DSAGLSTM-DTA: PREDICTION OF DRUG-TARGET AFFINITY USING DUAL SELF-ATTENTION AND LSTM

Lyu Zhijian, Jiang Shaohua and Tan Yonghao

College of Information Science and Engineering,
Hunan Normal University, Changsha, China

ABSTRACT

The research on affinity between drugs and targets (DTA) aims to effectively narrow the target search space for drug repurposing. Therefore, reasonable prediction of drug and target affinities can minimize the waste of resources such as human and material resources. In this work, a novel graph-based model called DSAGLSTM-DTA was proposed for DTA prediction. The proposed model is unlike previous graph-based drug-target affinity model, which incorporated self-attention mechanisms in the feature extraction process of drug molecular graphs to fully extract its effective feature representations. The features of each atom in the 2D molecular graph were weighted based on attention score before being aggregated as molecule representation and two distinct pooling architectures, namely centralized and distributed architectures were implemented and compared on benchmark datasets. In addition, in the course of processing protein sequences, inspired by the approach of protein feature extraction in GDGRU-DTA, we continue to interpret protein sequences as time series and extract their features using Bidirectional Long Short-Term Memory (BiLSTM) networks, since the context-dependence of long amino acid sequences. Similarly, DSAGLSTM-DTA also utilized a self-attention mechanism in the process of protein feature extraction to obtain comprehensive representations of proteins, in which the final hidden states for element in the batch were weighted with the each unit output of LSTM, and the results were represented as the final feature of proteins. Eventually, representations of drug and protein were concatenated and fed into prediction block for final prediction. The proposed model was evaluated on different regression datasets and binary classification datasets, and the results demonstrated that DSAGLSTM-DTA was superior to some state-of-the-art DTA models and exhibited good generalization ability.

KEYWORDS

Drug-Target Affinity, BiLSTM, pooling, Graph Neural Network, Self-Attention.

1. INTRODUCTION

According to incomplete statistics, the development of a new drug that can obtain marketing authorization is expected to cost hundreds of millions of dollars, and the rate of drug approval for clinical trials is only about 10%. Furthermore, due to the bottleneck of technological development, the development of new drugs is more difficult. Driven by these factors, researchers have to explore novel and more efficient approaches in drug discovery. Under this circumstances, the exploration of new uses of developed drugs has become a new hot spot. Discovering new associations between drugs and targets is critical for drug development and repurposing, However, the traditional study of drug-protein relationships in the wet laboratory [1][2] is time-consuming and expensive due to the huge range of chemical spaces to be searched, to solve this problem, some virtual screening(VS) has been proposed to accelerate the experimental drug discovery and reposition studies in silico [3], some of the more commonly

used VS methods, like structure-based VS, ligand-based VS and sequence-based VS have contributed to drug development to a large extent [4][5]. However, these VS methods have their own defects in application. For example, if the structural information of the protein is unknown, the structure-based approach cannot play its role. There is still a long way to go before accurately constructing the structure of proteins, to this end, some structure-free methods have sprung up.

In recent years, with the development and maturity of deep learning technology and its great breakthrough in the field of computer vision(CV) and natural language processing(NLP) [6][7], many people in the field of drug research have begun to turn their attention to deep learning. Methods based on deep learning to study drug-target relationships are usually computer-aided methods, and these methods can effectively speed up the virtual screening process of potential drug molecules because they can minimize unnecessary biological and chemical experiments by adjusting the search space. Moreover, with the advent of more and more biological activity data, a great deal of computer-aided work based on these data has been carried out to investigate the relationship between drugs and targets. These work is usually divided into two categories, one is binary classification-based approach, that is, to determine whether drugs and targets interact through data labels(i.e., active or inactive), and the other is a regression-based approach, which uses binding tightness(specific values) to describe the relationship between the drug and the target.

In binary classification-based drug-target (DT) prediction tasks, deep learning technologies seem to be used by more researches to deal with drug-target interaction (DTI) problems. When doing DTI prediction tasks in the past, compounds and proteins were represented using manually crafted descriptors and the final interaction prediction was made through several fully connected networks [8][9]. The problem with this approach is that the descriptors are designed from a specific perspective, that is, the design angle is too single, in addition, it remains fixed during the training process, so it cannot learn and adjust according to the results, and thus cannot extract task-related features. Therefore, some end-to-end models were proposed. Du *et al*. proposed a model called wide-and-deep to predict DTIs [10] where a generalized linear model and a deep feed-forward neural network were integrated to enhance the precise of DTI prediction. Molecular structural information is also of great significance for feature extraction. To learn the mutual interaction features of atoms in a sequence, Shin *et al*. proposed a Transformer-based DTI model [11], which used multilayer bidirectional Transformer encoders [12] to learn the high-dimensional structure of molecules from the Simplified Molecular Input Line Entry System (SMILES) strings. Some researchers obtained structural information of compounds or proteins from another perspective, they represented the corresponding compounds or proteins as graphs and utilized graph neural networks to extract their spatial features, related work such as GraphCPI [13], Graph-CNN [14], etc.

Compared with the binary classification model, it seems more convincing to describe the relationship between drug and target through a regression task, the use of regression model can provide us with more information about the relationship between compounds and proteins, since continuous values can tell us how strongly the two are bound. In the early stages of this field, drugs and proteins were represented by researchers using human experience or skillfully designed mathematical descriptors. Related studies include KronRLS [15] and SimBoost [16], both of which based on regression and utilized the similarity information of drugs and targets to predict DTAs. Although these approaches achieve good results in DTA tasks, they rely on chemical insights or expert experience, which in turn limits further optimization of these models. What's more, the rapid advancement in deep learning has also largely facilitated the affinity prediction of DT pairs and various data-driven models are applied to the description of drugs and targets. DeepDTA [17] is the first framework for predicting drug and target affinity based on deep learning, which utilized two CNN blocks to process SMILES strings of drugs and amino acid

sequences of proteins, respectively. Works related to DeepDTA include WideDTA [18]. The improvement of WideDTA over DeepDTA is that it combined several characters as words and proposed a word-based sequence representation method. In order to better capture the topological structure features of compounds, Nguyen et al. proposed GraphDTA [19] to predict drug and target affinity which utilized RDKit technology to represent drug strings as graphs that could reflect its structural characteristics, and used graph convolutional neural network to extract its spatial features. Furthermore, Lin proposed a similar approach called DeepGS [20], which used advanced techniques to encode amino acid sequences and SMILES strings. DeepGS also combined a GAT model to capture the topological information of molecular graph and a BiGRU model to obtain the local chemical context of drug.

