OXFORD

Briefings in Bioinformatics, 2022, **23**, 1–13
https://doi.org/10.1093/bib/bbab506
Problem Solving Protocol

FusionDTA: attention-based feature polymerizer and knowledge distillation for drug-target binding affinity prediction

Weining Yuan†, Guanxing Chen† and Calvin Yu-Chian Chen

Corresponding author: Calvin Yu-Chian Chen, Artificial Intelligence Medical Center, School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 510275, China; Department of Medical Research, China Medical University Hospital, Taichung, 40447, Taiwan; Department of Bioinformatics and Medical Engineering, Asia University, Taichung, 41354, Taiwan. Tel: +8615626413023; E-mail: chenyuchian@mail.sysu.edu.cn

†These authors contributed equally to this work.

Abstract

The prediction of drug-target affinity (DTA) plays an increasingly important role in drug discovery. Nowadays, lots of prediction methods focus on feature encoding of drugs and proteins, but ignore the importance of feature aggregation. However, the increasingly complex encoder networks lead to the loss of implicit information and excessive model size. To this end, we propose a deep-learning-based approach namely FusionDTA. For the loss of implicit information, a novel muti-head linear attention mechanism was utilized to replace the rough pooling method. This allows FusionDTA aggregates global information based on attention weights, instead of selecting the largest one as max-pooling does. To solve the redundancy issue of parameters, we applied knowledge distillation in FusionDTA by transfering learnable information from teacher model to student. Results show that FusionDTA performs better than existing models for the test domain on all evaluation metrics. We obtained concordance index (CI) index of 0.913 and 0.906 in Davis and KIBA dataset respectively, compared with 0.893 and 0.891 of previous state-of-art model. Under the cold-start constrain, our model proved to be more robust and more effective with unseen inputs than baseline methods. In addition, the knowledge distillation did save half of the parameters of the model, with only 0.006 reduction in CI index. Even FusionDTA with half the parameters could easily exceed the baseline on all metrics. In general, our model has superior performance and improves the effect of drug–target interaction (DTI) prediction. The visualization of DTI can effectively help predict the binding region of proteins during structure-based drug design.

Keywords: drug–target affinity, feature polymerizer, muti-head linear attention, model compression, knowledge distillation

Introduction

Drug discovery is a time-consuming, extremely expensive and gambling process. It takes more than 10 years and billions of dollars to develop new drugs, but 90% of the drugs entering clinical trials have not been approved by the FDA and entered the consumer market [1, 2]. In the past few decades, the rapid development of computer technology has enabled better drug design to assist drug design in experiments and accelerate the speed of drug development [3]. Nowadays, the key part of computer-aided drug design is to find matching drug molecules and proteins. Hence the drug-target interaction (DTI) has become a hot topic that has been widely studied [4].

Traditionally, virtual screening has been widely used to extract reasonable drug molecules from large compound databases. However, the molecular docking technology to measure the binding affinity between the drug and the target cost lots of time in the experiment [5]. For proteins with known structural information, drug molecules can be directly docked to obtain binding affinity. But there are still many proteins of unknown structure. Even if a large amount of time is spent on homology modeling, detailed structural information may not be obtained [6]. In response to this challenge, machine learning methods for drug–target affinity (DTA) prediction has gradually become an alternative to molecular docking.

Pahikkala *et al.* [7] proposed Kronecker regularized least-squares approach (KronRLS) that defined the similarity score of a drug–target pair through the Kronecker product of similarity matrix. He *et al.* [8] put forward Simboost, a cross-method that used a gradient booster to predict drug–target affinity. Öztürk *et al.* [9] suggested a deep learning model DeepDTA with two independent convolution blocks to learn representations from SMILES strings and protein sequences. Abbasi *et al.* [10] proposed a deep learning-based approach DeepCDA that combines convolutional layers and long short-term memory

Weining Yuan is a Bachelor in the School of Intelligent Engineering, Sun Yat-Sen University. His research interests focus on natural language processing, knowledge transfer and drug design.
Guanxing Chen is a Ph.D. candidate in the School of Intelligent Engineering, Sun Yat-Sen University. His research interests focus on explainable artificial intelligence, drug discovery, deep learning, biosynthesis, and vaccine design.
Calvin Yu-Chian Chen is the Dean of Intelligent Medical Center and a professor of school of intelligent systems engineering at Sun Yat-sen University. He also had been served as an Advisor or guest Professor in China Medical University, Massachusetts Institute of Technology (MIT), Peking University, University of Pittsburgh, and adjunct professor in Zhejiang University. His research interests include the computer vision, natural language processing and deep learning.
Received: August 8, 2021. **Revised:** October 21, 2021. **Accepted:** November 3, 2021

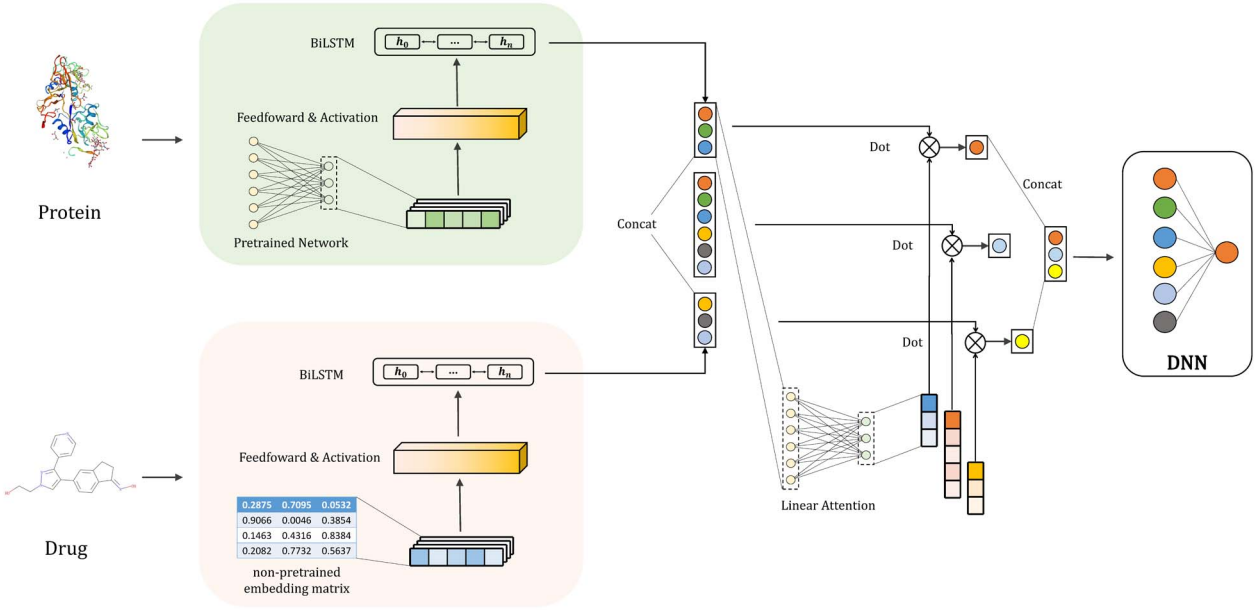

Figure 1. The overall architecture of FusionDTA. First, the original one-hot encoding of input vector is replaced by a novel distribution representation. Then, a feedforward layer and a LSTM is designed to construct the basic blocks of the encoder layer. Finally, the intermediate carriers of drug molecules and proteins are imported into the polymer layer to obtain an output carrier representation of binding affinity.

(LSTM) layers to effectively encode local and global temporal patterns for deep cross-domain compound-protein affinity prediction. Nguyen *et al.* [11] proposed a graph-based model GraphDTA encoding drug as an undirected graph with a feature map and an adjacent matrix. Graph convolutional network (GCN) [12], graph attention network (GAT) [13] and graph isomorphic network (GIN) [14] are designed to extract features from drugs, whereas convolution blocks are the feature encoder of protein.

