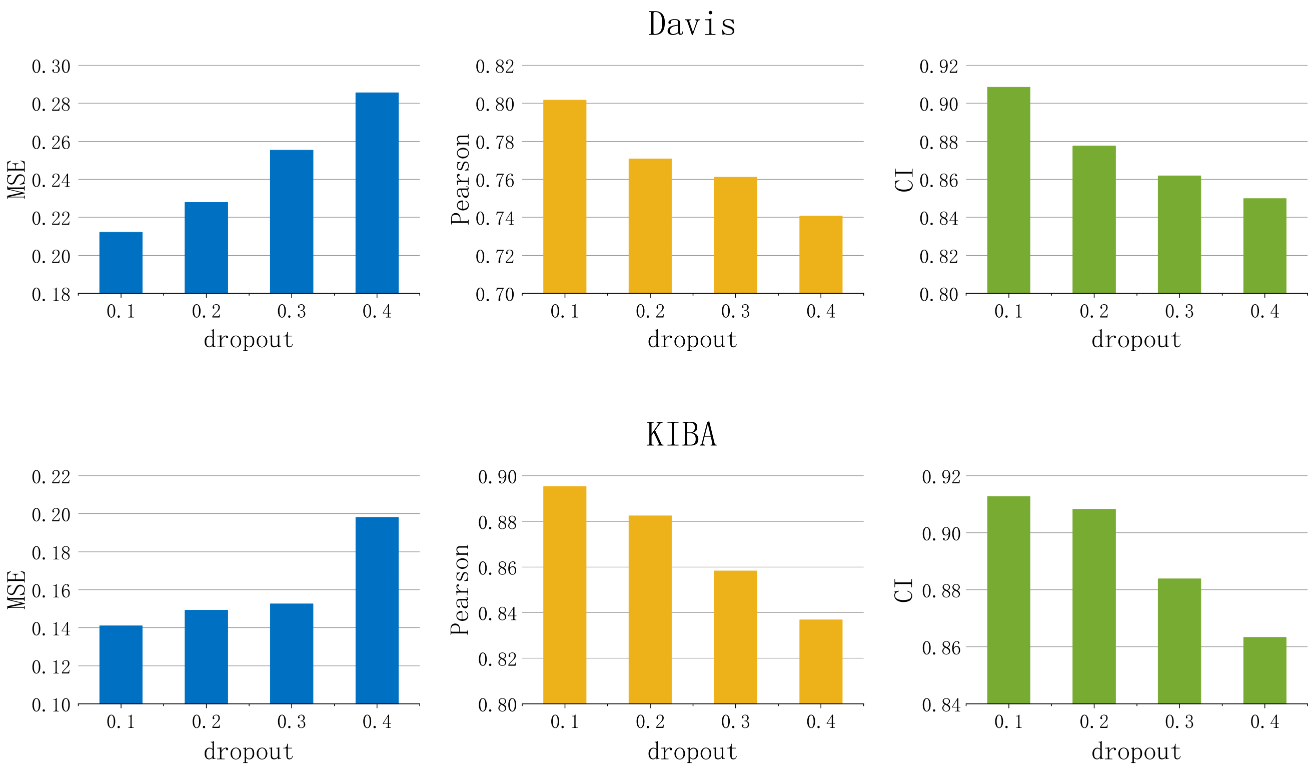


Fig. 4 Performance of various dropout rates on the Davis and KIBA datasets

datasets. In our experiments, we varied the dropout rates between 0.1 and 0.4 to mitigate the risk of underfitting caused by excessively high dropout rate. The results reveal that as the dropout rate increases, there is a decreasing trend in AUC, precision, and recall on the human dataset. Similarly, the *C. elegans* dataset demonstrates a declining pattern in AUC, while precision and recall exhibit fluctuations. Notably, at a dropout rate of 0.3, precision reaches 0.959, while recall decreases to 0.847. For the Davis and KIBA regression datasets, MSE increases with the dropout rate, whereas the Pearson correlation coefficient and CI decreased slightly.

Based on these findings, it can be concluded that the dropout rate significantly impacts the performance of GraphsformerCPI. Setting the dropout rate too high can result in inferior performance. Therefore, to achieve optimal performance, it is recommended to avoid excessively high dropout rates. In summary, GraphsformerCPI demonstrates optimal performance when the dropout rate is set to 0.1.