In the above two categories of research tasks, a new mechanism called attention is increasingly gaining the favor of researchers, since the performance of either the binary classification-basedor regression-based approach can be enhanced by introducing attention mechanisms. Examples of two typical applications are AttentionDTA [21] and HyperAttentionDTI [22], the novelty of HyperAttentionDTI and AttentionDTA lies in that they utilized an attention mechanism for learning important parts of each other's sequences. Zhang et al. proposed a different attention mechanism in SAG-DTA [23], which is based on graph structure rather than sequence. In SAG-DTA, a self-attention pooling network was used to learn the structural features of drug molecular graphs, in which the features of each atom node in the molecular graph were weighted using an attention score before being aggregated as molecule representation. In this work, inspired by the above attention mechanism, we proposed a novel framework based on self-attention to predict DTAs. In the process of extracting molecular graph features of drugs, the same as SAG-DTA, we utilized a novel attention structure that introduces self-attention mechanisms for node pooling named self-attention graph pooling (SAGPool) [24], in which the self-attention graph pooling approach is adopted to molecular graph representation. Moreover, two different self-attention network architectures called centralized pooling and distributed pooling were constructed and compared. For protein sequence feature extraction, due to the limitation of the convolution kernel size, the field of vision of the convolutional neural network (CNN) is limited, so it is difficult to effectively capture the long amino acid sequence features that are context-dependent, so we usedbidirectional Long Short-Term Memory(BiLSTM) networks to extract protein features. On this basis, we applied another self-attention mechanism to protein sequences, we combined the final hidden states of each element in the batch with each unit output of the LSTM as the final features of proteins to obtain comprehensive representations. Experimental results demonstrated that our model greatly improves the performance compared to previous models.

2. MATERIALS AND METHODS

DSAGLSTM-DTA is a computation-based end-to-end deep learning algorithm that takes the features of drugs and targets as inputs and affinity/interaction values between them as output. DSAGLSTM-DTA contains two prediction tasks, DTA and DTI, in which the context data of drug-protein pairs are input into the network for feature extraction, which is then used by the algorithm of the model to evaluate the inner association, and the predicted value is obtained. In this work, a more complicated graph neural network was implemented by using the self-attention pooling mechanism. Specifically, the features of the nodes were learned by attention scores to weight the atom nodes. Furthermore, attention scores were also utilized to sort and filter atom nodes. For the protein feature extraction, we hypothesized that each hidden layer cell of LSTM actually contains some extra features of protein, so it should not be ignored, therefore, we performed weighted attention calculation between the output of LSTM and its hidden layer cell to effectively capture its comprehensive representation. We believed that the above strategy will enable the network to pay more attention to the most important parts and thus obtain a more pure

feature representation for the prediction task. The overall flow of DSAGLSTM-DTA is expressed in Figure 1. As can be seen in Figure 1, the SMILES strings of drugs are preprocessed into molecular graphs, which are then fed into graph neural network with self-attention pooling for feature extraction. For the protein, the amino acid sequences of proteins are fed into the bidirectional LSTM network for contextual feature extraction. Subsequently, the output of the LSTM is weighted with the state of final hidden layer, and the result of the calculation is functioned as the final feature representation.

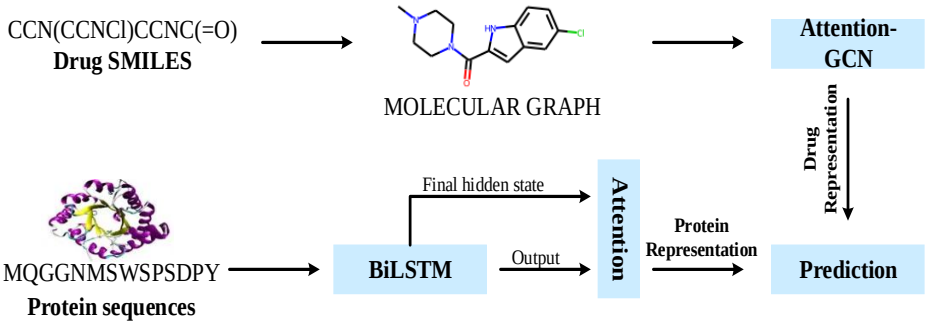

Fig. 1. Overall flow of DSAGLSTM-DTA.

2.1. Datasets

In our experimental evaluation, we used the two datasets most commonly used in DTAs prediction, namely Davis [25] and KIBA [26]. The Davis stores selectivity as say data for the kinase protein families and the relevant inhibitors, along with their respective disassociation constant (K_d) values, moreover, it contains 72 compounds and 442 proteins, and their corresponding affinity values, where the affinity values are measured by K_d values (kinase dissociation constant). There are a total of 30056 affinity values in Davis, and they range from 5.0 to 10.8. We converted K_d into the value of the corresponding logarithmic space, pK_d, as follows:

$$\mathrm{P}K_{\mathrm{d}} = -\log_{10}\left(\frac{K_{\mathrm{d}}}{10^{9}}\right) \tag{1}$$

The KIBA dataset contains 2116 compounds and 229 proteins, as well as 118,254 drug and target affinity values, where the affinity values range from 0.0 to 17.2. And the affinity value in KIBA is represented by KIBA score which is computed by combining heterogeneous information sources, i.e., IC_{50}, K_{i} and K_{d}. The data in this dataset have high quality, since the integrated heterogeneous metrics alleviate the coupling associated with using a single source of information.

In addition to using a regression task for drug-target relationship prediction, the proposed model was also applied to a binary classification task for evaluation, and two binary benchmark datasets called human [27] and BindingDB [28] for CPI prediction were used for the experiments. The positive CPI pairs in the Human dataset are derived from DrugBank [29] and Matador [30], and the negative CPI samples in this dataset is highly credible. BindingDB is another commonly used CPI prediction dataset, which is characterized by containing pre-processed training, validation, and test sets. The overview information of the four benchmark datasets is summarized in Table 1.

Table 1. Summary of the benchmark datasets

Datasets	Compound	Protein	Binding Entities	Task Type	Ref
Davis	72	442	30056	DTA(regression)	[25]
KIBA	2116	229	118254	DTA(regression)	[26]
Human	1052	852	3369(+)/3359(-)	CPI(binary-class)	[27]
BindingDB	53253	1696	39747(+)/31218(-)	CPI(binary-class)	[28]

2.2. Overview of the Network Architecture

In this section, we will introduce overview network architectures of our model. As mentioned earlier, DSAGLSTM-DTA contains two different network architecture, namely centralized and distributed architectures, the difference between these two architectures lies in the different positions of self-attention pooling in drug feature extraction. The centralized pooling architecture is presented in the left panel of Figure 2, consists of three graph neural network layers, and the outputs of each layer are concatenated and fed into the self-attention layer. These reserved nodes are then fed into a global_max_pool layer and several fully connected layers for final representations. The distributed pooling architecture is illustrated in the right panel of Figure 2, the difference between this architecture and the previous is that a single self-attention pooling is used after each graph convolutional network layer. Subsequently, the outputs of each layer after self-attention pooling are concatenated, that is, pooling in a distributed way. The above two architectures are consistent in the structure of the protein network. The features of proteins are differentiated into two parts (i.e. output and the hidden state) after passing through the bidirectional LSTM network block, and these two branches are then input into the attention block for weighting calculation.