Studies of attention-based methods also contribute to DTA prediction. DrugVQA [15] proposed a question-answering model for drug–target interaction tasks, in which a sequential attention mechanism is utilized to capture the dependency from dynamic convolutional neural network (CNN). From another perspective, MATT_DTI [16] designed a multi-head attention model that regarded drug representation as query while protein representation as key and value. Nguyen *et al.* [17] built a graph-in-graph architecture to fuse the drug–protein pair, with a self-attention mechanism to calculate the binding site in protein representation. Chen *et al.* [18] utilized a transformer decoder to translate protein sequence to interaction sequence, where protein representations are original texts and drug representations are previous translations. MT-DTI [19] proposed a new molecular representation method based on the self-attention mechanism, which is superior to the existing technology in terms of the area under the precise recall curve.

For pre-training of input vectors, Asgari and Mofrad [20] proposed a word2vec model Protvec to obtain the continuously distributed representation of proteins. Rao *et al.* [21] introduced the tasks assessing protein embeddings (TAPE) to evaluate semi-supervised learning on the protein sequence. In their study, self-supervised models should be tested on three mainstream tasks: structure prediction, detection of remote homologs and protein engineering. In addition, Rives *et al.* [22] learned a multiscale representation space from 86 billion amino acids across 250 million protein with a robust transformer ESM-1b. In existing work, one-dimensional (1D) CNN [23] and pooling method [24] are often applied to compress a sequence of n words in to a single token. However, each token contains unique semantic information. Crude use of 1D CNN layers or global pooling operations to aggregate features may result in the loss of a lot of useful information.

To solve this problem, we propose a novel neural network framework, FusionDTA. In model architecture, we first encode the inputs as continuously distributed representation depending on the raw input and the parameters of pre-trained model. For biological sequences, one-hot encoding cannot obtain the context information from a mass of unsupervised biological corpus. Thus, a pre-trained transformer is utilized to generate the distribute input representation in our work. Then, LSTM layers make up the basic block of encoder network. To capture the local and global dependencies of the feature vectors, we apply two-layers bi-directional LSTM on the feature map from embedding layers. Finally, we propose to replace the 1D CNN layer or the global pooling layer with a multi-head linear attention layer, which selectively focuses on each token from the entire biological sequence and aggregates global information based on the attention score. Different from the attention mechanism mentioned above, the proposed linear attention aims

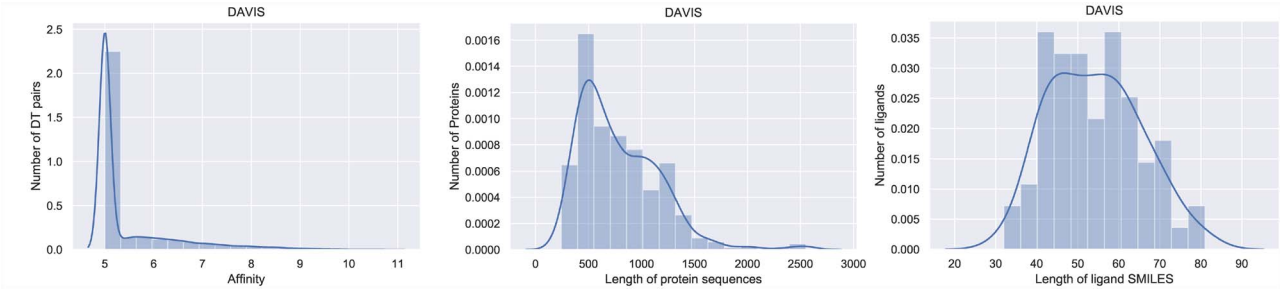

Figure 2. The frequency histogram of binding affinity, length of protein sequence and length of ligand SMILES in Davis dataset.

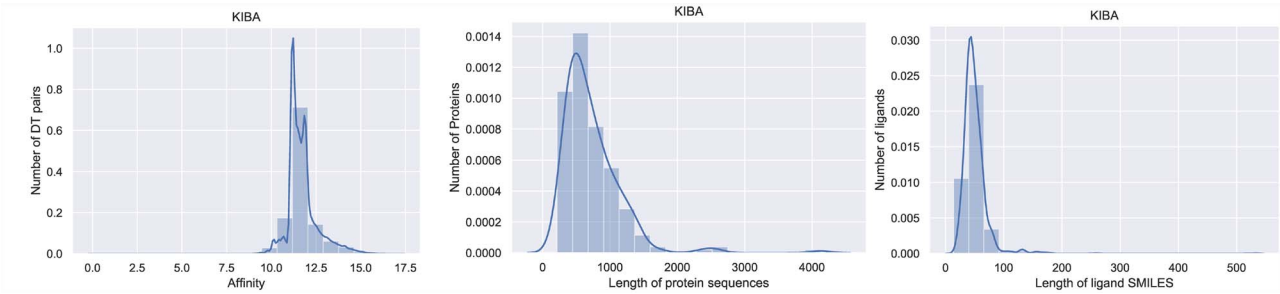

Figure 3. The frequency histogram of binding affinity, length of protein sequence and length of ligand SMILES in KIBA dataset.

to capture the direct reflection of each biological token on binding affinity rather than enhance the representational ability of the feature encoder.

With the deepening of the neural network encoder, we often face the phenomenon of excessive parameters in training process. This phenomenon is always accompanied by the problem of overfitting and slow training [25]. Therefore, we propose knowledge distillation for DTA tasks as an improvement in training strategy. Knowledge distillation establishes a teacher model and a student model. Through defining constraints and loss functions, the student model with less parameters obtains knowledge from the teacher model with more parameters. Through transferring knowledge from one model to another, knowledge distillation is an effective method for parameter regularization and model compression.

Material and methods

Datasets

We evaluated FusionDTA on two publicly available datasets, the Kinase dataset Davis [26] and KIBA dataset [27]. Both were regarded as benchmark datasets in previous drug–target affinity predictions.

- Davis dataset: Davis dataset contains 30 056 interactions from 442 proteins and 68 ligands, in which the binding affinity is evaluated by (K_d) value. It reflects the selective measurements of the kinase protein family and associated inhibitors with their constant values of dissociation.
  To solve the numerical explosion problem, Öztürk *et al.* proposed to replace the binding affinity value K_d with a novel measure pK_d by converting its value into the logarithmic domain. Specifically, K_d is first scaled to the appropriate range, and then the negative log is calculated as follows:

$$pK_d = -log_{10}\frac{K_d}{1e^9}. \quad (1)$$

  Figure 2 shows the histogram of affinity, drug length and protein frequency in the Davis data set. First graph illustrates the distribution of binding-affinity values of DT pairs in the DAVIS data set. Peaks with an affinity of 5 accounts for more than half of the data set. The dataset has a total of 30 056 DT pairs, of which 20 931 DT pairs have an affinity of 5. Most of the rest is distributed between 6 and 7. In addition, the length of most proteins is concentrated between 400 and 1500. The largest distribution is around 500, and the maximum length is 2549. The SMILES length of the ligands presents a Gaussian distribution, ranging from 35 to 80, most of which are between 40 and 60, and the maximum length is 103.
- KIBA dataset: KIBA dataset contains kinase inhibitor bioactivities measured by an approach called KIBA, which considers the different index of the inhibitor efficacy, such as K_i, K_d and IC_50. The binding affinity was measured by the interaction of 467 proteins and 52 498 ligands.
  Figure 3 shows the histogram of affinity, drug length and protein frequency in the KIBA data set. As shown in the figure, the affinities in the KIBA data set are mainly distributed between 10 and 13, and most of them fall around 11. The length of the protein sequence is concentrated between 200 and 1500, most of which are around 700, and the maximum length is 4128. The SMILES lengths of the ligands

