Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

System Biology

SG-DTA: Stacked Graph Drug-Target binding Affinity prediction

Cuong Vo [1,*], Tan Thanh Tran [1,*], Son Minh Nguyen [1,*] and Duc Hau Le [1,]

[1] Computational Biomedicine Department, VinBigData Institution, HaNoi, VietNam.

*To whom correspondence should be addressed. Equal contribution.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The development of new pharmaceuticals is expensive, time-consuming, and sometimes fraught with safety concerns. By identifying new uses for existing authorized pharmaceuticals, medication repurposing can circumvent the costly and time-consuming process of drug development. To efficiently repurpose pharmaceuticals, it is necessary to understand which proteins are targeted by specific medications. Currently, several papers agreed that using graphs representation will increase the accuracy due to the detail feature extracted from the input. Thus, we propose the Stacked Graph DTA (SG-DTA) that not only pushes the use of graph methods to the limit when using the stacking of both graph-representation and node-representation but also renders the higher accuracy model compared with anothers. Our result reaffirms the better of using graph embedding than any other input representation methods, specifically in the DTA task.

Results: We provide an end-to-end pair-wise input neural network, Stacked Graph Drug-Target Affinity (SG-DTA) algorithm, to leverage the relationship between drug and target. Our method achieves the remarkable result on both Davis and Kiba dataset when compares with the state-of-the-art methods. Moreover, the features generated from our method include the information of each neighbor and make a strong assumption that many drug's effects on a target might have relevant information and vice versa.

Availability: Source code is available at https://github.com/CuongVoThanh/SGDTI

Contact: v.cuongvt3@vinbigdata.org

Keyword: deep learning, drug-target binding affinity, graph neural network, bipartite graph

1 Introduction

As all of us know, the process of developing a new drug is time-consuming, this is more than ten years from target identification to drug approval (Rifaioglu *et al.*, 2019), and it costs billions of US Dollars (Mullard, 2014). Clearly, we need to compress the time as every problem in biomedical becomes more complex with high intensity as in the moment, for example, the COVID-19 pandemic.

Fortunately, the exploding digitization of data, which can be retrieved from a variety of sources ranging from clinical pharmacology to cheminformatics-driven databases, has reshaped biomedical discovery. This advantages approach has been accelerating the process of new research as well as improving and filling the gap that the previous study could not solve. Based on these advances, lots of methods have been studied in order to quicken the progress of drug development, such as de novo drug design and drug repurposing, etc.

At the moment, several computational techniques for drug-protein affinity prediction have been developed such as molecular docking method, non-parametric machine learning algorithms such as Random Forest (Ballester and Mitchell, 2010; Li *et al.*, 2015; Shar *et al.*, 2016), etc. In addition, collaborative filtering is another approach, the affinity similarities between medicines and targets might be attributed to other sources, for example, the Kernel-based techniques employ kernels constructed from molecular descriptors of medicines and targets inside a regularized least squares regression (RLS) framework (Cichonska *et al.*, 2017, 2018). The KronRLS model computes a pairwise kernel K from the Kronecker product of the drug-by-drug and protein-by-protein kernels to accelerate model training (Cichonska *et al.*, 2017, 2018). However, these methods have some disadvantages on the input or features selection. Specifically, although the molecular docking technique is instructive, it requires information and

concrete knowledge of the crystalline structure of proteins, which may not be available. Or in the Random Forest algorithm, features such as atom-pair co-occurrence oversimplified the description of the protein–ligand complex and resulted in the loss of information (Öztürk *et al.*, 2018).

Along with the explosion of data, deep learning step by step became a popular architecture recently, supported by high-capacity computing devices that challenged existing machine learning approaches. Deep learning methods are currently being used extensively in many other research fields, including bioinformatics such as genomics studies (K.K.Leung *et al.*, 2014; Y.Xiong *et al.*, 2015) and quantitative structure-activity relationship (QSAR) studies in drug discovery, inspired by the remarkable success rate in image processing and speech recognition (Ma *et al.*, 2015). The main advantage of deep learning architectures is that they offer improved representations of raw data through non-linear modifications in each layer (LeCun *et al.*, 2015), making it easier to understand the underlying patterns in the data.

A few researches have previously been conducted utilizing Deep Neural Networks (DNN) for DTI binary class prediction utilizing multiple input models for proteins and medicines, in addition to some researches that use stacked auto-encoders (Wang *et al.*, 2018) and deep-belief networks (Wen *et al.*, 2017). Similarly, stacked auto-encoder models incorporating Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) were used to describe chemical and genomic structures in real-valued vector forms (Gómez-Bombarelli *et al.*, 2018; Jastrzębski *et al.*, 2016). Deep learning algorithms have also been applied to protein-ligand interaction scores, with CNNs learning from the 3D structures of protein-ligand complexes being a frequent use (Gomes *et al.*, 2017; Ragoza *et al.*, 2017; Wallach *et al.*, 2015). This technique, however, is confined to known protein-ligand complex structures, with only 25000 ligands described in PDB (W.Rose *et al.*, 2016).

Besides, some known methods has using on 1D representations of drug and protein sequences. For example, the DeepDTA model captures predictive patterns in data using 1D representations and layers of 1D convolutions with pooling (Öztürk *et al.*, 2018). The final convolution layers are then concatenated, processed through many hidden layers, and regressed with drug–target affinity ratings. The WideDTA model is an expansion of the DeepDTA model in which the drug and protein sequences are first summarized as higher-order characteristics (Öztürk *et al.*, 2019). Drugs, for example, are represented by the most common sub-structures the Ligand Maximum Common Substructures (LMCS) (Woźniak *et al.*, 2018), whereas proteins are represented by the most conserved sub-sequences (the Protein Domain profiles or Motifs (PDM) from PROSITE (J.A.Sigrist *et al.*, 2009)). While the conducting of a latent feature vector for each protein is a trend of WideDTA (Öztürk *et al.*, 2019) and DeepDTA (Öztürk *et al.*, 2018), representing proteins of the PADME model (Feng *et al.*, 2019) by using fixed-rule descriptors and outperforms DeepDTA (Öztürk *et al.*, 2018).