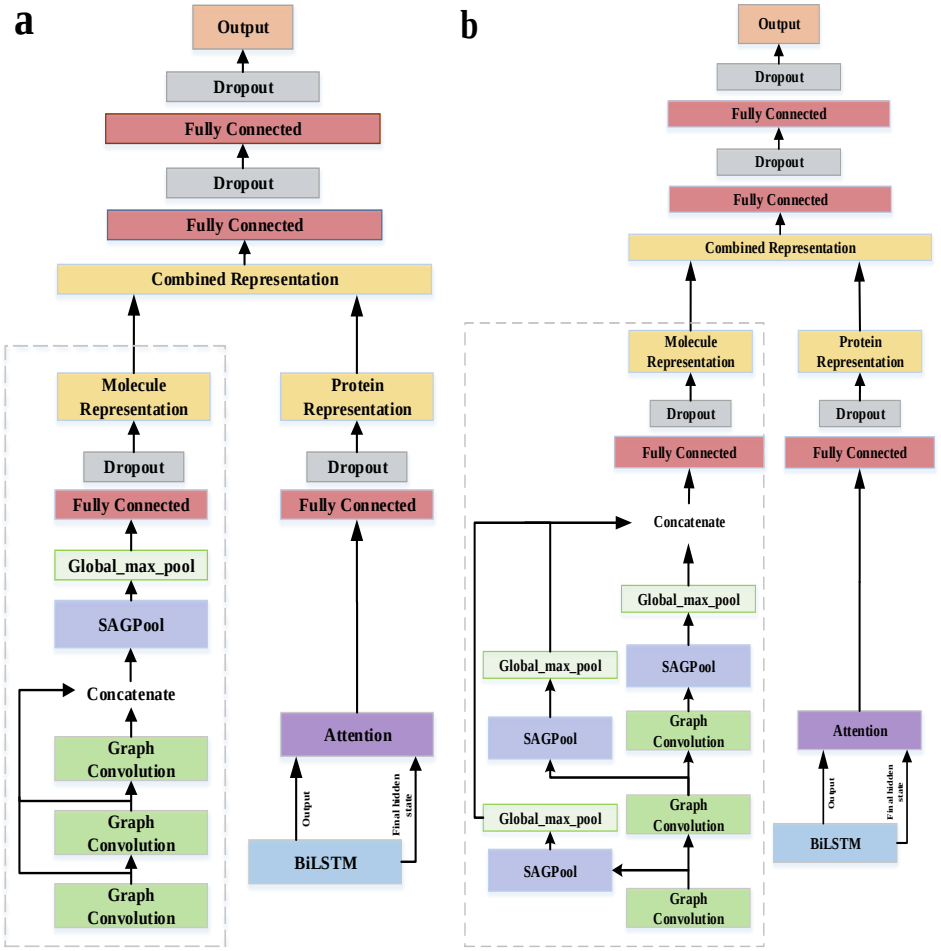

Figure 2. Network architectures of DSAGLSTM-DTA. Substructures surrounded by dashed lines indicate molecular graph representation, which is the major difference of the two architectures. (a) Centralized pooling architecture. (b) Distributed pooling architecture.

2.2.1. Data Preprocessing

The feature extraction of drugs and targets are two independent input channels. Before drugs and targets are input into their respective feature extraction blocks, data preprocessing is required for drugs and targets, respectively. The implementation details are as follows.

2.2.1.1. Drug Representation

For data preprocessing of drugs, we used the same method as GraphDTA, we utilized the open source technology RDKit to convert the SMILES strings of drugs into corresponding 2D molecule graphs. The molecule graph was denoted as $G = (V, E)$, and the vertexes V were represented as atoms and the edges E were represented as bonds, where $|V| = N$ is the number of nodes in the graph and $|E| = N^e$is the number of edges. Each atom was embedded with 78-dimensional features such as the atom's type, degree, implied valence, aromaticity, and the number of hydrogen atoms attached to the atom. The feature of the node was encoded as a one-hot vector of shape $(N, 78)$. The chemical bonds index was encoded as $(2, E)$ vector, which is used to store the edges of the undirected graph. The schematic diagram of the SMILES string of a drug converted into a two-dimensional molecule graph by rdkit technology is as follows:

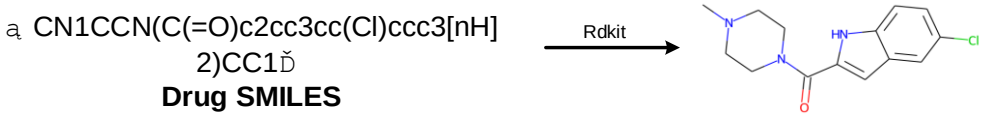

Fig. 3. Convert SMILES string to graph.

2.2.1.2. Target Representation

The sequence length of each protein is different and varies greatly. For uniform feature representation, we fixed the length of all protein sequences as 1000 according to the average length of protein sequences, if the sequence length of the protein exceeds 1000, the part more than 1000 will be cut off, and otherwise, the part less than 1000 will be padded with 0. In addition, protein sequences were represented by different combinations of 25 amino acids, and each represented by the one-letter code. We then mapped each amino acid to an integer, and each integer was embedded as a 128-dimensional feature.

2.2.2. Graph Convolution Layer

2.2.2.1. GatedGraph

In the graph neural network block, GatedGraph [31] convolution algorithm was utilized by us to extract the 2D molecular graph features of drugs and its details are as follows. GatedGraph is a feature learning technique that studies graph-structured inputs, it modifies previous graph neural network work using gated recurrent units (GRU) and modern optimization techniques, and then extends to output sequences, so this method can make full use of long-distance information and fit well with our model of extracting protein features. In addition, GatedGraph has favorable inductive biases relative to purely sequence-based models when dealing with graph structure problems, and thus is a flexible and widely useful class of neural network models. The features of the node are updated as follows:

$$\mathrm{h}_{\mathrm{i}}^{(0)} = \mathrm{x}_{\mathrm{i}} \,||\, 0 \tag{2}$$

$$\mathrm{m}_{i}^{(l+1)} = \sum_{j \in N(i)} \mathrm{e}_{j,i} \cdot \Theta \cdot \mathrm{h}_{j}^{(l)} \tag{3}$$

$$h_{i}^{(l+1)} = \mathrm{GRU}\left(m_{i}^{(l+1)}, h_{i}^{(l)}\right) \tag{4}$$

Where in formula (2), $\mathrm{h}_{\mathrm{i}}^{(0)}$ is the input state, $\mathrm{x}_i \in \mathrm{R}^F$ is the feature of node i, $\mathrm{x}_i \,||\, 0$ represents padding 0 after feature x_i to the specified dimension. In formula (3), Θ is the parameter matrix to be learned, that is, the aggregation information of surrounding nodes. Formula (4) is to use a GRU unit to take the above two formulas as input and get an output, which can be functioned as a new feature of node i.