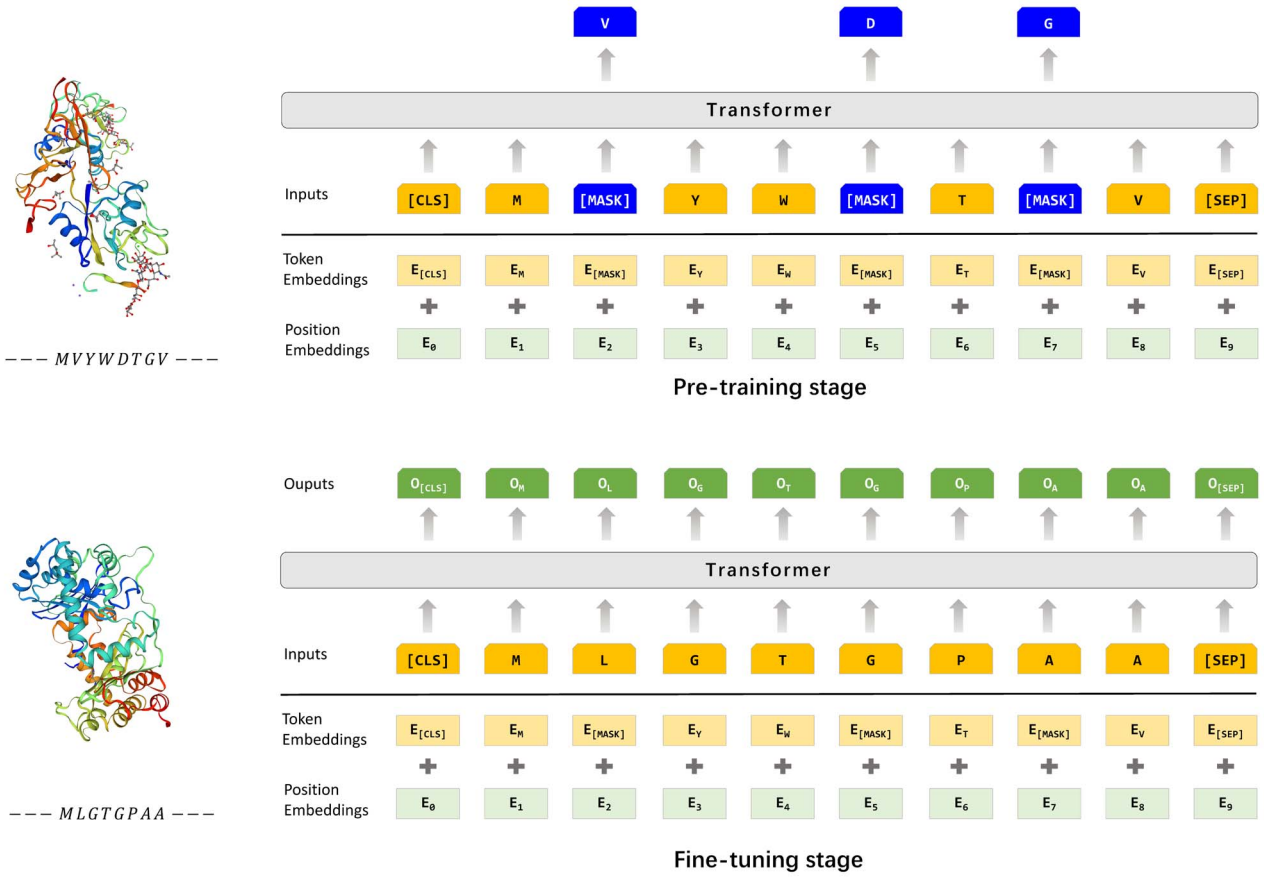

Figure 4. The training phase of the protein pre-training model. The top of the chart shows a strategy for the pre-training stage and the bottom shows a strategy for the fine-tuning stage.

range from 15 to 100, most of which are concentrated around 50 and the maximum length is 590.

Öztürk *et al.* proposed that for 99% of protein pairs, the Smith-Waterman (S-W) similarity between proteins in the KIBA data set is at most 60%. 92% of the protein pairs in the Davis data set have a target similarity of at most 60%. These statistics indicate that both data sets are non-redundant. To ensure the fairness of the experiment, 5-fold cross-validation was adopted in the experiment. All the data were divided into five parts on average, four parts for the training set and one part for the test set. Hence, a dataset can be divided into five schemes. We tested the proposed model on all schemes and regarded the average score as the final performance.

Model architecture

The overall architecture of FusionDTA is shown in Figure 1. The first step is to feed drug molecules and protein sequences into the embedding layer. In this layer, drug molecules are encoded as SMILES strings, and proteins are encoded as word embeddings. Then, the LSTM layers are designed to construct the basic blocks of the encoder layer. Finally, the intermediate carriers of drug molecules and proteins are imported into the fusion layer to obtain an output carrier representation of binding affinity.

Drug representation

For drug molecule, ASCII string SMILES [28] is widely used as a chemical describer for input representation. SMILES represent drug molecules as one-dimension sequence, from which the chemical properties of atoms and their arrangement is obtained. We project each SMILES character into a discrete space of one-hot encoding by creating a vocabulary in SMILES format. Each drug molecule is represented as follows:

$$x^D = \{x_1^D, \ldots, x_n^D\} \in \mathbb{R}^{V_D}, \tag{2}$$

where V_D is the vocabulary size in the format SMILES.

To avoid sparse matrix and high dimension, x^D is multiplied by a random embedding matrix. It convert x^D to a low-dimensional and dense continuous space e^D as follows:

$$e^D = \{e_1^D, \ldots, e_n^D\} \in \mathbb{R}^d, \tag{3}$$

where D is the dimension of drug in embedding layer.

Protein representation

With the development of natural language processing, pre-training of input vectors has become an indispensable part of the model [29]. The pre-trained model can

help the machine find an interpretable data representation to improve the performance of the algorithm. In the process of extracting biological sequence information, feature extraction can be conducted manually or in an unsupervised method. However, it is difficult to manually add various effective features to the biological sequence in the real scene, so a better choice is to use unsupervised learning to embed biological sequences into high-dimensional vectors [22].

Inspired by the ESM-1b [22], we borrow the pre-training transformer to replace the original one-hot encoding, which takes the distributed contextual vector as the protein representation. Figure 4 shows the overall pre-training and fine-tuning procedures. In pre-training stage, the original proteins are first divided into several sequences by a fixed maximum length. Each sequence begins with a token [CLS] and ends with a token [SEP]. Then, the input embeddings are the sum of token embeddings and position embeddings with a learnable weight. To capture the dependencies from tokens, some proportion of input tokens are masked at random and the final task of the pre-training model is to predict those masked tokens. Given the input sequence, we want to maximize the following negative log probability function:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{M} log\ p(m = m_i|\theta),\ m_i \in [1, 2, ..., |V_P|], \quad (4)$$

where M is the set of masked tokens and V_P is the vocabulary size of amino acids.

In the fine-tuning stage, the DTA task is regarded as the downstream task of protein pre-training. Similar to the pre-training stage, the protein is first divided into sequences with fixed maximum lengths. Then, these raw sequences are encoded into pre-trained ESM-1b, in which contextual dependencies are assigned to each amino acid. This allows the fine-tuning model to learn diverse knowledge from pre-training and fuse the sequential information at the biological word level. Finally, we utilize the top layer outputs of pre-trained ESM-1b as protein representation. Given the one-hot embedding $\{x_1^P, \cdots, x_m^P\} \in \mathbb{R}^{Vp}$, the output of the pre-training model is defined as follows:

$$e^P = \{e_1^P, \cdots, e_m^P\} \in \mathbb{R}^d, \quad (5)$$

where d is the dimension of the hidden layer in the pre-training model.

$$e^P = \{e_1^P, \cdots, e_m^P\} \in \mathbb{R}^d, \quad (6)$$

where V_p is the size of the vocabulary for amino acids and d is the dimension of the hidden layer in the pre-training model.

LSTM Layer

LSTM is a well-known variant of the recurrent neural network, which solves the long-term dependency problem of the general recurrent neural network (RNN) network [30]. For protein sequences and drug smiles, the input vector is represented as a set of multiple discrete biological words. Hence we can regard inputs as continuous time-series or sentences in the language model. In Davis and KIBA datasets, more than 80% of protein sequences exceed 200. Therefore, traditional methods (1D CNN or S-W) cannot extract high-level semantic features in a good and exact way. Due to the unique gate design, it is more suitable for LSTM to process longer biological sequence than vanilla RNN or hidden Markov models. Thus, we utilize LSTM as a feature encoder for the embeddings of drug and protein.