3.5 Comparison with Other Methods

To further evaluate the performance of our model, we conducted comparison with several classic machine learning models, including K-nearest neighbor (KNN), Random Forest (RF), L2-logistic (L2), and Support Vector Machine (SVM), as well as state-of-the-art (SOTA) deep learning models such as GCN, CPI-GNN [31], TransformerCPI [26], GraphDTA [29], CoaDTI [39], KronRLS [47], SimBoost [44], DeepDTA [23], WideDTA [48], and DGraphDTA [33]. In this comparison, GraphsformerCPI adopts a 4-layer structure, while TransformerCPI, GraphDTA, CoaDTI, and DGraphDTA followed the default settings described in the literature.

Table 6 presents results on the human and Celegans datasets. Notably, GraphsformerCPI achieves higher performance compared to the benchmark models. On the human dataset, GraphsformerCPI achieves an AUC of 0.990, precision of 0.952, and recall of 0.979, surpassing the classical models. Furthermore, when compared to the current state-of-the-art models, GraphsformerCPI shows improvements of 1.6%, 0.5%, and 5.3% in terms of AUC, precision, and recall, respectively. Similarly, on the *C. elegans* dataset, GraphsformerCPI exhibits improvements of 1.0% and 1.8% in AUC and recall, respectively.

The comparison results on regression datasets are presented in Table 7. These models employ diverse algorithms for extracting compound and protein features, such as Smith–Waterman (S-W), PubChem Sim, CNN, GCN, as well as sequence-based methods including PDM and LMCS [48]. GraphsformerCPI demonstrates competitive performance against the benchmark models on various metrics. For instance, when evaluating on the Davis dataset, GraphsformerCPI achieves an average CI value of 0.910 with a standard deviation of 0.013, indicating a mean increase of 2.1% in CI compared to the baseline performance. Similarly, on the KIBA dataset, GraphsformerCPI demonstrates an average CI value of 0.913 with a standard deviation of 0.021, corresponding to a mean improvement of 3.3% in CI. Moreover, the MSE indicators for GraphsformerCPI on these two datasets are 0.212 ± 0.019 and 0.141 ± 0.016, representing mean improvements of 1.4% and 7.2%, respectively.

These results demonstrate that incorporating spatial structural information into attention mechanisms empowers the model with strong learning capabilities, enabling it to capture not only the semantic features between nodes but also the influence of structure on feature learning. By employing multi-layer dual attention, the model effectively integrates relationship features between compounds and proteins with node features, resulting in a more comprehensive representation of affinity for CPI prediction. This approach allows the model to learn the intricate internal and mutual information of proteins and compounds, ultimately improving its performance compared to other deep learning models.

Table 6 Comparison results of GraphsformerCPI and baselines on the human and *C. elegans* datasets

Methods	human			*C. elegans*		
	AUC	Precision	Recall	AUC	Precision	Recall
KNN	0.860	0.927	0.798	0.858	0.801	0.827
RF	0.940	0.897	0.861	0.902	0.821	0.844
L2	0.911	0.913	0.867	0.892	0.890	0.877
SVM	0.898	0.916	0.921	0.890	0.776	0.807
CPI-GNN	0.966	0.915	0.917	0.965	0.919	0.923
CoaDTI	0.970	0.948	0.930	0.978	**0.950**	0.942
GCN	0.932 ± 0.040	0.849 ± 0.008	0.915 ± 0.010	0.963 ± 0.005	0.918 ± 0.007	0.920 ± 0.010
GraphDTA	0.955 ± 0.007	0.874 ± 0.040	0.908 ± 0.060	0.967 ± 0.006	0.915 ± 0.010	0.910 ± 0.018
TransformerCPI	0.972 ± 0.003	0.908 ± 0.006	0.917 ± 0.005	0.980 ± 0.004	0.945 ± 0.005	0.942 ± 0.006
GraphsformerCPI	**0.990 ± 0.002**	**0.952 ± 0.004**	**0.979 ± 0.004**	**0.989 ± 0.003**	0.935 ± 0.005	**0.959 ± 0.004**

Bold font indicates the best result in the column

Table 7 Comparison results of GraphsformerCPI and baselines on the Davis and KIBA datasets