2.2.3. LSTM Block

2.2.3.1. LSTM

When CNN is used to extract the context dependences of long sequences, the field of view is limited due to the influence of the size of convolution kernel, and multiple CNN layers need to be

used, which makes the model bloated and complex. In order to overcome the inability of CNN and RNN to deal with long-distance dependence, LSTM (Long-Short Term Memory) [32] is proposed. LSTM captures long-term dependencies by controlling the circulation and loss of features using a "gating" mechanism. As Figure 4 illustrated, LSTM unit is composed of a cell, a forget gate, an input gate, and an output gate. These gates are responsible for controlling the interactions among different memory cell.

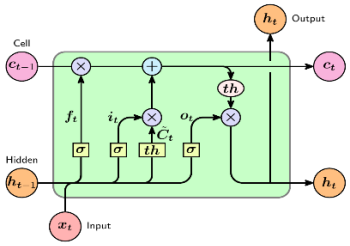

Fig. 4. Overview of a LSTM unit.

Some important elements contacted to LSTM can be expressed as follows:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{5}$$

The forget gate f_t is used to evaluate which features of previous cell state should be retained for calculation. Where σ is the sigmoid function, through which the data can be transformed into a value in the range of 0~1 to act as a gating signal. x_t is the input of the current node, h_{t-1} is the hidden state passed down by the previous node, and this hidden state contains the relevant information of the previous node. U_f and W_f are the corresponding weight matrices, respectively.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{6}$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{8}$$

The input gate i_t is responsible for updating current information in real time. $\tilde{C}_t$ represents the update value of the cell state at the current moment, which is obtained from the input data and the hidden state through a neural network layer, and the activation function usually uses tanh. $\otimes$ is the most important gate mechanism of LSTM, representing the unit multiplication relationship between two data.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{9}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{10}$$

The output gate o_t controls the output of the cell state into the rest of the network. The hidden state h_t is obtained by the output gate o_t and the cell state C_t. The calculation method of o_t is the same as that of f_t and i_t. It is worth noting that in Equation (9), the LSTM can be made to approximate its variant GRU by initializing the mean of b_o to 1 [33].

The feature extraction process of LSTM can be shown in Fig. 5.

2.2.3.2. BiLSTM

For some specific tasks, the information at a certain moment is not only related to the previous state, but also has some connection with the later state. When dealing with such problems, the traditional unidirectional LSTM is obviously not competent, therefore, the bidirectional LSTM is introduced. For protein sequences, we considered that the features of a certain part of the protein sequence are not only related to the previous part, but also related to the later part. Therefore, in our work, we used bidirectional LSTM networks to extract the amino acid sequence features of the proteins.

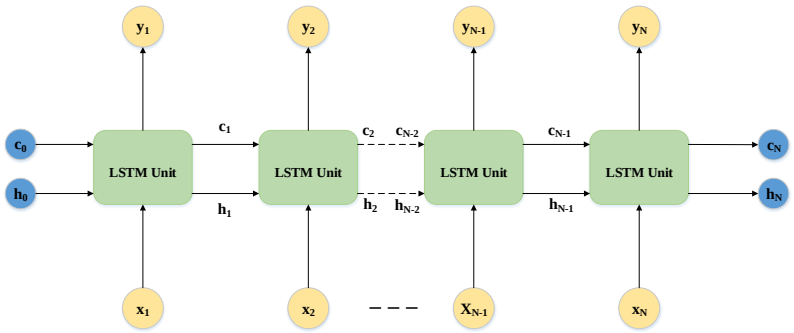

Fig. 5, the output of each stage is jointly determined by the hidden and cell state of its previous stage and the input of the current stage.

BiLSTM is composed of two unidirectional LSTMs with opposite directions. At each moment, the input will fuse the outputs of the two opposite LSTMs at the same time, and the output is jointly determined by these two unidirectional LSTMs. The feature extraction process of BiLSTM is as follows:

$$\overrightarrow{h_t} = \mathrm{LSTM}\left(x_t, \overrightarrow{h_{t-1}}, \overrightarrow{C_{t-1}}\right) \tag{9}$$

$$\overleftarrow{h_t} = \mathrm{LSTM}\left(x_t, \overleftarrow{h_{t-1}}, \overleftarrow{C_{t-1}}\right) \tag{10}$$

$$h_t = w_t\overrightarrow{h_t} + v_t\overleftarrow{h_t} + b_t \tag{11}$$

Where the function LSTM () represents a series of LSTM operations of the network layer. $\overrightarrow{C_{t-1}}$ and $\overleftarrow{C_{t-1}}$represent the cell state of the previous stage. $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represent the hidden layer state in the corresponding direction, respectively. w_t and v_t represent the weights corresponding to the forward hidden layer state $\overrightarrow{h_t}$ and reverse hidden layer state $\overleftarrow{h_t}$ of the bidirectional LSTM at time t, respectively. b_t represents the bias corresponding to the hidden layer state at time t. The feature extraction process of BiLSTM networks can be shown in Fig. 6.

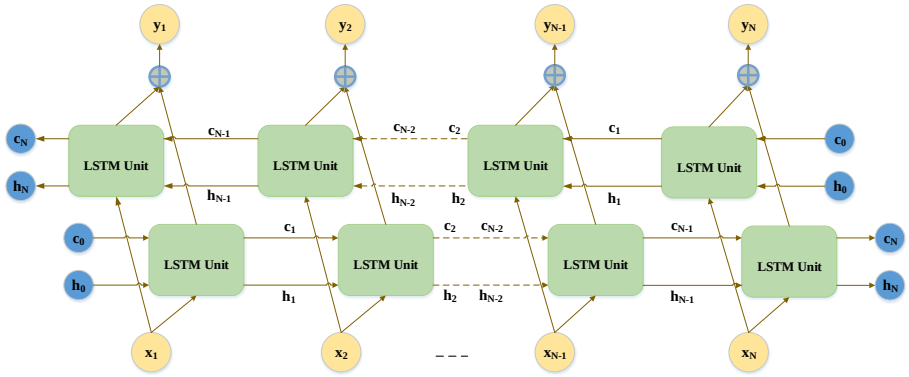

Fig. 6. the output of each stage is jointly determined by the hidden state and cell state of its previous and subsequent stages and the input of the current stage.