In our model, the embedding vectors of the sequence and SMILES are encoded into a two-layer bidirectional LSTM. First, the drug and protein embeddings are fed into the feedforward layer, which consists of a fully connected network and an activation function. The purpose of the feedforward layer is to map the features generated by the embedding layer into the space of the LSTM layer. Then, the LSTM layer is to capture the long-term dependence and short-term dependence from the feature map generated by the feedforward layer. Specifically, we apply two-layers bidirectional LSTM on the top of the feedforward layer. In addition, we superimpose the feedforward layer and the LSTM layer n times to obtain the local and global dependencies of feature vectors from different dimensions of semantic features. Since the bidirectional LSTM will bring the feature size multiplication, the input feature size of the LSTM layer is set to be half of the input feature size of the feedforward layer so as to maintain the consistency of the feature size.

Taking protein embedding as an example. Given an input sentence $\{e_1^P, \cdots, e_m^P\} \in \mathbb{R}^d$, the output of the LSTM layer is defined as $\{h_1^P, \cdots, h_m^P\} \in \mathbb{R}^F$, where F is the feature dimension of each biological token.

Muti-head linear attention mechanism

In existing work, 1D CNN is commonly applied to the output layer to ensure that protein sequences or SMILES of different lengths obtain the same size. In addition, GraphDTA recommends the use of a merge method to aggregate the characteristics of drug molecules, where the output vector can be calculated as the sum, the mean, or the maximum merge. However, powerful feature encoders often allocate feature maps for each token in a refined way. It means that the feature map of each token contains different shallow and in-depth semantic information. Roughly using 1D CNN layer or global pooling operations to compress the feature map may result in the loss of various information. Hence, we propose a novel multi-head linear attention mechanism to capture the meaningful information from each token.

Figure 5 shows the process of multi-head linear attention aggregation. As shown, the input vector is first

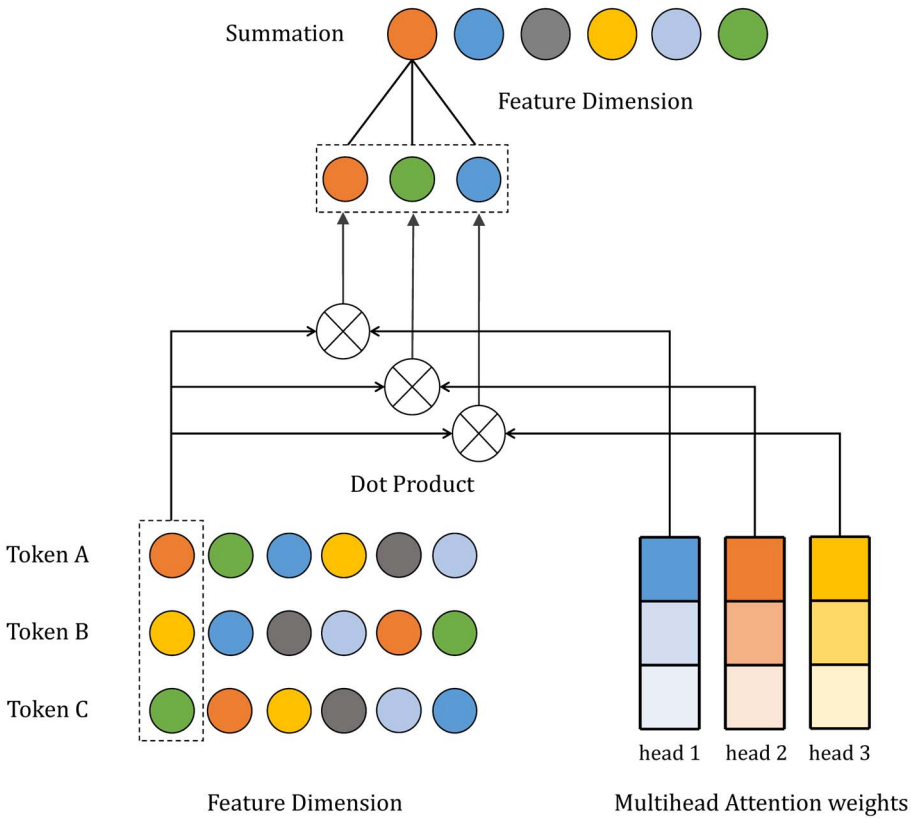

Figure 5. The process of multi-head linear attention layer aggregation. Each token is allocated n-heads attention weights. Then the output of each head is the dot product of tokens in dotted box and the specific head. A Summation formula is to aggregate the concatenation of n token into one. The fusion outputs can be expressed as the concatenation from each feature dimension.

mapped to the n-head attention vectors. We define the mapping function $LinearAttention(W, h_i)$ as follows:

$$LinearAttention(W, h_i) = \frac{\exp\left(\frac{Wh_i}{\sqrt{d_k}}\right)}{\sum_{j=1}^{m} \exp\left(\frac{Wh_j}{\sqrt{d_k}}\right)}, \tag{7}$$

where $W \in \mathbb{R}^{1\times F}$ is the attention weight matrix, d_k is the normalization coefficient.

Multi-head attention allows the machine to focus on information about the input features in different vector spaces and aggregate them as summations. For the coherence of the derivation, we first calculate the summation of n heads, instead of the dot product. Taking a protein as an input, we suppose the input vector is $\{h_1^P, \cdots, h_m^P\} \in \mathbb{R}^F$. Then the attention vector of n-heads, defined as $\{a_1^P, \cdots, a_m^P\} \in \mathbb{R}^1$, is calculated as follows:

$$a_i^P = \sum_{j=1}^{nheads} head_j, \tag{8}$$

$$head_j = LinearAttention(W_j, h_i^P). \tag{9}$$

Finally, the output of multi-head attention layer is defined as the dot product of multiple attention and the original input vector:

$$o^P = \sum_{i=1}^{m} a_i^P h_i^P. \tag{10}$$

Fusion layer

We propose a fusion layer composed of three multi-head linear attention block to fuse the properties of drug features and protein features. As is mentioned above, the multi-head linear attention can aggregate the drugs and proteins separately. However, when the feature maps of a protein or a drug aggregates independently, the relationship between the drug and the protein cannot be captured. Therefore, in this paper, three different linear attention blocks are applied in the fusion Layer for protein sequences, drug smiles and protein–drug information respectively.

Figure 1 also shows the mechanism of the fusion layer. As shown, the protein sequence and the drug Smiles were first spliced into a new sequence. Given the feature vector $h^P = \{h_1^P, \cdots, h_m^P\} \in \mathbb{R}^F$ for protein, the feature vector $h^D = \{h_1^D, \cdots, h_n^D\} \in \mathbb{R}^F$ for drug, the splicing vector is defined as $\hat{h} = \{h_1^P, \cdots, h_m^P, h_1^D, \cdots, h_n^D\} \in \mathbb{R}^F$. Then, the spliced sequence is fed into the multi-head linear

attention layer to obtain the aggregation feature $o^{PD} \in \mathbb{R}^F$ Similarly, the proteins and the drugs were respectively fed into the multi-head linear attention layer to obtain the features $o^P, o^D \in \mathbb{R}^F$. There are no shared parameters between the three attention layers. The formula derived above is expressed as follows:

$$\hat{h} = concat(h^P, h^D), \tag{11}$$

$$o^{PD} = MultiheadlinearAttn(\hat{h}), \tag{12}$$

$$o^P = MultiheadlinearAttn(h^P), \tag{13}$$

$$o^D = MultiheadlinearAttn(h^D). \tag{14}$$

Finally, a tensor concatenated by o^{PD}, o^P and o^D is fed into the fully connected layer as follows:

$$\hat{y} = FC(concat([o^{PD}, o^P, o^D])). \tag{15}$$

For the training stage, the goal is to make the prediction distribution as close to the ground truth as possible. Equivalently, we minimize the mean square error between the predicted value and ground truth:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2, \tag{16}$$

where N is the number of samples, y_i is the predicted value and $\hat{y}$ is the ground truth.

Knowledge distillation

It has been proved that knowledge distillation is an effective method to enhance the generalization of model and reduce the number of parameters [31, 32]. Regarding the DTA task as a regression problem, we apply the knowledge extraction mechanism in the training phase of FusionDTA and analyze its feasibility theoretically. On the one hand, self-knowledge distillation helps to improve the performance from the constrain of feature maps. On the other hand, knowledge distillation enhances a small-scale network to perform better than the same scale network without guides.

Knowledge distillation learning

Conceptually, we define a powerful network that has been trained as a teacher model, whereas networks of the same or smaller scale that can learn from the teacher model is a student model.