However, the 1D representation has its own disadvantage when it might lead to the loss of information about ligand and protein structure. Besides, the traditional alignment between these two objects would not depict in the right way how molecules and protein are combined. Previously, Thin et al. (Nguyen *et al.*, 2020a) improved the 1D input of the drug by embedding the SMILES structure into the graph, then learning the molecule's features using the graph-embedding method. By using this way, the characteristic of the drug will not be lost when aligning the drug structure matrix with the learned feature of the embedded protein. The model might not be stable for further use because the 1D protein is kept for later fusion between them. Some other approaches such as the early fusion of the GEFA architecture (Nguyen *et al.*, 2020b), tried to fill the gaps of target representation by using the 2D protein contact maps then pushing the combination of embedding features and the maps to the layers of GCN for extracting the protein characteristic before aligning to the given molecules. Compared to the GraphDTA, it seems to have better intuition and the way proteins are represented, but it cannot maintain the accuracy of the task.

In this paper, we proposed an upgrade version by exploiting the better use of graph representation for increasing the accuracy of the previous model GraphDTA by Thin et al. (Nguyen *et al.*, 2020a), meanwhile maintaining the dynamic of any combination drug-protein. As compared with the base model, our architecture has outperformed due to the different combination method. Specifically, we propose the extension of node-embedding to capture as much as possible the characteristics of the connection between molecule and protein target instead of the traditional alignment. In the next section, we discuss the data representation as well as the model architectures and their complements that will be implemented. The third section is our experiment on parameter setting, model training and evaluation metrics. In the fourth section, the model result and comparison will be discussed.

2 Materials and Methods

2.1 Data Description

This work reused the two popular datasets that have been leveraged by lots of previous working papers, specifically the Davis dataset (Mindy I Davis *et al.*, 2011) and the Kiba dataset (Tang *et al.*, 2014). These two datasets are also known as the benchmark dataset for binding affinity prediction and evaluation. For the main purpose of taking advantage of graph representation, SMILES and Protein Sequence will be used to build up the relevant graphs and used as the input for the training process.

The Davis dataset includes selectivity assays for the kinase protein family and their inhibitors, as well as their affinities observed between them which are measured by the dissociation constant (Kd) values. It consists of 442 protein interactions and 68 ligand interactions. The Kiba dataset, on the other hand, was created using a method known as Kiba, which included kinase inhibitor bioactivities from several sources such as Ki, Kd, and IC50. By employing the statistical information included in Kiba scores, the consistency between Ki, Kd, and IC50 was optimized. Originally, the Kiba dataset has 467 targets and 52 498 medicines. The information of datasets that we employed in our study is summarized in Table 1. Besides, instead of using the original dissociation constant (Kd) values in the Davis dataset, we use the log-transformed version as in the study of He *et al.*, 2017a.

Table 1. Summary of the datasets

	Proteins	Compounds	Interactions
Kiba	229	2 111	118 254
Davis (K_d)	442	68	30 056

$$pK_d = \log_{10}\left(\frac{k_d}{1e9}\right) \tag{1}$$

2.2 Methods

In this section, we are going to describe the end-to-end architecture of the SG-DTA model in detail. The goal of our work is to learn by exploring the relationship between drug and target to improve the performance of the task. We will demonstrate the benefits of our work in section 3.3.

The general setting of the SG-DTA model to predict the binding affinity between the target protein P and drug compound D depicts the entire SG-DTA architecture as the function mapping f in Figure 1. Let y be the