Methods	Davis			KIBA		
	CI	MSE	Pearson	CI	MSE	Pearson
KronRLS	0.861 ± 0.023	0.375 ± 0.031	–	0.774 ± 0.061	0.420 ± 0.034	–
SimBoost	0.868 ± 0.020	0.279 ± 0.035	–	0.822 ± 0.040	0.254 ± 0.022	–
DeepDTA	0.863 ± 0.022	0.270 ± 0.024	0.802 ± 0.048	0.841 ± 0.031	0.223 ± 0.020	0.807 ± 0.046
WideDTA	0.861 ± 0.023	0.312 ± 0.027	0.793 ± 0.052	0.845 ± 0.030	0.211 ± 0.019	0.803 ± 0.048
GraphDTA	0.875 ± 0.017	0.234 ± 0.021	0.812 ± 0.044	0.879 ± 0.015	0.183 ± 0.017	0.823 ± 0.039
DGraphDTA	0.891 ± 0.010	0.215 ± 0.025	**0.867 ± 0.020**	0.884 ± 0.013	0.152 ± 0.014	**0.900 ± 0.022**
GraphsformerCPI	**0.910 ± 0.013**	**0.212 ± 0.019**	0.853 ± 0.026	**0.913 ± 0.021**	**0.141 ± 0.016**	0.891 ± 0.031

Bold font indicates the best result in the column

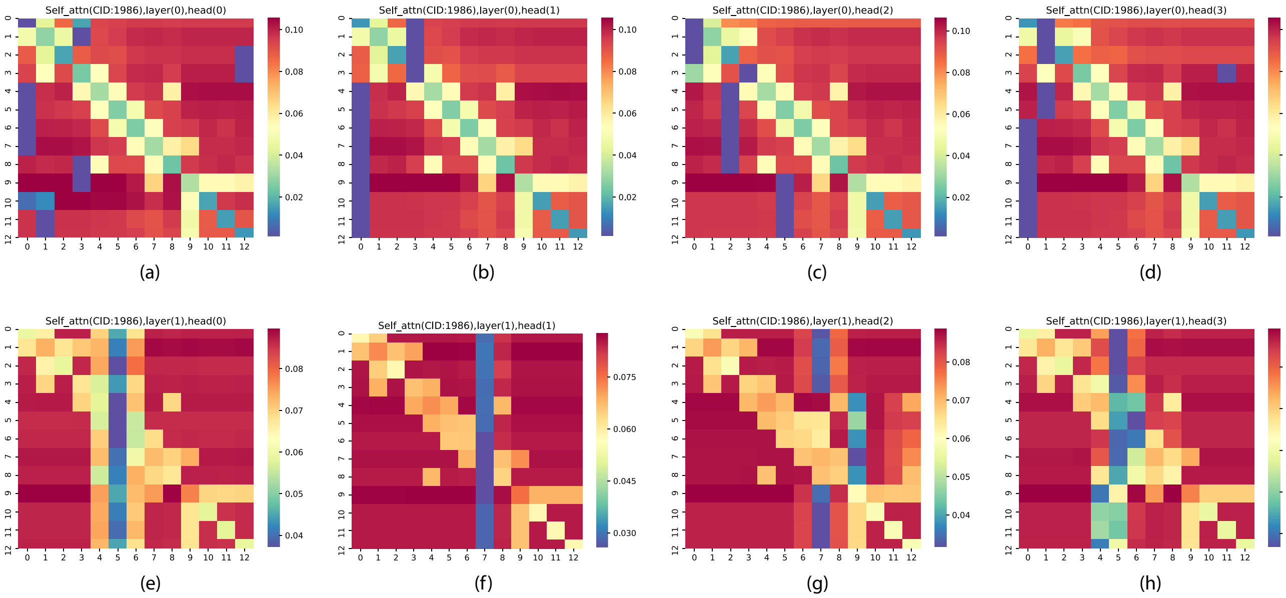
* Representations of proteins and compounds in methods: KronRLS: S-W & Pubchem Sim, SimBoost: S-W & Pubchem Sim, DeepDTA: CNN & CNN, WideDTA: PS + PDM & LS + LMCS, GraphDTA: GIN & 1D, DGraphDTA: GCN & GCN, GraphsformerCPI: Graph+Transformer & Graph+Transformer

3.6 Model Interpretation

To demonstrate the interpretability of our model, we present a case study using a two-layer GraphsformerCPI (with 4 attention heads) and randomly select a test record from the human dataset: acetazolamide (PubChem CID: 1986) and mitochondrial carbonic anhydrase 5B (UniProt ID: Q9Y2D0). The canonical smiles of acetazolamide is $CC(=O)NC1=NN=C(S1)S(=O)(=O)N$. It consists of 13 non-hydrogen atoms and acts as a non-competitive inhibitor of mitochondrial carbonic anhydrase 5B. Mitochondrial carbonic anhydrase 5B comprises 317 amino acids and is found in cells in the proximal tube of kidney, eye, and glial cells.