Figure 6 shows the training stage of the teacher and the student models. First, FusionDTA is trained as a teacher model via an effective network. We define target and drug inputs as X, affinity as Y, then what we are concerned about is function $f(x) : X \rightarrow Y$. In deep learning models, the function $f(x)$ is approximated by a parametrized function $f(x, \theta_1)$, where $\theta_1 \in \theta$. Specifically, stochastic gradient descent aims to learn the parameters θ_1^* by minimizing some objective function:

$$\theta_1^* = argmin_{\theta_1}\mathcal{L}(y, f(x, \theta_1)). \tag{17}$$

Second, we train a new network as a student model. The knowledge of the student model is obtained from both the teacher model and the real affinity. Therefore, the objective function of the student model consists of two-part, one is the loss measured by $f(x, \theta_2)$ and real target, the other is the loss measured by $f(x, \theta_2)$ and $f(x, \theta_1^*)$:

$$Loss1 = \mathcal{L}_1(argmin_{\theta_1}, \mathcal{L}(y, f(x, \theta_1)), \theta_2), \tag{18}$$

$$Loss2 = \mathcal{L}_2(y, f(x, \theta_2)), \tag{19}$$

$$\theta_2^* = argmin_{\theta_2}(\alpha Loss1 + (1 - \alpha)Loss2), \tag{20}$$

where α is the impact factor that determine the weight between *Loss*1 and *Loss*2.

Knowledge distillation for DTA task

In the abovementioned derivation, our ultimate goal is to minimize the objective function of the teacher. Due to the difference of outputs and the need of models, no universal L_1 can be found for each tasks. Hinton and Salakhutdinov [33] introduced a generalized softmax function as L_1, in which the concept of temperature was introduced to ensure that the softmax distribution generated by the teacher model is soft enough. In this way, the student model can extract knowledge from the softmax output distribution at a higher temperature, and then restore the low temperature during the test stage.

However, in the regression task, the real situation is a 1D continuous variable, rather than a 'hot' label. Therefore, the logit of the teacher model does not contain more information than the real situation. In other words, in DTA task, the student model will not learn additional hidden knowledge from the output logits of the teacher model.

To solve this problem, we suggest that the student model should learn transferable knowledge from the feature map of the hidden layer, instead of logits in the output layer. Define L_1 as follows:

$$\mathcal{L}_1(\theta_{Hint}, \theta_{Guide}) = \|g(x, \theta_{Hint}) - r(g(x, \theta_{Guide}), \theta_r)\|_2^2, \tag{21}$$

where g is a transfer function from the input x to the hidden layer with parameters θ_{Hint} and θ_{Guide}, r is a nonlinear regression function at the top of the guidance

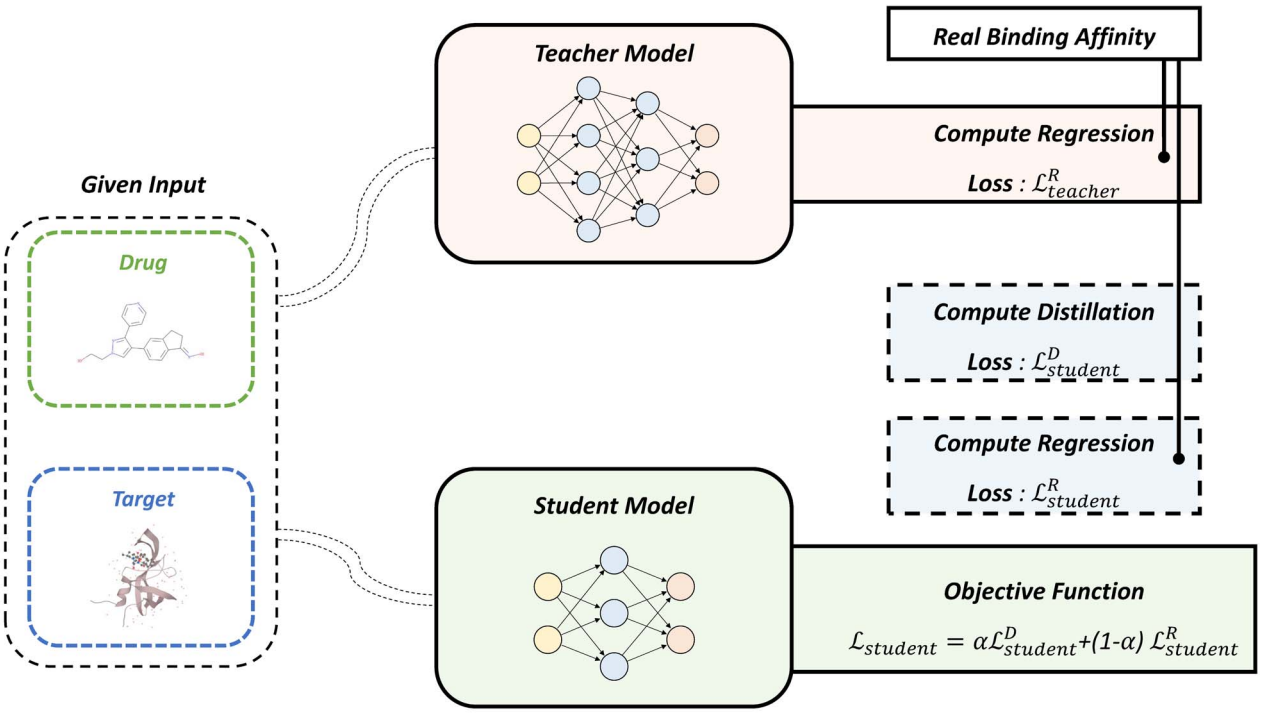

Figure 6. The training stage of the teacher and the student models. We divide the loss function into three parts: (A) $\mathcal{L}^R_{teacher}$ for teacher model. (B) Distillation loss$\mathcal{L}^D_{student}$ and regression loss $\mathcal{L}^R_{student}$. (C) $\mathcal{L}_{student}$ for student model, which is the weighted sum of $\mathcal{L}^D_{student}$ and $\mathcal{L}^R_{student}$.

layer with parameters θ_r, regression function r is to map $g(x, \theta_{Guide})$ to a vector space with the same dimension as $g(x, \theta_{Hint})$. Specifically, we define the vectors generated by the first fully connected layer after the multi-head linear attention layer as the output of function g.

L_2 is defined as mean square error (MSE):

$$\mathcal{L}_2(Y_i, P_i) = MSE(Y_i, P_i) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - P_i)^2, \tag{22}$$

where Y_i is the true value of binding affinity and P_i is the predicted value. From the perspective of the loss function, the update strategy of the student model is to ensure that hidden layer outputs are as close as possible to the teacher model. Therefore, the teacher model can provide student with guidance during the training process. Meanwhile, the student's parameters after hidden layer are not affected by the teacher, so the flexibility of the network will not be greatly inhibited. From a biological point of view, the binding information between protein and drug is not only expressed through binding affinity [34]. Assuming that the feature map of the middle layer contains more hidden information, e.g. the location of binding site, the student model can learn transferable biological structure information, rather than just bringing the feature map closer to the verified better parameters.

In addition, knowledge distillation contributes to the restrain of model parameters and overfitting. Considering the feature map of teacher model as a constraint, knowledge distillation limits the difference between the parameters of teacher and student. Compared with L2-normalization, loss function L_1 allows model to learn more effective verified network parameters (rather than simply zero).