2.2.4. Self-Attention Graph Pooling Block

The self-attention graph pooling (SAGPool) was first proposed by Lee et al[24]. It was subsequently applied by SAG-DTA to molecular graph feature extraction for drug and target prediction. The innovations of SAGPool are: 1.The SAGPool method can learn a hierarchical representation in an end-to-end manner with relatively few parameters; 2.Self-attention mechanisms are utilized to distinguish the nodes that should be deleted from those that should be kept; 3.Self-attention mechanisms based on graph convolution to calculate attention scores, and node features and graph topological structure are taken into account. In general, in SAGPool, a total of four graph convolution methods are used to obtain the self-attention score of each node in the molecular graph, then these nodes are ranked according to the corresponding attention score and a certain ratio is used to determine the preserved atoms. Finally, the pooled molecular graph is obtained through the mask operation. The process of self-attention pooling is illustrated in Figure 7.

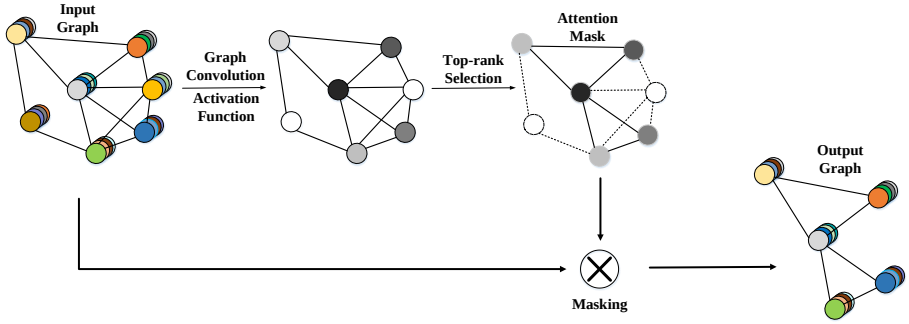

Fig. 7. the self-attention scores of the input graph are obtained using the graph convolution method, the nodes corresponding to the self-attention scores are ranked, and finally the mask operation is applied to obtain the output graph.

Since there are various graph convolution methods for obtaining node self-attention scores in SAGPool, according to SAG-DTA, we used GraphConv (Graph convoluton) as our final scoring method. The GraphConv scoring method is presented as Equation (12).

$$Z = \sigma(h_v\Theta_1 + \sum_{u\in N(v)} h_u\Theta_2) \tag{12}$$

Where h represents the feature of the corresponding node, $N(v)$ is the set of all nodes adjacent to node v, Θ_1 and Θ_2 correspond to trainable convolutional weights with specific feature dimension. σ represents the activation function ReLU.

The mask operation is based on the self-attention scores of the atoms computed by the scoring method to determine which parts in the input graph should be reserved. The mask operation is illustrated in Equation (13).

$$\text{idx} = \text{top} - \text{rank}(\text{Z}, \lceil kN \rceil), \quad Z_{mask} = Z_{idx} \tag{13}$$

Where the $k \in (0, 1]$ is the pooling ratio to decide which portion of nodes should be retained. Z represents the self-attention scores of nodes. $\lceil kN \rceil$ is the number of reserved nodes, and the order of nodes is ranked according to self-attention score.

2.2.5. Attention Block

The BiLSTM layer generates new protein representation $P \in R^{M \times N \times F}$ and final hidden state $H \in R^{2 \times M \times f}$, where N represents data batch, M is the length of the protein sequence, f and F are the output sequence feature dimensions. Subsequently, the two matrices are multiplied after being transformed, and we obtain an attention matrix with weight called $\text{A} \in R^{N \times M}$. After that, the elements of the weight matrix are mapped to the interval (0, 1) by utilizing the softmax activation function. Finally, the feature matrix of the protein $P' \in R^{N \times M \times F}$ is weighted by the activated weight matrix A' to obtain the final representation of the protein $P_{final} \in R^{N \times F}$. The related operations can be formulated as Equation (14-17).

$$\text{A} = P' \times H' \tag{14}$$

$$\text{A}' = Softmax(A) \tag{15}$$

$$P_{final} = P' \times A' \tag{16}$$

$$Protein = F(W_p P_{final} + b_p) \tag{17}$$

Where P' and H' represent the matrix of protein features and hidden states after a series of matrix transformations. The series of operations include dimension increase and reduction, matrix transposition, etc. Softmax is a nonlinear activation function, which is usually used for classification tasks. $\times$ stands for matrix multiplication. $F(\cdot)$ is a non-linear activation function (e.g., ReLU), $W_p \in R^{F' \times F}$ is the weight matrices, and b_p is the bias vector.

2.2.6. MLP Block

The features of the drug and protein are concatenated after being extracted and then fed into the Multilayer Perceptron block for final prediction. The Multilayer Perceptron block consists of two fully connected layers, each of which is followed by a Dropout of rate 0.5 to prevent over fitting. The activation function of fully connected layer is the Rectified Linear Unit (ReLU). The output of the last layer identifies the final predicted affinity value for the drug and protein.

2.3. Implementation

DSAGLSTM-DTA was implemented in Pytorch[34] and its extension library PyTorch Geometric (PyG)[35]. We used the Adam optimizer with the default learning rate of 2e-4 for regression

tasks and 1e-3 for binary classification tasks. In addition, the number of training epochs for regression task and binary classification task was set to 2000 and 1000, respectively. The batch size was set to 512 and dropout rate was set to 0.5. According to the experimental conclusion of SAG-DTA, the pooling ratio of SAGPool was set as 1.

The SMILES string for each drug was converted into 2-dimensional molecular graph where each node of the molecular graph was embedded with 78-dimensional features. For the centralized architecture, GNN block consists of three stacked GNN layers with 78, 156 and 312 output features, respectively. Features extracted by each layer of graph convolution were concatenated and filtered through a self-attention pooling layer, which then followed by a global max pooling layer to get the most striking features. The distributed architecture contains three same graph neural networks with the 78 output feature, which is same with the hierarchical architecture used by SAG-DTA and Lee et al. However, the output of each layer of graph convolution is directly concatenated after being filtered by its own self-attention pooling layer instead of continuing to the next layer, which is different from the hierarchical structure. The purpose of our design is to expect the input graph will not lose important information due to excessive screening. The features extracted by the above two structures get the final drug representation after passing through two fully connected layers with 1024 and 128 neurons respectively.

The protein input embedding is of size 128, which means that we represent each character in amino acid sequence with a 128-dimensional dense vector. The number of layers of BiLSTM was set to 1, and the hidden feature dimension of LSTM was 32. The output and hidden state of the LSTM were weighted and passed through a fully connected network with 128 neurons as the final representation of the protein.