Figures 5 and 6 display the self-attention distribution of compounds and proteins. These distribution exhibit an overall diagonal symmetry but also local asymmetry. Each figure represent the fusion of the Laplacian matrix, Euclidean distance matrix, and semantic weight matrix. The color differences indicate that the model assigns different weights to local regions within the molecules, where nodes receive varying levels of attention. This demonstrates the effective fusion of structural and semantic features.

After two rounds of training, the model assigns lower attention to atom pairs formed by any two atoms located at positions 4th to 8th in the compounds, as shown in Figs. 5e–h. This is reasonable because these five atoms belong to a heterocycle, which exhibits stability in compound-protein interactions. This provides evidence for the consistency between attention scores and molecular structures.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 5 Visualization of self-attention matrices of acetazolamide in the 2-layer, 4-head GraphsformerCPI

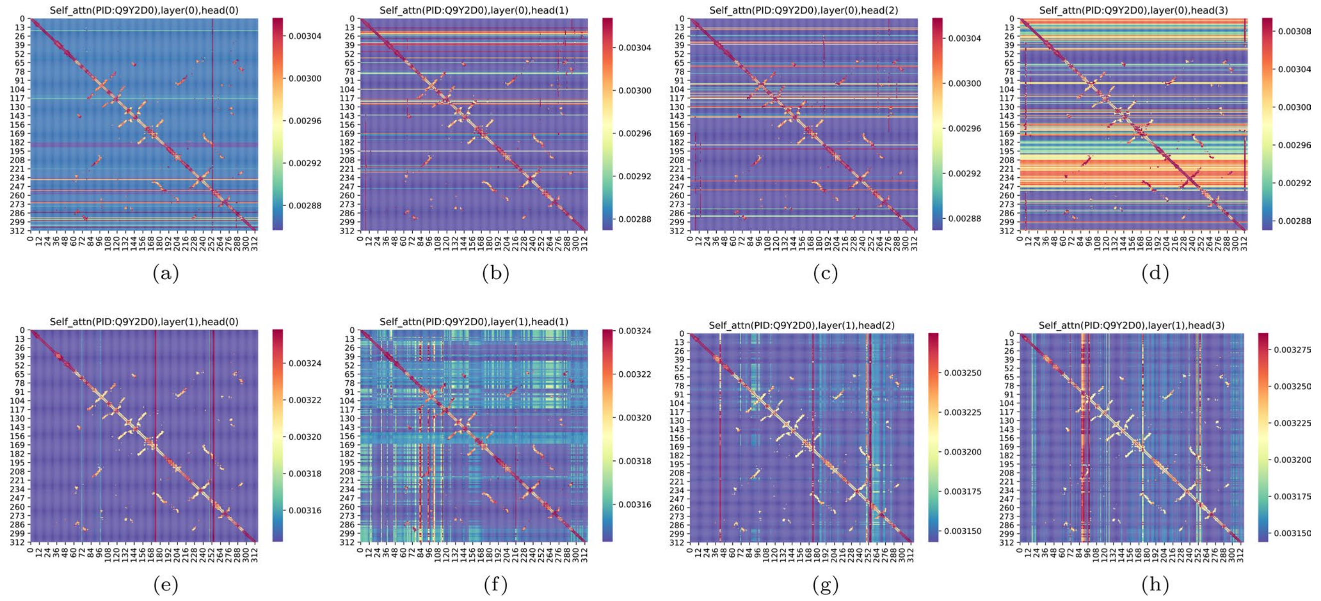


Fig. 6 Visualization of self-attention matrices of mitochondrial carbonic anhydrase 5B in the 2-layer, 4-head GraphsformerCPI

Furthermore, Fig. 5a–h reveal that the attention scores at diagonal positions are not necessarily higher than those at non-diagonal positions. This is due to the use of 1 × 1 convolutions in the encoder module, which promotes interaction between feature spaces and enhances the non-linear fusion ability of attention, instead of simply summing the Laplacian matrix, Euclidean distance matrix, and semantic weight matrix.

Figure 7 illustrates the inter-attention between acetazolamide atoms and mitochondrial carbonic anhydrase 5B residues in GraphsformerCPI. The *x*-axis represents protein residues, while the *y*-axis represents compound atoms. In Fig. 7a–h, the attention at the 0-layer mainly focuses on residues near positions 38th, 84th, 170th, 194th, 219th, 251th, and 280th. After the 1-layer, the attention of the compound atoms gradually converged towards residues near positions 84th, 192th, 219th, 221th, and 251th.