Evaluation metrics

Concordance index (CI), a model evaluation index proposed by GÖnen and Heller [35], was designed to calculate the difference between the predicted value of the model and the ground truth. CI is defined as follows:

$$CI = \frac{1}{Z}\sum_{\delta_j > \delta_i} h(b_i - b_j), \tag{23}$$

where b_i is the prediction value for δ_i, b_j is the prediction value for δ_j, $h(x)$ is the step function and Z is the normalized hyperparameter. Commonly, the step function $h(x)$ is defined as follows:

$$h(x) = \begin{cases} 0, & x < 0 \\ 0.5, & x = 0 \\ 1, & x > 0 \end{cases}. \tag{24}$$

MSE is a statistical measure that evaluates the error directly. Assuming there are estimated n sample and corresponding true values of n sample, MSE is expressed as the expectation of the square loss:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{25}$$

Table 1. The performance of FusionDTA and baseline models on the Davis dataset

Model	CI (std)	MSE	r_m^2 (std)
KronRLS	0.871 (±0.001)	0.379	0.407 (±0.005)
SimBoost	0.872 (±0.002)	0.282	0.644 (±0.006)
DeepDTA	0.878 (±0.004)	0.261	0.630 (±0.017)
WideDTA	0.886 (±0.003)	0.262	–
MT-DTI	0.887 (±0.003)	0.245	0.665 (±0.014)
DeepCDA	0.891 (±0.003)	0.248	0.649 (±0.009)
MATT_DTI	0.891 (±0.002)	0.227	0.683 (±0.017)
GraphDTA	0.893 (±0.001)	0.229	–
FusionDTA	0.913 (±0.002)	0.208	0.743 (±0.007)

Table 2. The performance of FusionDTA and baseline models on the KIBA dataset

Model	CI (std)	MSE	r_m^2 (std)
KronRLS	0.782 (±0.001)	0.441	0.342 (±0.001)
SimBoost	0.836 (±0.001)	0.222	0.629 (±0.007)
DeepDTA	0.863 (±0.002)	0.194	0.673 (±0.009)
WideDTA	0.875 (±0.001)	0.179	–
MT-DTI	0.882 (±0.001)	0.152	0.738 (±0.006)
DeepCDA	0.889 (±0.002)	0.176	0.682 (±0.008)
MATT_DTI	0.889 (±0.001)	0.150	0.756 (±0.011)
GraphDTA	0.891 (±0.002)	0.139	–
FusionDTA	0.906 (±0.001)	0.130	0.793 (±0.008)

where $\hat{y}_i$ is the estimate of i_{th} sample and y_i is the true value of i_{th} sample.

Regression toward the mean (r_m^2 index) is a measure evaluating the external predictive performance of a model. If a variable is very large, then r_m^2 means how much it tends to approach the average next time. r_m^2 index is calculated as follows:

$$r_m^2 = r^2 * (1 - \sqrt{(r^2 - r_0^2)}), \tag{26}$$

where r is the squared correlation coefficients with intercepts and r_0 is the coefficients without intercepts.

Result and discussion

In this section, the Davis data set and KIBA data set were utilized to evaluate the performance of the model. In FusionDTA, the hyperparameters used in these two data sets are shown in Supplementary Tables S1 and S2. To evaluate the performance of the multi-head linear competition, we compared it with the largest pool and the average pool in the Davis data set. In addition, a comparative experiment was set up in the experiment, in which the performance of knowledge distillation is measured with existing models and vanilla FusionDTA. We compared our model with the following benchmark models: KronRLS [7], SimBoost [8], DeepDTA [9], WideDTA [36], GraphDTA [11], DeepCDA [10], MT-DTI [19] and MATT_DTI [16].

The performance of FusionDTA

In Table 1, we listed the performance of the proposed model evaluated on the Davis dataset and compared it with the baseline model. As shown, FusionDTA is superior to the existing models in all aspects. In detail, FusionDTA improves CI index by 0.020 and reduced MSE by 0.021, compared with previous the baseline model, GraphDTA. In addition, FusionDTA also achieves a 0.060 improvement in r_m^2 index compared to the baseline model, MATT_DTI. In particular, for some previous models that have proposed more than two kinds of model architecture, we only report the best performing architecture on the Davis dataset.

Table 2 presents the performance of FusionDTA and baseline models on the KIBA dataset. The results show that FusionDTA also achieves significantly better results than baseline models in all of the evaluation measures. FusionDTA improves 0.015, 0.009 in CI index and MSE, compared with previous the state-of-art model, GraphDTA. Moreover, FusionDTA also achieves a 0.037 improvement in r_m^2 index over MATT_DTI.

Figure 7 illustrates the real affinity against the predicted value on both Davis and KIBA datasets. Assuming ground truth as x-axis and prediction as y-axis, the vertical distance $|\Delta y|$ from each point to $y = x$ represents the discrepancy between its predicted affinity value and the real value. Histograms at the edges represent the overall distribution of true and predicted affinity. As shown, the samples have a tendency to be symmetric about $y = x$ for both the Davis and Kiba datasets. Especially,

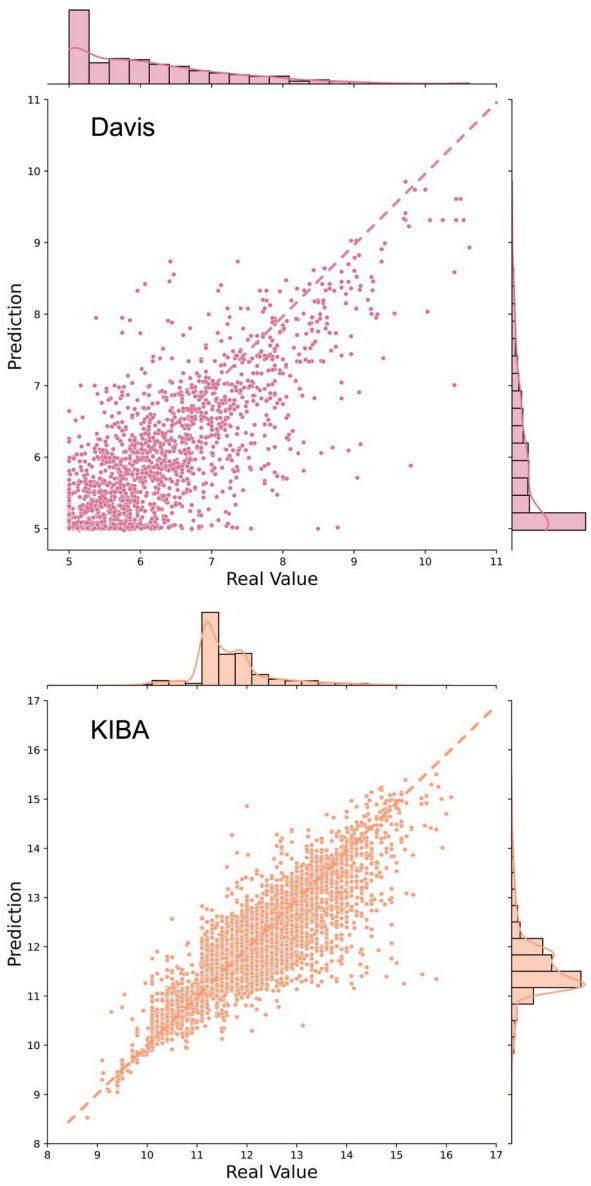

Figure 7. The real affinity against the predicted value on Davis and KIBA datasets. For each point, x-axis reflects its real value and y-axis reflects its predicted value. The vertical distance |Δy| from each sample to y=x represents the discrepancy between its predicted affinity value and the real value.

the sampling points in Kiba dataset are more densely distributed around $y = x$.

The performance of various pooling methods

In the model architecture, different pooling methods allow the model to pay attention to different parts of the intermediate sequence, determining the parameters of each layer to be updated according to different gradients. The common pooling methods include max-pooling and mean-pooling, which respectively aggregate the features map of a sequence into a token with a maximizing function or an averaging function. In this model, multi head linear attention layer is proposed to replace the traditional pooling layer, with an end to selectively focus on the information of each biological token on the whole protein sequence, or an entire SMILES chain.

Table 3. The performance of max-pooling, mean-pooling and multi head linear attention layer on the Davis dataset

Pooling method	CI	MSE
Max-pooling	0.904	0.220
Mean-pooling	0.910	0.211
Multi-head linear attention	0.913	0.208

Table 4. The performance of max-pooling, mean-pooling and multi head linear attention layer on the KIBA dataset

Pooling method	CI	MSE
Max-pooling	0.897	0.137
Mean-pooling	0.904	0.132
Multi-head linear attention	0.906	0.130

To evaluate the impact of different pooling methods on model performance, three controlled experiments were set up in the verification stage. It is worth mentioning that the model parameters of each experiment group are the same except for different pooling methods. In Table 3, we list the performance of max-pooling, mean-pooling and multi-head linear attention layer on the Davis dataset. As it is shown, CI index of multi-head linear attention layer is 0.913, whereas the CI index of max-pooling and mean-pooling is 0.904 and 0.910, respectively. Obviously, multi-head linear attention layer performs better than the other two pooling methods on Davis dataset.