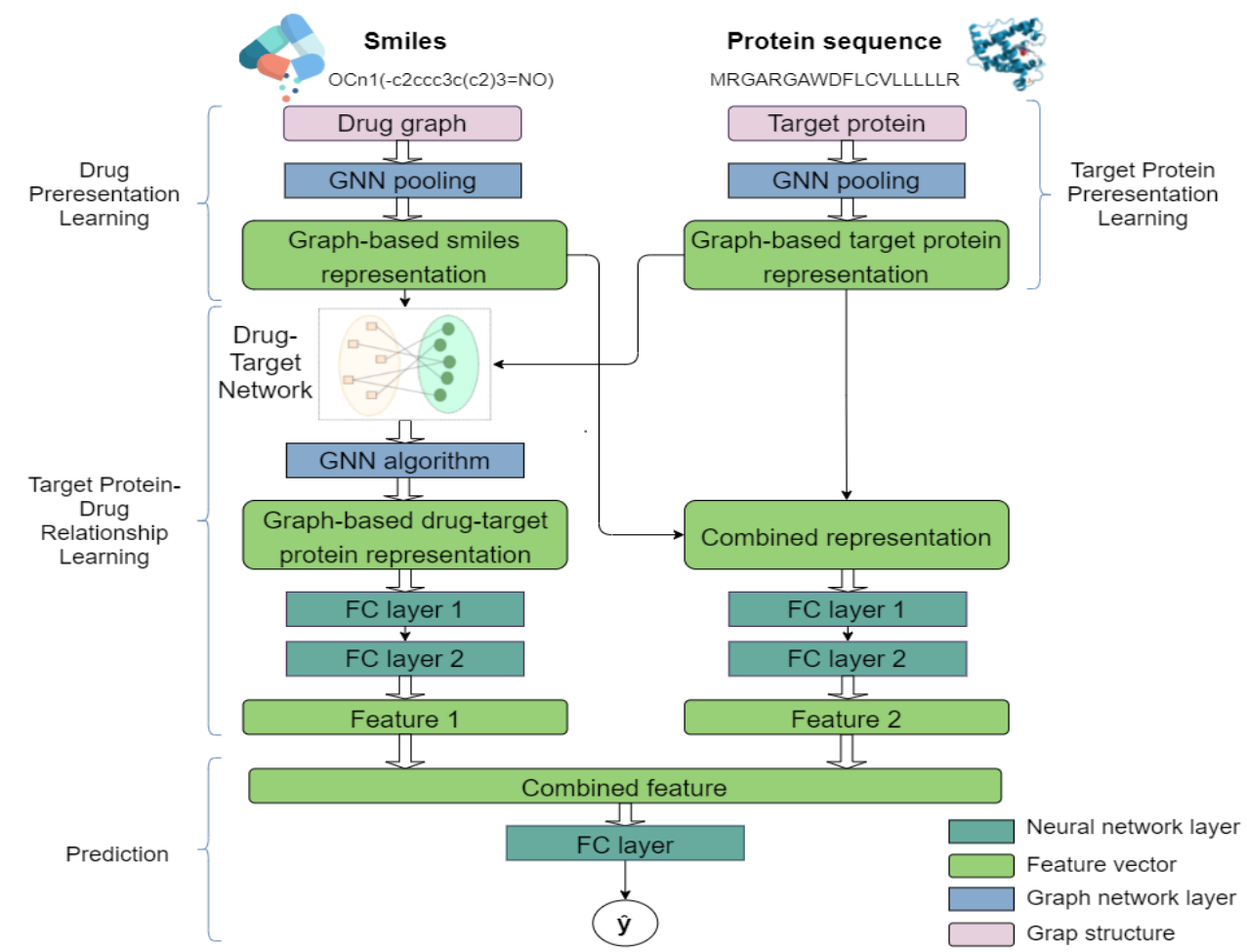

Fig. 1. Illustration of the Stack Graph DTA (SG-DTA) for drug and protein binding affinity prediction. SG-DTA takes as input a graph representation of the drug molecule and protein target. We first use a deep learning algorithm to learn a graph representation to refine the drug feature and protein-target feature. The two representation vectors are concatenated and passed through several fully connected layers to estimate the value of the first feature. The second feature is obtained by using graph representation algorithms to learn the features of drug and protein-targeted interaction graphs. Finally, the two features are paired and over some layers are fully connected to estimate the drug-target output affinity value.

affinity score showing how the strength of the binding force, function f maps the input vector P and D to y with parameter θ, which formulates as the equation below:

$$y = f_\theta(D, P) \tag{2}$$

In section 2.2.1, we briefly introduce the overview of various well-known Graph Neural Network algorithms. We also discussed the pros and cons of distinct methods. In the two next sections 2.2.2 and 2.2.3, we concisely present the mechanism to represent drug and protein features into the vector space relied on in section 2.2.1. And the last section, we explain details of the SG-DTA model and also give the flexible setting for it.

2.2.1 Background on Graph Neural Networks

Graph Neural Networks (GNNs) is one of the most powerful frameworks for representation learning of graphs. Graphs could both learn the elements of a system and their relationship through a set of nodes and edges. The features output of GNNs methods aggregates by the neighbors of this element. It turns out to be dependencies across instances and beyond the independent and identically distributed assumption. Let $\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{X})$ is represented as a graph in common where $\mathcal{V}$ is a set of N nodes concerning node feature vectors $\mathcal{X}_v \in \mathbb{R}^{N \times D}$ (D is the dimension of features each node) and adjacency matrix $\mathcal{A}$ indicates the connection of nodes in the graph. Learning a representation vector of a node v or the entire graph $\mathcal{G}$ can be utilized the graph structure and node features $\mathcal{X}_v$ in GNNs. Following a neighborhood aggregation strategy, the information of a node is iteratively updated in several steps. Assume that we choose l iterations to aggregate information, a node's representation captures the l-hop network neighborhood structural information. There are several ways to exploit and aggregate the relationship between each node. In this work, we briefly introduce four regular mechanisms in graph research as follows.

Graph Convolution Neural Network Graph Convolution Neural Network (GCN) was first proposed in (Kipf and Welling, 2017), which endeavors to mimic the convolution network scheme in the Computer Vision area. All of the neighbors and themselves are aggregated and combined as the formula below.

$$h_v^{(l+1)} = \sigma(W.Mean(h_u^{(l)}, \forall u \in N(v) \cup \{v\})). \tag{3}$$

where σ is any non-linear activation function, such as the $ReLU$ while W is the trainable weight matrix. Concretely, in the first step we initialize $h_v^{(0)} = \mathcal{X}_v$. Then, k-hop layer-wise convolution operation is established as follows

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{\mathcal{A}}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}). \tag{4}$$

The term $\widetilde{D}^{-\frac{1}{2}}\widetilde{\mathcal{A}}\widetilde{D}^{-\frac{1}{2}}$ is completely equivalent to $Mean$ aggregation method (Eq. 4) between neighbors and itself where $\widetilde{\mathcal{A}} = \mathcal{A} + I_N$ is the adjacency matrix $\mathcal{A}$ of bi-directed graph $\mathcal{G}$ with self-connection identity matrix I_N. $\widetilde{D}_{ii} = \sum_j \widetilde{\mathcal{A}}_{ij}$ is the degree matrix of $\mathcal{A}$. Finally, the encoding node embedding feature is the last layer-wise: $Z = H^L$ where L is the number of layers of the model. However, the limitation of this approach is based on the assumption that equal importance of self-connections vs edges to neighboring nodes while it is not true for graphs when there are relationships that are more important than others.