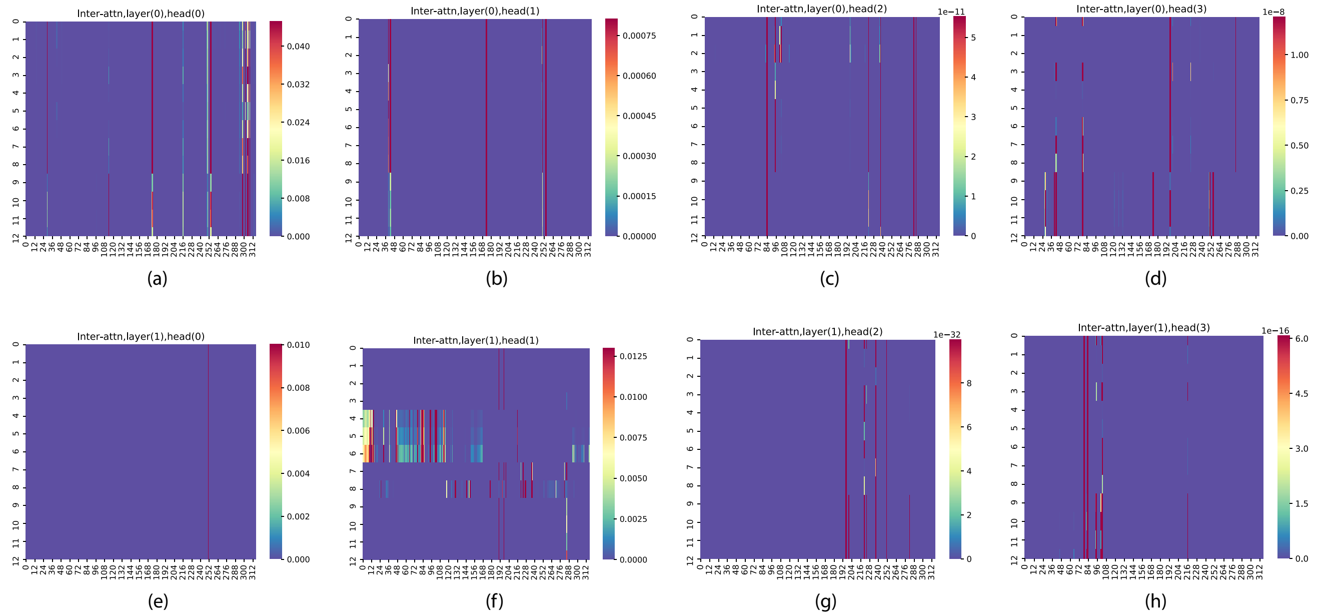


Fig. 7 Visualization of attention matrices of acetazolamide and mitochondrial carbonic anhydrase 5B in the 2-layer, 4-head GraphsformerCPI

To further explore the role of inter-attention in CPIs, we visualize the molecular docking of mitochondrial carbonic anhydrase 5B with acetazolamide using AutoDock [49], according to geometric and energetic complementarity. Molecular docking is a widely used computational chemistry method to predict and understand the binding mode and affinity between ligands and receptors. The process involves several steps: loading the 3D structure file of mitochondrial carbonic anhydrase 5B; performing operations such as dehydration, hydrogenation, and charge calculation for the receptor protein; loading the 3D structure file of acetazolamide; preoperating the ligand by hydrogenation, automatic charge distribution, detection of rotatable bonds, and selection of torsional bonds; defining the search space for the ligand in the protein pocket; placing the ligand at the active site of the target protein, and continuously optimizing the positions and conformations of the ligand molecules as well as the dihedral angles of the rotatable bonds while adjusting the side chains and backbone of the receptor residues to find the optimal ligand-receptor interaction conformation and calculate the key binding sites.

Figure 8 highlights the pocket residues in magenta and displays the compound atoms in iridescence. Protein residues located in or around the binding pocket easily interact with the compound via hydrogen bonds, van der Waals forces, or hydrophobic interactions. Specifically, in mitochondrial carbonic anhydrase 5B, pocket residues such as HIS-190, CYS-219, MET-221, THR-223, and PRO-251 form hydrogen bonds with atoms of acetazolamide. The attention distributions of compound atoms and protein residues are very close to the results of molecular docking.

As progressing from the 0-layer to the 1-layer, the extracted features become increasingly precise. Interestingly, Fig. 7f shows that the distribution of attention to protein residues for compound atoms located at positions 4th to 8th significantly deviates from that of other atoms. This may indicate that atoms in the heterocyclic structure of the compound pay more attention to protein residues.