Table 4 reports the performance of max-pooling, mean-pooling and multi-head linear attention layer on the KIBA dataset. As shown, CI index of multi-head linear attention is 0.906, which is higher than 0.897 of max-pooling and 0.904 of mean-pooling. In addition, the MSE multi-head linear attention layer is 0.130, lower than 0.137 of max-pooling and 0.132 of mean-pooling. For KIBA dataset, multi-head linear attention layer as feature aggregator performs better than mean-pooling and max-pooling.

The performance of cold-start

The cold-start problem refers to evaluate model performance on the unseen inputs. From an application point of view, a high proportion of protein or drug representations may not appear in the training set. Therefore, the challenge is whether a model with an excellent score in specific datasets can also perform well with unknown data. In this regard, the performance of cold-start indicates the model's robustness facing a new environment (e.g. mutate proteins).

We compare our model with the following benchmark models: GraphDTA [11], GLFA and GEFA [17]. Table 5 reports the performance of drug cold-start, protein cold-start and drug-protein cold-start on Davis dataset, corresponding to the unseen drug, unseen protein, and unseen

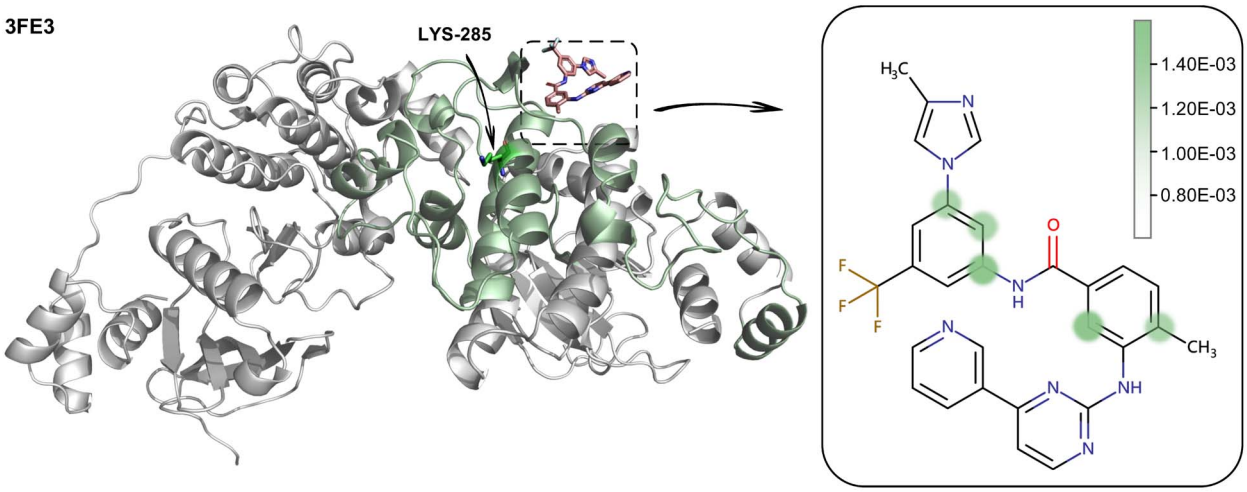

Figure 8. The example of the weight visualization of the proposed model. MARK3 (PDB ID: 3FE3) is expressed in cartoon form, while Nilotinib is expressed in stick form. The cyan color highlights the highly focused position of the protein and the focused drug atom in the binding bag, and the darker color indicates the darker attention weight.

Table 5. The performance of drug cold-start, protein cold-start, and drug–protein cold-start on Davis dataset

	CI	MSE
Drug cold-start		
GraphDTA	0.675	0.920
GLFA	0.670	0.861
GEFA	0.709	0.846
FusionDTA	0.747	0.681
Target cold-start		
GraphDTA	0.706	0.510
GLFA	0.780	0.4531
GEFA	0.795	0.4335
FusionDTA	0.826	0.331
Drug-target cold-start		
GraphDTA	0.627	1.130
GLFA	0.636	1.144
GEFA	0.639	0.989
FusionDTA	0.685	0.716

Table 6. The performance of vanilla FusionDTA, KD + FusionDTA (large-scale) and KD + FusionDTA (small-scale) on Davis dataset

	CI	MSE
Vanilla FusionDTA (large-scale)	0.913	0.208
Vanilla FusionDTA (small-scale)	0.905	0.221
KD + FusionDTA (large-scale)	0.914	0.205
KD + FusionDTA (small-scale)	0.908	0.213

drug and protein. As shown, FusionDTA gained 0.747 CI index and 0.681 MSE under drug cold-start constrain, 0.826 CI index and 0.331 MSE under protein cold-start constrain, 0.685 CI index and 0.716 MSE under protein–drug cold-start constrain. As a result, our model is better than all of the baseline models on the cold-start problem and more likely to perform robustly in undiscovered applications.

The performance of knowledge distillation

In this section, we evaluated the contribution of knowledge distillation on Davis dataset. As is mentioned above, knowledge distillation is an effective way to facilitate knowledge transfer and parameter regularization. Two experiments with different parameters, therefore, were set up in the verification stage to examine the various effects of knowledge distillation. In one experiment, the parameter size of the student model was exactly the same as that of the teacher model, aiming to evaluate the effect of teacher guidance on students' model performance. In the other experiment, a student model with parameters of only half the size was used to evaluate the capability of model compression. For each experiment, the teacher model was set up with frozen pre-trained FusionDTA, while the student model was initialized with untrained FusionDTA. Then, the student model would learn new distribution from teacher's output and true value, by the training strategy of knowledge distillation.

Table 6 shows the performance of FusionDTA (large-scale), knowledge distillation + FusionDTA (large-scale) and knowledge distillation + FusionDTA (small-scale) evaluated on the Davis dataset. As shown, knowledge distillation + FusionDTA (large-scale) improves CI index by 0.001 and reduced MSE by 0.003 compared with FusionDTA (large-scale). Knowledge distillation with FusionDTA of small scale also achieved the CI index of 0.908.

Supplementary Figure S1 shows the performance of the baseline models and the proposed models measured by of CI, MSE and model scale. The number of parameters of DeepDTA, DeepCDA, GraphDTA, FusionDTA(large-scale), KD+FusionDTA(large-scale) and KD+FusionDTA(small-scale) is 1 967 745, 3 641 345, 4 749 573, 5 362 081, 5 362 081 and 2 013 537. As shown, knowledge distillation + FusionDTA (large-scale) achieves the best performance around all the methods.

Meanwhile, with a little loss of accuracy, knowledge distillation can be regarded as an effective model compression method for DTA task.

Visualization with attention weights

The attention weights obtained by FusionDTA can be used to analyze which part of the interaction between the small drug molecule and the target protein plays a key role in binding pocket. The attention mechanism can calculate some key areas of interaction between protein sequence and drug compounds. In order to visualize the main areas of interaction, we first calculated the weights of the protein sequence and the SMILES characters of the drug compound and then selected the corresponding interaction site with a relatively large attention value. Figure 8 shows an example of the weight visualization of the proposed model. We chose the complex of MARK3 (PDB ID: 3FE3) and Nilotinib for interactive visual analysis. The results showed that the weight value is mainly from 5.69E-4 to 1.43E-3. We colored the positions where the attention weight is greater than 9.80E-4 in the drug compound and the attention weight is greater than 9.57E-4 in the protein. The cyan color highlights the highly focused position of the protein and the focused drug atom in the binding bag, and the darker color indicates the smaller attention weight. Obviously, our model mainly captured the main amino acid regions, residues 194–339. Interestingly, the attention weight captured by our model in residues 194–339 is almost close to 9.57E-4, and residues 285–287 are relatively larger. The peak value is at LYS-285, which just falls in the binding pocket, indicating that our model accurately predicted the potential docking site measurement. Overall, some of the residues 194–339 are in the docking pocket of MARK3 and Nilotinib, while some are located outside the region, which also indicates that most of the regions captured by our model are located at the docking interface, but part of them captures the wrong area. The weights calculated by our model are mainly concentrated in the binding pocket, which shows that our model can predict the interaction between protein and compound more accurately. In short, the proposed model can extract useful information from the two channels of drug SMILES and protein sequence.