The prediction block is made up of three fully connected layers, in which the numbers of neurons are 1024, 512 and 1, respectively. Each drug and protein are converted into a 128-dimensional vector after their respective feature extraction, and are concatenated into a 256-dimensional vector for the final prediction. The number of training epochs is set to 2000 for regression tasks and 1000 for binary classification tasks. Our experiments are run on Windows 10 professional with Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz and GeForce GTX 1660Ti(6GB).

3. EXPERIMENTS AND RESULTS

3.1. Evaluation Metrics

MSE (Mean Squared Error), CI (Concordance Index) and r_m^2 (Regression toward the mean) are the most commonly used evaluation metrics in regression tasks to study drug-target interactions [15-21]. Therefore, we continue to use these evaluation metrics, the details of each metric are as follows:

MSE is the mean square error, which is used to measure the gap between the predicted value of the model and the actual label value. The smaller the gap is, the better the performance of the model is; otherwise, the worse the performance of the model is.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(P_i - Y_i)^2 \tag{18}$$

Where P_i is the prediction value, Y_i corresponds to the label value and n is the total number of samples.

CI is the Concordance Index, which is a measure of whether the order of predicted binding affinity values for two random drug-target pairs is consistent with their true values, which value exceeds 0.8 indicates a strong model.

$$\mathrm{CI} = \frac{1}{Z} \sum_{y_i > y_j} h(p_i - p_j) \tag{19}$$

$$h(x) = \begin{cases} 1, x > 0 \\ 0.5, x = 0 \\ 0, x < 0 \end{cases} \tag{20}$$

In (19), sample *i* has a bigger label value than sample *j*.

R_m^2 index is the regression toward the mean, which is used to evaluate the external predictive performance. The metric can be described as follows:

$$r_m^2 = r^2 * \left(1 - \sqrt{r^2 - r_0^2}\right) \tag{21}$$

Where r^2 and r_0^2 are the squared correlation coefficients with and without intercept, respectively. A value of r_m^2 above 0.5 is considered an ideal model.

To demonstrate the generalization ability of our model, we also applied the proposed model to a binary classification problem, namely drug and target interaction prediction (DTI). Precision and recall are the most frequently used metrics to evaluate binary classification tasks, therefore, in this work, we continue to use them to evaluate our model. They can be formulated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$Recall = \frac{TP}{TP + FP} \tag{23}$$

Where *TP*, *FP*, and *FN* represent the sample numbers of true positive, false positive, and false negative, respectively. In addition to the above two evaluation metrics, other commonly used strategies like the area under the receiver operating characteristic curve (AUROC) and the area under the precision recall curve (AUPRC) are also used by us to measure the performance of the model.

3.2. Results

3.2.1. Comparison with other Regression Models

The DSAGLSTM-DTA model combines GNN and BiLSTM, and we conducted experiments on two different datasets, Davis and KIBA. We compared the model with traditional learning methods and some current deep learning methods, where for deep learning methods, we divided it into two types according to the dimensional representation of the drug. In above models, drugs and proteins are represented by different descriptors, and these descriptors are subsequently extracted by various feature extraction approaches. In this experiment, we divided the dataset into five equal parts, four of which were used as training set and one was used as test set, in order to

prevent data overfitting, we also divided the validation set from the training set and used cross-validation to train the data set.

The experimental results in Table 2 demonstrate that compared with other DTA methods, DSAGLSTM-DTA has a huge improvement in performance on Davis dataset. For each DTA model, we used its optimal data for comparison. In the analysis based on below table, the two different architecture approaches of the proposed model outperform the baseline model to varying degrees in three indicators, where the optimal data of our model is 0.01 lower than the optimal data of the baseline model in MSE, an increase of about 4.6%. In addition, CI and r_m^2 increased by 0.013 and 0.041 respectively. Although distributed architecture of DSAGLSTM-DTA is slightly inferior to centralized architecture in comprehensive performance, it still far exceeds other baseline models. It is also worth noting that the CI values of two methods of the proposed model all exceed 0.9, which proves that they have strong consistency, moreover, the r_m^2 values are all over 0.7, indicating that they have strong external prediction performance and is an acceptable model.

Table 2. Results of various DTA prediction models on the Davis dataset

Method	Protein	Compound	CI	MSE	r_m^2
Traditional methods					
KronRLS[15]	S-W	Pubchem	0.871	0.379	0.407
SimBoost[16]	S-W	Pubchem	0.872	0.282	0.644
1D Representation-based Approaches					
WideDTA[18]	CNN	CNN	0.886	0.262	—
DeepDTA[17]	CNN(PS+PDM)	CNN(LS+LMCS)	0.878	0.261	0.630
AttentionDTA[21]	CNN	CNN	*0.893*	*0.216*	0.677
2D Representation-based Approaches					
DeepGS[20]	GAT+Smi2Vec	CNN(Prot2Vec)	0.882	0.252	*0.686*
GraphDTA[19]	CNN	GNN	*0.893*	0.229	—
DSAGLSTM-DTA(Distributed)	BiLSTM	GNN	**0.904**	**0.208**	**0.711**
DSAGLSTM-DTA(Centralized)	BiLSTM	GNN	**0.906**	**0.206**	**0.718**

Table 2**.** Italics represent the best data of the baseline model, and bolds represent the data that is better than the baseline model. The models in the above table are arranged in descending order of MSE (The following table is the same).

The proposed model was also evaluated on the KIBA dataset, which has more data and these data sources are more extensive, so the effects on the KIBA dataset are more convincing. The performance of the model on the KIBA dataset is presented in Table 3, and its experimental parameter settings are consistent with the Davis dataset. Similarly, the centralized architecture achieves the best performance in the evaluation of all models with an MSE of 0.131, a CI of 0.898 and a r_m^2 of 0.805. Although the improvement effect of the distributed architecture is not as obvious as that of the centralized architecture, it still has a considerable improvement. In addition, it should be emphasized that the r_m^2 of the centralized architecture exceeds 0.8 for the first time, which is a huge improvement over previous models.

Table 3. Results of various DTA prediction models on the KIBA dataset

Method	Protein	Compound	CI	MSE	r_m^2
Traditional methods					
KronRLS[15]	S-W	Pubchem	0.782	0.411	0.342
SimBoost[16]	S-W	Pubchem	0.836	0.222	0.629
1D Representation-based Approaches					
DeepDTA[17]	CNN(PS+PDM)	CNN(LS+LMCS)	0.863	0.194	0.673
WideDTA[18]	CNN	CNN	0.875	0.179	—
AttentionDTA[21]	CNN	CNN	0.882	0.155	*0.755*
2D Representation-based Approaches					
DeepGS[20]	GAT+Smi2Vec	CNN(Prot2Vec)	0.860	0.193	0.684
GraphDTA[19]	CNN	GNN	*0.891*	*0.139*	—
DSAGLSTM-DTA(Distributed)	BiLSTM	GNN	**0.893**	**0.134**	**0.786**
DSAGLSTM-DTA(Centralized)	BiLSTM	GNN	**0.898**	**0.131**	**0.805**

The baseline data in table 2 and table 3 above is obtained from [15-21]. From table 2 and table 3, it is not difficult to analyze that the 2D representation of drug is more advantageous than its 1D representation. In addition, the introduction of the self-attention block also greatly improves the comprehensive performance of the model. Finally, the approach of extracting features for drugs and proteins also has a big impact on the final result. In summary, the results of proposed model illustrate that our model has better performance than some other DTA models, which is of great significance to the research of DTA and will greatly promote the development of DTA.