By visualizing residues PRO-82, THR-84, and acetazolamide in Fig. 9, we investigate the high attention scores of acetazolamide near the 84th residue in Fig. 7h. The nearest atomic distances of the ligand to residues PRO-82 and THR-84 are 16.6Å and 14.3Å, respectively. This indicates that both residues are relatively close to the compound, leading the model to mistakenly assign more attention to them during the learning process.

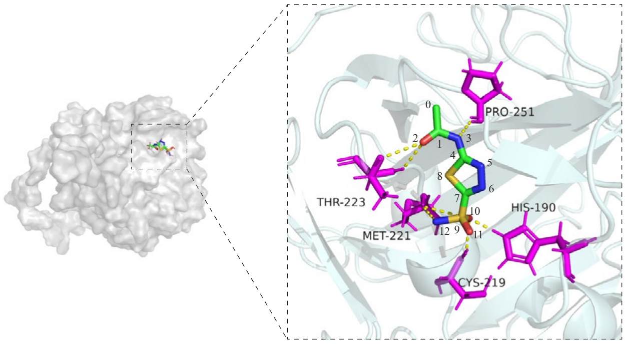


Fig. 8 Visualization of the binding sites of acetazolamide and mitochondrial carbonic anhydrase 5B with highlighted atoms and residues by autodock

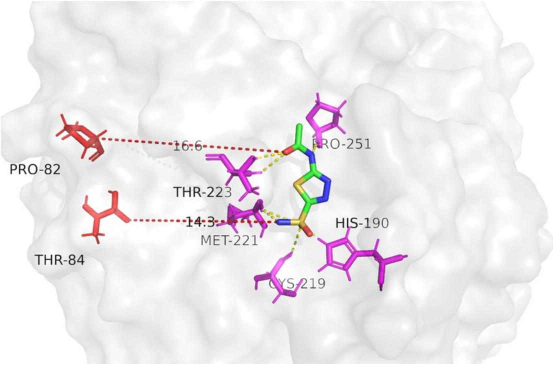


Fig. 9 Visualization of residues with high attention scores of mitochondrial carbonic anhydrase 5B

4 Conclusion

In this study, we proposed GraphsformerCPI, a novel deep learning framework that effectively predicts compound–protein interactions by fusing spatial structure and semantic information. By treating compounds and proteins as node sequences with spatial structure, GraphsformerCPI leverages a structure-enhanced self-attention mechanism to capture the impact of structural and semantic features on molecular representations. Additionally, a dual attention mechanism is employed to uncover hidden relationship features between compound atoms and protein residues. GraphsformerCPI incorporates AlphaFold2 to generate contact maps in protein feature extraction, which is rarely used in other studies. The integration of attention mechanisms in GraphsformerCPI not only enhances its predictive performance but also provides a level of interpretability that is often lacking in other blackbox deep learning models. We conducted extensive evaluations of GraphsformerCPI using diverse datasets, including human, *C. elegans*, Davis, and KIBA datasets and explored the impact of model depth and dropout rate on performance. Through comparative experiments, GraphsformerCPI exhibited higher performance to the state-of-the-art baseline models in AUC, precision, and recall on classification datasets, and achieved competitive performance in CI and MSE on regression datasets. Furthermore, the interpretable results of the proposed model were compared with molecular docking experiments, which provide a novel view for explaining the compound–protein interactions and binding mechanisms, differentiating it from other black-box deep learning models. Based on these evaluations, we can conclude that GraphsformerCPI effectively predicts CPIs and binding affinities, which has practical implications for identifying key atoms and residues, discovering drug candidates, and even downstream pharmaceutical tasks.

Acknowledgements This work is supported by Natural Science Foundation of Gansu Province, China (Project No. 21JR7RA460), the Fundamental Research Funds for the Central Universities (Project No. lzujbky-2022-39) and National Natural Science Foundation of China (Project No. 61702240).