Graph Attention Network Attention mechanism (Bahdanau *et al.*, 2015) is usually known as self-attention or intra-attention which is the main factor that makes Graph Attention Network (GAT) more outstanding than previous approaches is that not all node's neighbors are equally important. That leads to a change of Eq. 3 a new formula.

$$h_v^{(l+1)} = \sigma\Big(\sum_{u \in N(v)} \alpha_{vu} W^{(l+1)} h_u^{(l)}\Big). \tag{5}$$

In terms of attention α_{vu}, concentrates on the important parts of the input data and fades out the rest. When explicitly defining $\alpha_{vu} = \frac{1}{|N(v)|}$, all neighbors $u \in N(v)$ are equally important to node v. We need a new implicitly definition for α_{uv} (such as a neural network) to specify arbitrary importance to different neighbors. A weight matrix $W \in \mathbb{R}^{D' \times D}$ is applied to every node to extract high-level features of node input features. Then an attention mechanism functions a (such as a neural network) maps $\mathbb{R}^{D' \times D'} \mapsto \mathbb{R}$ shows the importance of node u to node v.

$$e_{vu} = a(W^{(l)} h_u^{(l-1)}, W^{(l)} h_v^{(l-1)}) \tag{6}$$

The attention coefficients e_{vu} give exact information about the first-order neighbors of v (including v). Then a soft-max normalization is applied to e_{vu} to convert the value to the probability of the importance between u and v.

$$\alpha_{vu} = softmax_u(e_{vu}) = \frac{exp(e_{vu})}{\sum_{k \in N(v)} exp(e_{vk})} \tag{7}$$

Finally, the final representation vector is the output vector after stacked k-hop layers $Z = H^L$ (Eq. 5).

Furthermore, multi-head attention (Vaswani *et al.*, 2017) could also apply to stabilize the learning process of self-attention. To put it simply, multi-head attention computes K independent attention mechanisms (Eq. 5) and aggregates AGG all the features by concatenation or summation methods, as the following formula:

$$h_v^{(l+1)} = AGG_{k=1}^{K}\Big(\sigma\Big(\sum_{u \in N(v)} \alpha_{vu} W^{(l+1)} h_u^{(l)}\Big)\Big). \tag{8}$$

Graph Isomorphism Network However, the above techniques have a drawback that relates to the isomorphism property of the graph. The disadvantage is shown clearly that the aggregation information of a node is indistinguishable two different isomorphism graphs, though they should be embedded in the different vectors in space. Recently, a noticeably more powerful algorithm is Graph Isomorphism Network (GIN) (Xu *et al.*, 2019). The advantage of GIN are low-dimensional node embedding and the parameters of the update function can be learned for the downstream tasks. GIN also can model injective multi-set functions. In order to maximize the discrimination among non-isomorphism graphs, GIN utilizes a multi-layer perceptron (MLP) to learn the difference and update the node features as the equation below where ϵ can be a learnable parameter or fixed scalar.

$$h_v^{(l+1)} = MLP^{(l+1)}\Big((1 + \epsilon^{(l+1)}).h_v^{(l)} + \sum_{u \in N(v)} h_u^{(l)}\Big). \tag{9}$$

2.2.2 Representation of Drug

The drug input of our model is in the Simplified Input Line Entry System format (SMILES) (Weininger, 1988) as a standard format of 1D drug's structure in much recent research (Öztürk *et al.*, 2018, 2019; Nguyen *et al.*, 2020a; Zhao *et al.*, 2019). We encode the molecular structure of the drug into vector representation of the drug via the **RDKit** tool. To make it more concrete, the drug SMILES D is a sequence of atoms, atoms degrees, and ligands between each atom (e.g CC(C)SC1=NN=C(C=C1)NN). The SMILES string is converted into a molecular graph by **RDKit**. In a single molecular graph, vertices are the atoms of a compound while edges are the bond connecting between atoms. The node feature after encoding comprises of five properties: atom elements, the degree of the atom in the molecule, which is the number of directly-bonded neighbors (atoms), the total number of H bonds to the atom, the number of implicit H bound to the atom and whether the atom is aromatic. The adjacency list of edges is the undirected graph describing the bonds between two atoms. After pre-processing, a drug compound data can be referred to as $\mathcal{G}_D = (\mathcal{V}_D, \mathcal{A}_D, \mathcal{X}_D)$, where $\mathcal{V}_D \in \mathbb{R}^{nd}$ is a set of N nodes of the graph, nd is the size of D and $\mathcal{A}_D \in \mathbb{R}^{nd \times nd}$ denotes the bond adjacency list. Belong to the nodes, a feature node matrix $\mathcal{X}_D \in \mathbb{R}^{nd \times n}$ is also essential for learning representation of drug where n is the dimensions of compound properties. Finally, we can leverage one of the methods (we denote GNN represents for graph layer) which is mentioned in section 2.2.1 or mixed between GAT and GCN. A layer embedding can establish as following.

$$h_v^{(l+1)} = GNN(\mathcal{G}_D(\mathcal{V}_D, \mathcal{A}_D, \mathcal{X}_D)). \tag{10}$$

The graph representation can achieve by a $READOUT$ function. A $READOUT$ function is utilized in the last layer to aggregate all information to global features.

$$Z_d = READOUT(\{h_v^{(L)} | v \in \mathcal{G}_\mathcal{P}\}). \tag{11}$$

$READOUT$ can be any function such as $GlobalMaxPooling$, $GlobalMeanPooling$ (Murphy *et al.*, 2018; Ying *et al.*, 2018).

2.2.3 Representation of Protein

On the other hand, the protein P is a sequence of amino acids. The sequence contains many ASCII characters and alphabet symbols associated with each amino acid type. We can capture the primary structure of the protein as previous approach (Öztürk *et al.*, 2018). This approach digitizes the protein sequence by one-hot encoding. The advantage of this approach is the low time complexity when embedding residues into a vector space. Vector encoded $P \in \mathbb{R}^{np \times m}$ where np is the sequence length after cutting or padding and e is then embedded via an embedding layer as the formula.

$$X_P = Embedding(P). \tag{12}$$

The output of the embedding operation is $X_P \in \mathbb{R}^{np \times e}$ and expresses information through some LSTM or Conv1D layers. By doing so, however, the output representation just simply covers the primary structure but not the tertiary structure. In addition, the similarity between elements is not concretely defined and this choice of representation fails to leverage the contextual dependencies between residues (Nguyen *et al.*, 2020b). To overcome this limitation, DGraphDTA (Jiang *et al.*, 2020) constructs the tertiary structure gives critical information in terms of drug-target interaction. By constructing 2D structure of protein via a contact map, we could gain more information about protein in the real world. The protein sequence is converted into a protein graph by **Pconsc4** (Michel *et al.*, 2019). The output of **Pconsc4** is the probability of whether the residue pair contacts. The adjacency matrix $\mathcal{A}_p \in \mathbb{R}^{nd \times nd}$ is created by the output of **Pconsc4** with a threshold. Also, the node feature consists of those properties: residue symbol, position-specific scoring matrix