3.2.2. Evaluation of Performance on Binary Classification Tasks

In order to demonstrate that our model has good generalization ability, we applied our model to two binary datasets, namely human and bindingDB for drug and target interaction experiments. The difference between the binary classification task and the regression task lies in that the final result is a label value (i.e., 0, 1) instead of a continuous value. Since our model is based on a regression task, we used the activation function sigmoid to map the predicted values to the interval (0, 1) when processing the output of the model, after that, the round function were utilized by us to convert the mapped values to interaction values. The binary classification model and its data are shown in Table 4 and Table 5.

Table 4. Performances of various DTI prediction approaches on the Human dataset.

Models	**AUROC**	**AUPRC**	**Precision**	**Recall**
K-NN[36]	0.860		0.927	0.798
RF[36]	0.940		0.897	0.861
L2[36]	0.911		0.913	0.867
SVM[36]	0.910		*0.966*	*0.969*
GraphDTA[19]	0.960±0.005		0.882±0.040	0.912±0.040
GCN[37]	0.956±0.004		0.862±0.006	0.912±0.010
CPI-GNN[38]	0.970		0.918	0.923
DrugVQA[39]	0.964±0.005		0.897±0.004	0.948±0.003
TransformerCPI[36]	*0.973±0.002*		0.916±0.006	0.925±0.006
DSAGLSTM-DTA(Distributed)	**0.982±0.002**	**0.983±0.002**	0.917±0.017	0.953±0.015
DSAGLSTM-DTA(Centralized)	**0.983±0.003**	**0.984±0.002**	0.917±0.014	0.950±0.007

In order to fully validate the performance of the model and prevent overfitting, we used five-fold cross-validation on the human dataset. In this work, we compared the model with some classic machine learning algorithms, including k-nearest neighbors (k-NN), random forest (RF), etc. In addition, some deep learning methods based on molecular graphs, such as CPI-GNN and DrugVQA, were also functioned as evaluation references. From the performance of the above models, it can be concluded that the two architectures of the proposed model far exceed the performance of other models on AUROC and AUPRC. Even though they are slightly lower than support vector machines (SVMs) on Precision and Recall, they still far superior to other models. In other words, after accounting for all the metrics, our model still performed best.

In addition to evaluating on the human dataset, another widely used binary classification dataset, BindingDB, was also considered for our measurement. The BindingDB dataset is characterized in that the dataset has been preprocessed in advance. Therefore, we directly conducted experiments on the preprocessed dataset. The evaluation results on the BindingDB dataset are summarized in Table 5. We selected some recent 2D molecular graph-based models for comparison, and the results illustrate that our two architectures outperform the baseline model on all metrics, which demonstrates the excellent adaptability of our model.

Table 5. Performances of various DTI prediction approaches on the BindingDB dataset.

Models	AUROC	AUPRC	Precision	Recall
CPI-GNN[38]	0.603	0.543		
GCN[37]	0.927	0.913		
GraphDTA[19]	0.929	0.917		
TransformerCPI[36]	*0.951*	0.949		
DSAGLSTM-DTA(Distributed)	0.952	0.950	0.913	0.844
DSAGLSTM-DTA(Centralized)	0.959	0.960	0.898	0.899

In conclusion, the superior performance of the proposed model on the two types of tasks exhibits the excellent generalization ability of the model. In addition, by comparing the performance of centralized and distributed architecture of proposed model on two types of tasks, it is not difficult to conclude that the centralized architecture is more competitive than the distributed architecture, which also verifies the point of Lee et al. Finally, the comparison with the data of GraphDTA also fully confirms the role of feature extraction and self-attention mechanism in improving the performance of the model, which thus demonstrates the effectiveness of our innovation.

4. CONCLUSION

In this work, we proposed a novel DTA prediction method named DSAGLSTM-DTA, which introduced self-attention mechanisms in different ways in the feature extraction process of drug molecular graphs, and formed two architectures called centralized and distributed. To capture the context dependencies in long amino acid sequences of proteins, a bidirectional LSTM network was applied. In addition, in order not to ignore the hidden features in the LSTM unit, an attention block was used as the weight calculation. We applied the model to regression tasks and binary classification tasks for drug and target prediction, respectively. Evaluation of the model on benchmark datasets demonstrated that the proposed model achieves superior performance to that of various existing DTA and DTI prediction methods, suggesting the effectiveness of the proposed approach in predicting the interaction of drug and protein pairs. Furthermore, it also demonstrated the good generalization ability of the extraction method as well as the effectiveness of the self-attention mechanisms.

REFERENCES

[1] A. Ezzat, M. Wu, X. L. Li, and C. K. Kwoh, "Computational prediction of drug-target interactions using chemogenomic approaches: An empirical survey," *Brief. Bioinform.*, vol. 20, no. 4, pp. 1337–1357, 2018, doi: 10.1093/bib/bby002.

[2] X. Chen *et al.*, "Drug-target interaction prediction: Databases, web servers and computational models," *Brief. Bioinform.*, vol. 17, no. 4, pp. 696–712, 2016, doi: 10.1093/bib/bbv066.

[3] A. S. Rifaioglu, H. Atas, M. J. Martin, R. Cetin-Atalay, V. Atalay, and T. Doğan, "Recent applications of deep learning and machine intelligence on in silico drug discovery: Methods, tools and databases," *Brief. Bioinform.*, vol. 20, no. 5, pp. 1878–1912, 2019, doi: 10.1093/bib/bby061.

[4] E. H. B. Maia, L. C. Assis, T. A. de Oliveira, A. M. da Silva, and A. G. Taranto, "Structure-Based Virtual Screening: From Classical to Artificial Intelligence," *Front. Chem.*, vol. 8, no. April, 2020, doi: 10.3389/fchem.2020.00343.

[5] M. Himmat, N. Salim, M. M. Al-Dabbagh, F. Saeed, and A. Ahmed, "Adapting document similarity measures for ligand-based virtual screening," *Molecules*, vol. 21, no. 4, pp. 1–13, 2016, doi: 10.3390/molecules21040476.