Conclusion

This paper has presented a novel DTA prediction framework, FusionDTA. A new multi-head linear attention mechanism is applied to replace the coarse pooling method, which uses attention weights to aggregate global information. Additionally, we applied knowledge distillation in the framework, by transferring the learnable information from the teacher model to the student model. In order to evaluate the proposed work, it was applied to two common data sets: KIBA, Davis. Experimental results show that our model performs better than existing models on all evaluation indicators. When drugs and proteins are unknown, FusionDTA proved to be more robust and more effective than other models in the benchmark, which will help in the development of some new drugs. Meanwhile, knowledge distillation is of great help to performance improvement, saving half of the parameters of the model while the CI index hardly changes. More importantly, in this case, our model can easily exceed the baseline of all indicators. In additional experiments concluded in Supplementary Tables S3 and S4, the pre-training representation of protein proves to be effective for DTA model with sequential inputs, while pre-training of drug fails to show the superiority for DTA task. Furthermore, the model has been shown to provide biological insights for understanding the nature of molecular interactions and capture the binding pockets of proteins and molecules. In general, our model has superior performance and improves the effect of DTI prediction. The visualization of DTI can effectively help predict the binding region of proteins during structure-based drug design.

Key Points

- Due to the maximize or average operator, crude use of pooling method may result in the loss of hidden information. To this end, we propose a multi-head linear attention to capture the deep dependency from each token.
- To solve the redundancy issue of parameters, we propose a constraint where small scale network can learn knowledge from large scale network and real affinity.
- Our method achieves the state-of-the-art performance in Davis dataset and KIBA dataset. FusionDTA with half the parameters can easily exceed the baseline on all metrics in terms of model compression.

Supplementary Data

Supplementary data are available online at http://bib.oxfordjournals.org/.

Code and data availability

The source code and data of this study are available at https://github.com/yuanweining/FusionDTA.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62176272), Guangzhou Science and Technology Fund (Grant No. 201803010072), Science, Technology and Innovation Commission of Shenzhen Municipality (JCYL 20170818165305521) and

China Medical University Hospital (DMR-111-102, DMR-111-143, DMR-111-123). We also acknowledge the start-up funding from SYSU's "Hundred Talent Program".

References

1. Newman DJ, Cragg GM. Natural products as sources of new drugs over the nearly four decades from 01/1981 to 09/2019. *J Nat Prod* 2020;**83**(3):770–803.
2. Takebe T, Imai R, Ono S. The current status of drug discovery and development as originated in United States academia: the influence of industrial and academic collaboration on drug discovery and development. *Clin Transl Sci* 2018;**11**(6):597–606.
3. Lin X, Li X, Lin X. A review on applications of computational methods in drug screening and design. *Molecules* 2020;**25**(6):1375.
4. Wen M, Zhang Z, Niu S, *et al.* Deep-learning-based drug-target interaction prediction. *J Proteome Res* 2017;**16**(4):1401–9.
5. Kairys V, Baranauskiene L, Kazlauskiene M, *et al.* Binding affinity in drug design: experimental and computational techniques. *Expert Opin Drug Discovery* 2019;**14**(8):755–68.
6. Yadav AR, Mohite SK. Homology modeling and generation of 3d-structure of protein. *Res J Pharm Dosage Forms Technol* 2020;**12**(4):313–20.
7. Pahikkala T, Airola A, Pietilä S, *et al.* Toward more realistic drug-target interaction predictions. *Brief Bioinform* 2015;**16**(2):325–37.
8. He T, Heidemeyer M, Ban F, *et al.* Simboost: a read-across approach for predicting drug-target binding affinities using gradient boosting machines. *J Chem* 2017;**9**(1):1–14.
9. Öztürk H, Özgür A, Ozkirimli E. Deepdta: deep drug-target binding affinity prediction. *Bioinformatics* 2018;**34**(17):i821–9.
10. Abbasi K, Razzaghi P, Poso A, *et al.* Deepcda: deep cross-domain compound-protein affinity prediction through lstm and convolutional neural networks. *Bioinformatics* 2020;**36**(17):4633–42.
11. Nguyen T, Le H, Quinn TP, *et al.* Graphdta: predicting drug–target binding affinity with graph neural networks. *Bioinformatics* 2021a;**37**(8):1140–7.
12. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. 2016.
13. Veličković P, Cucurull G, Casanova A, *et al.* Graph attention networks. arXiv preprint arXiv:1710.10903. 2017.
14. Xu K, Hu W, Leskovec J, *et al.* How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826. 2018.
15. Zheng S, Li Y, Chen S, *et al.* Predicting drug-protein interaction using quasi-visual question answering system. *Nat Mach Intell* 2020;**2**(2):134–40.
16. Zeng Y, Chen X, Luo Y, *et al.* Deep drug-target binding affinity prediction with multiple attention blocks. *Brief Bioinform* 2021;**22**(5):1–10.
17. Nguyen TM, Nguyen T, Le TM, *et al.* Gefa: early fusion approach in drug-target affinity prediction. *IEEE/ACM Trans Comput Biol Bioinform* 2021b.
18. Chen L, Tan X, Wang D, *et al.* TransformerCPI: improving compound-protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics* 2020;**36**(16):4406–14.
19. Shin B, Park S, Kang K, *et al.* Self-attention based molecule representation for predicting drug-target interaction. arXiv preprint arXiv:1908.06760, 2019.
20. Asgari E, Mofrad MRK. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS One* 2015;**10**(11):e0141287.
21. Rao R, Bhattacharya N, Thomas N, *et al.* (eds). Evaluating protein transfer learning with tape. In: *Advances in Neural Information Processing Systems*. Vancouver, Canada: Neural Information Processing Systems Foundation, Inc., Vol. **32**, 2019, 9689.
22. Rives A, Meier J, Sercu T, *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci* 2021;**118**(15):e2016239118.
23. Hirohara M, Saito Y, Koda Y, *et al.* Convolutional neural network based on smiles representation of compounds for detecting chemical motif. *BMC Bioinform* 2018;**19**(19):83–94.
24. Jiang M, Li Z, Zhang S, *et al.* Drug-target affinity prediction using graph neural network and contact maps. *RSC Adv* 2020;**10**(35):20701–12.
25. Buciluǎ C, Caruana R, Niculescu-Mizil A. Model compression. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, Pennsylvania, USA: ACM, Inc., 2006, 535–41.
26. Davis MI, Hunt JP, Herrgard S, *et al.* Comprehensive analysis of kinase inhibitor selectivity. *Nat Biotechnol* 2011;**29**(11): 1046–51.
27. Tang J, Szwajda A, Shakyawar S, *et al.* Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *J Chem Inf Model* 2014;**54**(3):735–43.
28. Weininger D. Smiles: a chemical language and information system. *J Chem Inf Comput Sci* 1988;**28**(1):31–6.
29. Qiu X, Sun T, Yige X, *et al.* Pre-trained models for natural language processing: a survey. *Sci China Technol Sci* 2020;**63**(10): 1872–97.
30. Sundermeyer M, Schlüter R, Ney H. Lstm neural networks for language modeling. In: *Thirteenth Annual Conference of the International Speech Communication Association*. Portland, Oregon, USA: International Speech Communication Association (ISCA), 2012.
31. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. 2015.
32. Clark K, Luong M-T, Khandelwal U, *et al.* Bam! born-again multi-task networks for natural language understanding. arXiv preprint arXiv:1907.04829. 2019.
33. Hinton GE, Salakhutdinov RR. Replicated softmax: an undirected topic model. *Adv Neural Inform Process Syst* 2009;**22**: 1607–14.
34. Vuignier K, Schappler J, Veuthey J-L, *et al.* Drug-protein binding: a critical review of analytical tools. *Anal Bioanal Chem* 2010;**398**(1):53–66.
35. GÖnen M, Heller G. Concordance probability and discriminatory power in proportional hazards regression. *Biometrika* 2005;**92**(4):965–70.
36. Öztürk H, Ozkirimli E, Özgür A. Widedta: prediction of drug-target binding affinity. arXiv preprint arXiv:1902.04166. 2019.