(PSSM), residue weight, hydrophobicity of residue, whether the residue is aliphatic, aromatic, polar neutral, acidic charged or basic charged and the dissociation constant for the –COOH group, -NH3 group and any other group in the molecule. Subsequently, a protein can be presented as $\mathcal{G}_{\mathcal{P}} = (\mathcal{V}_P, \mathcal{A}_P, \mathcal{X}_P)$, where $\mathcal{V}_{\mathcal{P}} \in \mathbb{R}^{np}$ is a set of amino acids of a protein sequence, $\mathcal{A}_{\mathcal{P}}$ is the residues adjacency matrix and a feature node $\mathcal{X}_{\mathcal{P}} \in \mathbb{R}^{nd \times m}$ can enhance the learning capability where m is the dimension of protein features. Eventually, protein representation can be expressed as below

$$h_v^{(l+1)} = GNN(\mathcal{G}_P(\mathcal{V}_P, \mathcal{A}_P, \mathcal{X}_P)). \tag{13}$$

$$Z_p = READOUT(\{h_v^{(L)} | v \in \mathcal{G}_{\mathcal{P}}\}). \tag{14}$$

2.2.4 Stacked Graph DTA

The affinity score shows the strength of the binding interaction between a single biomolecule (e.g. protein) to its ligand binding partner (e.g. drug). Binding affinity is influenced by non-covalent intermolecular interactions such as hydrogen bonding, electrostatic interactions, hydrophobic and Van der Waals forces between the two molecules or by the presence of other molecules. Hence, we make the assumption that (i) many drugs affect the same a single biomolecule might have the relevant information and vice versa. We (ii) leverage both information from a distinct factor and information which aggregates from (i) as local features and global features, respectively, to make a powerful model. In addition, modern deep learning architectures are often designed according to (iii) the end-to-end approach. From then on, the learning information from the distribution of labels can be back-propagated through all parts of the model.

Figure 1 summarizes the entire architecture. Specifically, SG-DTA constructs a bipartite graph between drug and protein representation in sections 2.2.2 and 2.2.3. A bipartite graph gives information about the relationship between drugs and targets. We denote $\mathcal{G}_{bp} = (\mathcal{V}_{bp}, \mathcal{A}_{bp}, \mathcal{X}_{bp})$ as referred for drug-target bipartite graph, where $\mathcal{V}_{bp} \in \mathbb{R}^{(\upsilon)}$ is a set of drug and target nodes of the graph, $\mathcal{A} \in \mathbb{R}^{\upsilon \times \upsilon}$ indicates the interaction between drug and target where $\upsilon = nd + np$ and feature vectors $\mathcal{X}_{bp}$ is the output of graph-based representation of drug Z_d and protein Z_p. The embedding vector Z_{bp} satisfies (i) which is expressed as follow

$$Z_{gf} = h_v^{(l+1)} = GNN(\mathcal{G}(\mathcal{V}_{bp}, \mathcal{A}_{bp}, \mathcal{X}_{bp})). \tag{15}$$

On the other hand, we combine Z_d and Z_p by concatenating method to represent as local features Z_{lf} between each drug-target pairwise.

$$Z_{lf} = [Z_d, Z_p] \tag{16}$$

The output of SG-DTA attaining (ii) can be expressed as below where ρ is a fully connected non shared parameter layer with non-linear activation function

$$\hat{y} = \rho([\rho(Z_{lf}), \rho(Z_{gf})]) \tag{17}$$

During the training phase, the dataset is chunked into each batch. It turns out that we do not have enough information to compute feature vectors $\mathcal{X}_{bp}$ in a direct way because each batch consists of a full set of the id of neither drug nor protein. It leads to the limitation that the implicit way to define $\mathcal{X}_{bp}$ and the intractable derivative term $\frac{\partial y}{\partial x_{bp}}$. To exceed this limitation, $\mathcal{X}_{bp}$ is initialized by a distribution $\mathscr{D}$, (In this work, $\mathcal{X}_{bp}$ is drawn from a Uniform distribution, $\mathcal{X}_{bp} \sim \mathcal{U}(a, b)$). Hence, we can gain (iii) directly without out of memory error when we try to compute $\frac{\partial y}{\partial x_{bp}}$ by all datasets.

Table 2. The hyperparameters setting for SG-DTA

Learning rate	0.0005
Learning rate decay	cosine
T_0	10
Optimizer	Adam
Adam(β_1, β_2)	(0.9, 0.999)
Weight decay	0.00001
Activation function	$GeLU$
Dropout rate	0.2
Epochs	4000
$\mathcal{U}(a, b)$	(0, 1)
Embedding size	128
Loss function	MSE

3 Results and Discussion

In this section, our proposed SG-DTA method experiences are evaluated using two public benchmark datasets (Section 2.1). We firstly mention our setting for hyper-parameter and well-known evaluation metrics in sections 3.1 and 3.2. Moreover, a full detail results and several case studies also discussed in the two last sections 3.3,

3.1 Parameter Setting

We select hyperparameters based on the performances of the validation set of k-fold. The training dataset is divided into a 5-fold cross-validation set. We train all models using Adam (P.Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, weight decay 0.00001 and batch size 512. We also use the Cosine Annealing Warm Restarts technique with $T_o = 10$ (Loshchilov and Hutter, 2017) to schedule the learning rate. We get learning rate $lr = 0.0005$ from the grid search over $lr \in \{0.00001, 0.0005, 0.0001, 0.005, 0.001, 0.05\}$. The values of all metrics in 3.2 are obtained after 49 and 193 steps for Davis and Kiba datasets respectively in each epoch. For dropout, we try dropping rates $d = 0.2$. The embedding size for SG-DTA is in a set $\{16, 32, 64, 128, 256, 512\}$ and models achieve the best results at size 128. We exclude L1 and Smooth L1 loss because of non-significant increasing performance. To recap, table 2 reports fully details all those hyperparameters settings.