Data availability The datasets and source code of our method can be downloaded at https://github.com/happay-ending/graphsformerCPI.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Sangsoo L, Yijingxiu L, Chang Yun C et al (2021) A review on compound–protein interaction prediction methods: Data, format, representation and model. Comput Struct Biotechnol J 19:1541–1556. https://doi.org/10.1016/j.csbj.2021.03.004
2. Masoudi-Sobhanzadeh Y, Omidi Y, Amanlou M et al (2020) Drug databases and their contributions to drug repurposing. Genomics 112:1087–1095. https://doi.org/10.1016/j.ygeno.2019.06.021
3. Abbasi K, Razzaghi P, Poso A et al (2021) Deep learning in drug target interaction prediction: current and future perspectives. Curr Med Chem 28:2100–2113. https://doi.org/10.2174/0929867327666200907141016
4. D'Souza S, Prema KV, Balaji S (2020) Machine learning models for drug-target interactions: current knowledge and future directions. Drug Discov Today 25:748–756. https://doi.org/10.1016/j.drudis.2020.03.003
5. Schneider G (2010) Virtual screening: an endless staircase? Nat Rev Drug Discov 9:273–276. https://doi.org/10.1038/nrd3139
6. Du B, Qin Y, Jiang Y et al (2022) Compound-protein interaction prediction by deep learning: databases, descriptors and models. Drug Discov Today 27:1350–1366. https://doi.org/10.1016/j.drudis.2022.02.023
7. Macarron R, Banks MN, Bojanic D et al (2011) Impact of high-throughput screening in biomedical research. Nat Rev Drug Discov 10:188–195. https://doi.org/10.1038/nrd3368
8. Sadybekov AA, Sadybekov AV, Liu Y et al (2022) Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. Nature 601:452–459. https://doi.org/10.1038/s41586-021-04220-9
9. Deane C, Mokaya M (2022) A virtual drug-screening approach to conquer huge chemical libraries. Nature 601:322–323. https://doi.org/10.1038/d41586-021-03682-1
10. Huang K, Fu T, Gao W et al (2021) Therapeutics data commons: machine learning datasets and tasks for drug discovery and development. Proc Neural Inf Process Syst NeurIPS Datasets Benchmarks. https://doi.org/10.48550/arXiv.2102.09548
11. Lavecchia A (2019) Deep learning in drug discovery: opportunities, challenges and future prospects. Drug Discov Today 24:2017–2032. https://doi.org/10.1016/j.drudis.2019.07.006
12. Voulodimos A, Doulamis N, Doulamis A et al (2018) Deep learning for computer vision: a brief review. Comput Intell Neurosci. https://doi.org/10.1155/2018/7068349
13. Li J (2022) Recent advances in end-to-end automatic speech recognition. APSIPA Trans Signal Inf Process. https://doi.org/10.1561/116.00000050
14. Chen M, Firat O, Bapna A et al (2018) The best of both worlds: Combining recent advances in neural machine translation. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol 1. pp 76–86. https://aclanthology.org/P18-1008
15. Wu S, Sun F, Zhang W et al (2022) Graph neural networks in recommender systems: a survey. ACM Comput Surv 55:1–37. https://doi.org/10.1145/3535101
16. Li J, Zheng S, Chen B et al (2016) A survey of current trends in computational drug repositioning. Brief Bioinform 17:2–12. https://doi.org/10.1093/bib/bbv020
17. Weininger D (1988) Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28:31–36. https://doi.org/10.1021/ci00057a005
18. Pearson WR, Lipman DJ (1988) Improved tools for biological sequence comparison. Proc Natl Acad Sci 85:2444–2448. https://doi.org/10.1073/pnas.85.8.2444
19. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–44. https://doi.org/10.1038/nature14539
20. Sherstinsky A (2020) Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D Nonlinear Phenom 404:132306. https://doi.org/10.1016/j.physd.2019.132306
21. Kiranyaz S, Avci O, Abdeljaber O et al (2021) 1D convolutional neural networks and applications: a survey. Mech Syst Signal Process 151:107398. https://doi.org/10.1016/j.ymssp.2020.107398
22. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. In: NIPS'17 6000-6010. https://doi.org/10.48550/arXiv.1706.03762
23. Öztürk H, Özgür A, Ozkirimli E (2018) DeepDTA: deep drug-target binding affinity prediction. Bioinformatics 34:i821–i829. https://doi.org/10.1093/bioinformatics/bty593
24. Wan F, Zhu Y, Hu H et al (2019) DeepCPI: a deep learning-based framework for large-scale in silico drug screening. Genom Proteom Bioinform 17:478–495. https://doi.org/10.1016/j.gpb.2019.04.003
25. Karimi M, Wu D, Wang Z et al (2019) DeepAffinity: interpretable deep learning of compound-protein affinity through unified recurrent and convolutional neural networks. Bioinformatics 35:3329–3338. https://doi.org/10.1093/bioinformatics/btz111
26. Chen L, Tan X, Wang D et al (2020) TransformerCPI: improving compound-protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. Bioinformatics 36:4406–4414. https://doi.org/10.1093/bioinformatics/btaa524