[6] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[7] A. Hazra, P. Choudhary, and M. Sheetal Singh, *Recent advances in deep learning techniques and its applications: An overview*. Springer Singapore, 2021.

[8] M. Wen *et al.*, "Deep-Learning-Based Drug-Target Interaction Prediction," *J. Proteome Res.*, vol. 16, no. 4, pp. 1401–1409, 2017, doi: 10.1021/acs.jproteome.6b00618.

[9] K. Tian, M. Shao, S. Zhou, and J. Guan, "Boosting compound-protein interaction prediction by deep learning," *Proc. - 2015 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2015*, pp. 29–34, 2015, doi: 10.1109/BIBM.2015.7359651.

[10] Y. Du, J. Wang, X. Wang, J. Chen, and H. Chang, "Predicting drug-target interaction via wide and deep learning," *ACM Int. Conf. Proceeding Ser.*, pp. 128–132, 2018, doi: 10.1145/3194480.3194491.

[11] B. Shin, S. Park, K. Kang, and J. C. Ho, "Self-Attention Based Molecule Representation for Predicting Drug-Target Interaction," pp. 1–18, 2019, [Online]. Available: http://arxiv.org/abs/1908.06760.

[12] A. Vaswani *et al.*, "Attention Is All You Need," *CoRR*, vol. abs/1706.0, 2017, [Online]. Available: http://arxiv.org/abs/1706.03762.

[13] Z. Quan, Y. Guo, X. Lin, Z. J. Wang, and X. Zeng, "GraphCPI: Graph Neural Representation Learning for Compound-Protein Interaction," *Proc. - 2019 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2019*, pp. 717–722, 2019, doi: 10.1109/BIBM47256.2019.8983267.

[14] W. Torng and R. B. Altman, "Graph Convolutional Neural Networks for Predicting Drug-Target Interactions," *J. Chem. Inf. Model.*, 2019, doi: 10.1021/acs.jcim.9b00628.

[15] T. Pahikkala *et al.*, "Toward more realistic drug-target interaction predictions," *Brief. Bioinform.*, vol. 16, no. 2, pp. 325–337, 2015, doi: 10.1093/bib/bbu010.

[16] T. He, M. Heidemeyer, F. Ban, A. Cherkasov, and M. Ester, "SimBoost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines," *J. Cheminform.*, vol. 9, no. 1, pp. 1–14, 2017, doi: 10.1186/s13321-017-0209-z.

[17] H. Öztürk, A. Özgür, and E. Ozkirimli, "DeepDTA: Deep drug-target binding affinity prediction," *Bioinformatics*, vol. 34, no. 17, pp. i821–i829, 2018, doi: 10.1093/bioinformatics/bty593.

[18] H. Öztürk, E. Ozkirimli, and A. Özgür, "WideDTA: prediction of drug-target binding affinity," 2019, [Online]. Available: http://arxiv.org/abs/1902.04166.

[19] T. Nguyen, H. Le, and S. Venkatesh, "GraphDTA: prediction of drug–target binding affinity using graph convolutional networks," *BioRxiv*, p. 684662, 2019, doi: 10.1101/684662.

[20] X. Lin, K. Zhao, T. Xiao, Z. Quan, Z. J. Wang, and P. S. Yu, "Deepgs: Deep representation learning of graphs and sequences for drug-target binding affinity prediction," *Front. Artif. Intell. Appl.*, vol. 325, no. i, pp. 1301–1308, 2020, doi: 10.3233/FAIA200232.

[21] Q. Zhao, F. Xiao, M. Yang, Y. Li, and J. Wang, "AttentionDTA: Prediction of drug-target binding affinity using attention model," in *Proceedings - 2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019*, Nov. 2019, pp. 64–69, doi: 10.1109/BIBM47256.2019.8983125.

[22] 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, doi: 10.1093/bioinformatics/btab715.

[23] S. Zhang, M. Jiang, S. Wang, X. Wang, Z. Wei, and Z. Li, "Sag-dta: Prediction of drug–target affinity using self-attention graph network," *Int. J. Mol. Sci.*, vol. 22, no. 16, 2021, doi: 10.3390/ijms22168993.

[24] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 6661–6670, 2019.

[25] M. I. Davis *et al.*, "Comprehensive analysis of kinase inhibitor selectivity," *Nat. Biotechnol.*, vol. 29, no. 11, pp. 1046–1051, 2011, doi: 10.1038/nbt.1990.

[26] J. Tang *et al.*, "Making sense of large-scale kinase inhibitor bioactivity data sets: A comparative and integrative analysis," *J. Chem. Inf. Model.*, vol. 54, no. 3, pp. 735–743, 2014, doi: 10.1021/ci400709d.

[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, doi: 10.1093/bioinformatics/btv256.

[28] K. Yingkai Gao, A. Fokoue, H. Luo, A. Iyengar, S. Dey, and P. Zhang, "Interpretable drug target prediction using deep neural representation," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 3371–3377, 2018, doi: 10.24963/ijcai.2018/468.

[29] D. S. Wishart *et al.*, "DrugBank: A knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Res.*, vol. 36, no. SUPPL. 1, pp. 901–906, 2008, doi: 10.1093/nar/gkm958.

[30] S. Günther *et al.*, "SuperTarget and Matador: Resources for exploring drug-target relationships," *Nucleic Acids Res.*, vol. 36, no. SUPPL. 1, pp. 919–922, 2008, doi: 10.1093/nar/gkm862.

[31] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated graph sequence neural networks," *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, no. 1, pp. 1–20, 2016.

[32] S. Hochreiter, "Long Short-Term Memory," vol. 1780, pp. 1735–1780, 1997.

[33] H. V. Adikane, R. K. Singh, D. M. Thakar, and S. N. Nene, "Single-step purification and immobilization of penicillin acylase using hydrophobic ligands," *Appl. Biochem. Biotechnol. - Part A Enzym. Eng. Biotechnol.*, vol. 94, no. 2, pp. 127–134, 2001, doi: 10.1385/ABAB:94:2:127.

[34] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, no. NeurIPS, 2019.

[35] M. Fey and J. E. Lenssen, “Fast Graph Representation Learning with PyTorch Geometric,” no. 1, pp. 1–9, 2019, [Online]. Available: http://arxiv.org/abs/1903.02428.

[36] 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, doi: 10.1093/bioinformatics/btaa524.

[37] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–14, 2017.

[38] 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, doi: 10.1093/bioinformatics/bty535.

[39] S. Zheng, Y. Li, S. Chen, J. Xu, and Y. Yang, “Predicting drug–protein interaction using quasi-visual question answering system,” *Nat. Mach. Intell.*, vol. 2, no. 2, pp. 134–140, 2020, doi: 10.1038/s42256-020-0152-y.