3.2 Evaluation Metrics

For filtering and ranking several models in our experiment as well as comparing easily with other studies, some popular evaluation metrics have been used. There has four metrics which include Concordance Index (CI), Mean Square Error, Pearson and Spearman Rank Correlation.

Firstly, the Concordance Index (CI) has the vital role of evaluating the predictive accuracy of a nonlinear statistical model (Gönen and Heller, 2005). The formula of CI has been given as

$$CI = \frac{1}{Z} \sum_{d_i > d_j} h(b_i - b_j)$$

Normally, there is the predicted value b_n for each ground-truth value d_n, then we need to implement pairwise checking whether the predicted value is higher than the other or not. Because the result of this subtraction can be negative, the step function $h(x)$ would be employed to hands-on the positive index for ranking. In which, $h(x)$ is defined as below

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

Table 3. The result of quantitative experiments in Davis dataset

Method	Architecture	MSE↓	Pearson↑	Spearman↑	CI↑
DeepDTA (Öztürk *et al.*, 2018)	CNN & CNN	0.24050	0.84260	0.69339	0.88618
SG-DTA (our)	CNN & CNN & GCN	**0.22983**	**0.84462**	**0.71210**	**0.89806**
	CNN & CNN & GAT	**0.22300**	**0.85016**	**0.71564**	**0.90063**
	CNN & CNN & GAT-GCN	**0.21762**	**0.85495**	**0.70868**	**0.89600**
GraphDTA (Nguyen *et al.*, 2020a)	GIN & CNN	0.22818	0.84649	0.70839	0.89580
SG-DTA (our)	GIN & CNN & GCN	**0.22333**	**0.85002**	**0.71319**	**0.89863**
	GIN & CNN & GAT	**0.21936**	**0.85246**	**0.71903**	**0.90192**
	GIN & CNN & GAT-GCN	**0.22175**	**0.85052**	**0.71679**	**0.90095**
DGraphDTA (Jiang *et al.*, 2020)	GCN & GCN	0.21238	0.85850	0.70696	0.89619
SG-DTA (our)	GCN & GCN & GCN	**0.21224**	0.85841	**0.71322**	**0.89931**
	GCN & GCN & GAT	**0.20946**	**0.86081**	**0.72415**	**0.90580**
	GCN & GCN & GAT-GCN	0.22880	0.84789	0.69686	0.88976

Table 4. The result of quantitative experiments in Kiba dataset

Method	Architecture	MSE↓	Pearson↑	Spearman↑	CI↑
DeepDTA (Öztürk *et al.*, 2018)	CNN & CNN	0.14362	0.88816	0.88181	0.88865
SG-DTA (our)	CNN & CNN & GCN	**0.14092**	**0.89108**	**0.88741**	**0.89453**
	CNN & CNN & GAT	**0.14285**	**0.88890**	**0.88790**	**0.89549**
	CNN & CNN & GAT-GCN	0.14717	0.88520	0.87967	**0.89041**
GraphDTA (Nguyen *et al.*, 2020a)	GAT-GCN & CNN	0.13912	0.89180	0.88389	0.88929
SG-DTA (our)	GAT-GCN & CNN & GCN	**0.12983**	**0.89974**	**0.89336**	**0.89786**
	GAT-GCN & CNN & GAT	**0.12714**	**0.90223**	**0.89778**	**0.90211**
	GAT-GCN & CNN & GAT-GCN	0.15613	0.87796	0.86779	0.87959
DGraphDTA (Jiang *et al.*, 2020)	GCN & GCN	0.12608	0.90283	0.89566	0.90089
SG-DTA (our)	GCN & GCN & GCN	**0.12505**	**0.90375**	**0.89772**	**0.90314**
	GCN & GCN & GAT	**0.12636**	0.90265	**0.89816**	**0.90318**
	GCN & GCN & GAT-GCN	**0.12653**	0.90247	**0.89653**	**0.90209**

and Z is the normalization constant.

Secondly, the MSE is calculated in order to measure the mean square error of the predicted value. Given that, $\hat{y_i}$ is the predicted value, y_i is the true value, the formula of MSE is constructed as below:

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

Lastly, two types of correlation are measured between the ground truth and predicted value to see the general pattern of the predicted process, or in another way, these two measurements show how good the model is in the out-of-sample data set. The first measurement is Pearson correlation for linear correlation:

$$Pearson = \frac{\phi(\hat{y}, y)}{\phi(\hat{y})\phi(y)}$$

in which, $\phi(\hat{y}, y)$ is the covariance between the true and predicted affinity value.

The other is Spearman for measuring the rank correlation:

$$SpearmanRank = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

where d_i is the difference between predicted and ground-truth value ordered in rank.

3.3 Results

To prove the benefit of our proposal method could bring, we re-conduct several previous state of the art the methods such as DeepDTA (Öztürk *et al.*, 2018), GraphDTA (Nguyen *et al.*, 2020a), and DGraphDTA (Jiang *et al.*, 2020). Those methods have their own different ways to represent drug and protein feature in the vector space. Then two represented vectors are combined together and ingest into some fully connected layers and output the prediction. This is the common way to exploit the pair drug-target representation. We continue to drop out this combination and implement our method instead and show the effectiveness of our method in enhancing the performances. All evaluation metrics are mentioned in section 3.2. For serving the fairness comparison, these methods are

Table 5. Performances of various methods on Davis dataset

Method	Proteins and compounds	CI↑	MSE↓	Pearson↑	Spearman↑
KronRLS (Nascimento *et al.*, 2016)	S–W & Pubchem Sim	0.871	0.379	-	-
SimBoost (He *et al.*, 2017b)	S–W & Pubchem Sim	0.872	0.282	-	-
WideDTA (Öztürk *et al.*, 2019)	PS + PDM & LS + LMCS	0.886	0.262	0.820	-
MATT_DTI (Zeng *et al.*, 2021)	Rel_sa:CNN & CNN	0.890	0.229	-	-
SG-DTA (Our)	GCN & GCN & GAT	**0.906**	**0.209**	**0.861**	**0.724**

Note: Notation "-" marked on the table, due to the unavailable of some study's sources code and paper.