27. Wang Y, You Z, Yang S et al (2020) A deep learning-based method for drug-target interaction prediction based on long short-term memory neural network. BMC Med Inf Decis Mak 20:49. https://doi.org/10.1186/s12911-020-1052-0
28. Jin W, Barzilay R , Jaakkola T (2018) Junction tree variational autoencoder for molecular graph generation. In: Proceedings of the 35th international conference on machine learning, vol 80. pp 2323–2332. https://doi.org/10.48550/arXiv.1802.04364
29. Nguyen T, Le H, Quinn TP et al (2020) GraphDTA: predicting drug-target binding affinity with graph neural networks. Bioinformatics 37:1140–1147. https://doi.org/10.1093/bioinformatics/btaa921
30. Wang E, Wang F, Yang Z et al (2020) A graph convolutional network-based method for chemical-protein interaction extraction: algorithm development. JMIR Med Inform 8:e17643. https://doi.org/10.2196/17643
31. Tsubaki M, Tomii K, Sese J (2018) Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics 35:309–318. https://doi.org/10.1093/bioinformatics/bty535
32. Torng W, Altman RB (2019) Graph convolutional neural networks for predicting drug-target interactions. J Chem Inf Model 59:4131–4149. https://doi.org/10.1021/acs.jcim.9b00628
33. Jiang M, Li Z, Zhang S et al (2020) Drug-target affinity prediction using graph neural network and contact maps. RSC Adv 10:20701–20712. https://doi.org/10.1039/D0RA02297G
34. Landrum G (2013) RDKit: open-source cheminformatics. Release 1:4. https://doi.org/10.5281/zenodo.591637
35. Jumper J, Evans R, Pritzel A et al (2021) Highly accurate protein structure prediction with alphafold. Nature 596:583–589. https://doi.org/10.1038/s41586-021-03819-2
36. Li Y, Hsieh C, Lu R et al (2022) An adaptive graph learning method for automated molecular interactions and properties predictions. Nat Mach Intell 4:645–651. https://doi.org/10.1038/s42256-022-00501-8
37. Davis MI, Hunt JP, Herrgard S et al (2011) Comprehensive analysis of kinase inhibitor selectivity. Nat Biotechnol 29:1046–1051. https://doi.org/10.1038/nbt.1990
38. Tang J, Szwajda A, Shakyawar S et al (2014) Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. J Chem Inf Model 54:735–743. https://doi.org/10.1021/ci400709d
39. Huang L, Lin J, Liu R et al (2022) CoaDTI: multi-modal co-attention based framework for drug-target interaction annotation. Brief Bioinform 23:bbac446. https://doi.org/10.1093/bib/bbac446
40. Liu H, Sun J, Guan J et al (2015) Improving compound-protein interaction prediction by building up highly credible negative samples. Bioinformatics 31:i221–i229. https://doi.org/10.1093/bioinformatics/btv256
41. Wishart DS, Knox C, Guo AC et al (2008) DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic Acids Res 36:D901–D906. https://doi.org/10.1093/nar/gkm958
42. Günther S, Kuhn M, Dunkel M et al (2007) SuperTarget and Matador: resources for exploring drug-target relationships. Nucleic Acids Res 36:D919–D922. https://doi.org/10.1093/nar/gkm862
43. Kuhn M, Szklarczyk D, Pletscher-Frankild S et al (2013) STITCH 4: integration of protein–chemical interactions with user data. Nucleic Acids Res 42:D401–D407. https://doi.org/10.1093/nar/gkt1207
44. He T, Heidemeyer M, Ban F et al (2017) SimBoost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines. J Cheminform 9:1–14. https://doi.org/10.1186/s13321-017-0209-z
45. Gönen M, Heller G (2005) Concordance probability and discriminatory power in proportional hazards regression. Biometrika 92:965–970. https://doi.org/10.1093/biomet/92.4.965
46. Wikipedia (2023) Pearson correlation coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
47. Nascimento AC, Prudêncio RB, Costa IG (2016) A multiple kernel learning algorithm for drug-target interaction prediction. BMC Bioinform 17:46. https://doi.org/10.1186/s12859-016-0890-3
48. Öztürk H, Ozkirimli E , Özgür A (2019) WideDTA: prediction of drug-target binding affinity. https://doi.org/10.48550/arXiv.1902.04166
49. Morris GM, Huey R, Lindstrom W et al (2009) AutoDock4 and AutoDockTools4: automated docking with selective receptor flexibility. J Comput Chem 30:2785–91. https://doi.org/10.1002/jcc.21256