Table 6. Performances of various methods on Kiba dataset

Method	Proteins and compounds	CI↑	MSE↓	Pearson↑	Spearman↑
KronRLS (Nascimento *et al.*, 2016)	S–W & Pubchem Sim	0.782	0.411	-	-
SimBoost (He *et al.*, 2017b)	S–W & Pubchem Sim	0.836	0.222	-	-
WideDTA (Öztürk *et al.*, 2019)	PS + PDM & LS + LMCS	0.875	0.179	0.856	-
MATT_DTI (Zeng *et al.*, 2021)	Rel_sa:CNN & CNN	0.889	0.150	-	-
SG-DTA (Our)	GCN & GCN & GCN	**0.903**	**0.125**	**0.904**	**0.898**

Note: Notation "-" marked on the table, due to the unavailable of some study's sources code and paper.

implemented in the similar model system as well as the reproducibility of other study results based on their original sources code and papers.

Table 3 and Table 4 give information about the performance of the mentioned methods among the variety of evaluation metrics. Overall, the figures witness an increase from DeepDTA to GraphDTA and DGraphDTA. In DeepDTA, two vectors represented by leveraging CNN while GraphDTA the vector represented for drug utilizing GNN algorithms and DGraphDTA using both graph-based represented for drug and target. We can see clearly that the improvement throughout these methods. It is also readily apparent that when we keep the same architecture and adding our proposal method, the performance improves among four different evaluation metrics in both Davis and Kiba datasets. Results by leveraging the SG-DTA contribute to a performance improvement in general. It can be observed that the performance gains on SG-DTA with GCN and GAT algorithms. However, the model tends to overfit with mixed GAT-GCN and DGraphDTA (Jiang *et al.*, 2020). This is because the mixture GNN models increases the complexity of the model. Although our result is slightly superior to the DGraphDTA model on the Kiba dataset, SG-DTA still achieves an overall strong performance. Notably, for the Davis dataset, the CI, Pearson, and Spearman scores are better when compare with the others. SG-DTA also achieves the minimum error on the MSE metric at 0.209. This is also true for the Kiba dataset, while all the metrics outperform the others except the DGraphDTA model, specially for CI, with the increase nearly 0.003 in its figures. All in all, these produced result shows that SG-DTA obtains a good predictive performance.

Last but not least, we also establish two tables 5 and 6, which illustrate the comparison among some extra studies versus our proposed models. The figures for them are taken from the original paper. Difference from the others, our SG-DTA architecture puts the bipartite graph on in order to indicate the relationship between the two objects and indicates by the third part of the architecture. We consider just only the highest accuracy models in each methods and once again SG-DTA attains the best performance.

3.4 Cases Study Examples For CHEMBL1684800 And CHEMBL373751

In this case study, we select pairs that have not yet interacted in the training and test datasets to predict affinity. We then use the docking method to simulate the binding process between molecules and proteins. Here we use the tool **1-CLICK DOCKING** to simulate the docking process and predict the outcome. We chose the protein RAC-beta serine/threonine-protein kinase and drug and the drug with id CHEMBL1684800 to simulate docking because the affinity value of this pair is highly predicted by the model. As a result, you can see in Figure 2 that the protein and the drug are tightly bound together. Besides, we also tested on another pair when the affinity value predicted by our model is small, the result is as shown in figure 3 protein and drug are not tightly bound. close together but only standing close to each other. [1]

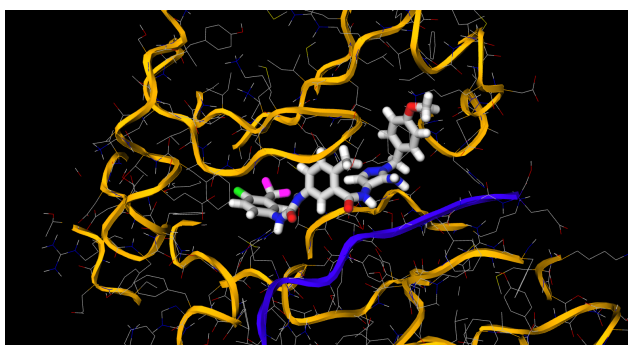

Fig. 2. 3D docking protein RAC-beta serine/threonine-protein kinase and drug with id CHEMBL1684800 generate by 1-click docking

[1] Here is the link for Click Docking tool: https://mcule.com/apps/1-click-docking

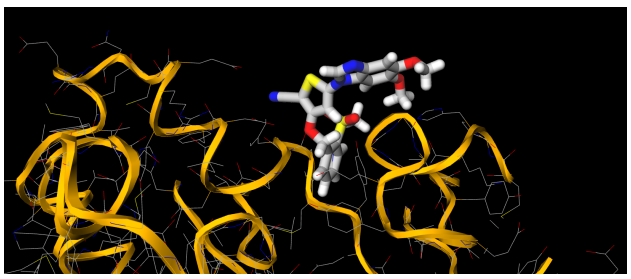

Fig. 3. 3D docking protein Tyrosine-protein kinase ABL2 and drug with id CHEMBL373751 generate by 1-click docking

4 Conclusion

In this paper, we have proposed the upgraded architecture model, which fully implementing the advances of graph representation, called SG-DTA (Stacked Graph for Drug-Target biding Affinity prediction). In which, both the protein and drug are represented under the graph method, and the binding level is exhibited using the bipartite graph (Figure 1). For the comparison purpose, we have measured the performance of our architecture using common methods such as Concordance Index, correlation, and the Mean Square Error. As the result, we found that the stacked block is the better feature extractor on the rendered bipartite graph, and it gives a better result on all chosen metrics. The result is also proven on both two datasets Kiba and Davis.

Our study opens the new idea of using the graph to exemplify the protein feature as well its relationship with drugs at lower cost while still reach a better result than the other. Again, the outperform of deep-learning methods, the graph representation is proven, and this paper opens the room for future research on re-implementing various deep-learning architectures from other fields to improve this crucial task for faster virtual drug screening and repurposing.

Acknowledgements

This work receives partial support from Vingroup Big Data Institute who gave us the valuable resources to do this project on the topic Drug-Target binding Affinity prediction.

Besides, we would like to thank pharmacist Yen Nguyen Thi Hai for the meaningful assist and the patient guidance on the ablation study. We also special thank Vuong Ho Ngoc and Dat Nguyen Quoc for contributing computational resources for experiments.

References

Bahdanau, D. *et al.* (2015). Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations (ICLR)*.

Ballester, P. J. and Mitchell, J. B. O. (2010). A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. *Bioinformatics*, **26**, 1169–1175.

Cichonska, A. *et al.* (2017). Computational-experimental approach to drug-target interaction mapping: A case study on kinase inhibitors. *Plos computational biology*.

Cichonska, A. *et al.* (2018). Learning with multiple pairwise kernels for drug bioactivity prediction. *Bioinformatics*, **34**, i509–i518.

Feng, Q. *et al.* (2019). PADME: A Deep Learning-based Framework for Drug-Target Interaction Prediction. *Cornell University*.

Gomes, J. *et al.* (2017). Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity. *Cornell University*.

Gómez-Bombarelli, R. *et al.* (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.*, page 268–276.

Gönen, M. and Heller, G. (2005). Concordance Probability and Discriminatory Power in Proportional Hazards Regression. *Oxford University Press*.

He, T. *et al.* (2017a). SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of Cheminformatics*.

He, T. *et al.* (2017b). SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. *Journal of Cheminformatics*.

J.A.Sigrist, C. *et al.* (2009). PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Research*, **38**, D161–D166.

Jastrzębski, S. *et al.* (2016). Learning to SMILE(S). *ICLR*.

Jiang, M. *et al.* (2020). Drug–target affinity prediction using graph neural network and contact maps. *Royal Society of Chemistry*.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *ICLR*.

K.K.Leung, M. *et al.* (2014). Deep learning of the tissue-regulated splicing code. *Bioinformatics*, **30**, i121–i129.

LeCun, Y. *et al.* (2015). Deep learning. *Nature*, **521**, 436–444.

Li, H. *et al.* (2015). Low-Quality Structural and Interaction Data Improves Binding Affinity Prediction via Random Forest. *Molecules*, **20**, 10947–10962.

Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic Gradient Descent with Warm Restarts. *ICLR*.

Ma, J. *et al.* (2015). Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.*

Michel, M. *et al.* (2019). PconsC4: fast, accurate and hassle-free contact predictions. *Bioinformatics*, **35**, 2677–2679.

Mindy I Davis, Jeremy P Hunt, S. H. *et al.* (2011). Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, **29**, 1046–1051.

Mullard, A. (2014). New drugs cost US $2.6 billion to develop. *Nat Rev Drug Discov 13, 877 (2014).*

Murphy, R. L. *et al.* (2018). Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *arXiv preprint arXiv*.

Nascimento, A. C. A. *et al.* (2016). A multiple kernel learning algorithm for drug-target interaction prediction. *Bioinformatics*.

Nguyen, T. *et al.* (2020a). GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, **37**, 1140–1147.

Nguyen, T. M. *et al.* (2020b). GEFA: Early Fusion Approach in Drug-Target Affinity Prediction. *Cornell University*.

P.Kingma, D. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. *ICLR*.

Ragoza, M. *et al.* (2017). Protein-Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.*, page 942–957.

Rifaioglu, A. S. *et al.* (2019). Recent applications of deep learning and machine intelligence on in silico drug discovery: methods, tools and databases. *Briefings in Bioinformatics*.

Shar, P. A. *et al.* (2016). Pred-binding: large-scale protein–ligand binding affinity prediction. *Journal of Enzyme Inhibition and Medicinal Chemistry*, pages 1443–1450.

Tang, J. *et al.* (2014). Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *J Chem Inf Model*.

Vaswani, A. *et al.* (2017). Attention Is All You Need. *Cornell University*.

Wallach, I. *et al.* (2015). Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *Cornell*

University.

Wang, L. *et al.* (2018). A Computational-Based Method for Predicting Drug–Target Interactions by Using Stacked Autoencoder Deep Neural Network. *Journal of Computational Biology*, **25**.

Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci. 1988, 28, 1, 31–36*.

Wen, M. *et al.* (2017). Deep-Learning-Based Drug–Target Interaction Prediction. *J. Proteome Res*, page 1401–1409.

Woźniak, M. *et al.* (2018). Linguistic measures of chemical diversity and the 'keywords' of molecular collections. *Scientific Reports*.

W.Rose, P. *et al.* (2016). The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Research*, **45**, D271–D281.

Xu, K. *et al.* (2019). How Powerful are Graph Neural Networks? *ICLR*.

Ying, R. *et al.* (2018). Hierarchical graph representation learning with differentiable pooling. *Advances in Neural Information Processing Systems (NIPS)*.

Y.Xiong, H. *et al.* (2015). The human splicing code reveals new insights into the genetic determinants of disease. *Science*, **347**.

Zeng, Y. *et al.* (2021). Deep drug-target binding affinity prediction with multiple attention blocks. *Briefings in Bioinformatics*.

Zhao, Q. *et al.* (2019). AttentionDTA: prediction of drug–target binding affinity using attention model. *2019 IEEE International Conference on Bioinformatics and Biomedicine*, pages 64–69.

Öztürk, H. *et al.* (2018). DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, **34**, i821–i829.

Öztürk, H. *et al.* (2019). WideDTA: prediction of drug-target binding affinity. *A PREPRINT*.