MAC: An Interpretable and Self-Attention Based Graphical Convolutional Neural Network in DTI Prediction

Chun Chi 518021910298
Shengyuan Hou 518021910604
Yifan Liu 518021910609

Abstract. Drug Target Interaction (DTI) prediction now has attracted a large number of scholars to dig in because of its potential practical values including shortening drug development cycle, exploring new applications of existing drugs and providing personalized drugs recommendation for patients. The existing DTI research methods include Docking method, Similarity Based method and Deep Learning. In this paper, our research group raised a new frame called MAC, which combines Message Passing Neural Network(MPNN) and CNN respectively for the encoding of Drugs and Proteins. MAC has outstanding ability to learn and extract the spatial features of drug molecules and excellent ability to accurately locate proteins' action sites because of MPNN and attention mechanism so that MAC achieved Accuracy=0.99 and F1=0.98 on the first 100,000 pairs from dataset provided by TA. Subsequent control variate experiments, robustness test and visualization verified the structural effectiveness, strong robustness and great interpretability of MAC. Programming code of our projected has been proposed on github website[1].

Keywords: MPNN, CNN, Spatial Feature, Attention Mechanism

1 Introduction

1.1 Background and Motivation

Traditional drug discovery process is often accompanied with huge amounts of validation experiments which are time-consuming and highly cost. To develop a new drug and make it accessible to the market, at least three phases of clinical trial has to be executed, where decades of years have to be taken. Unfortunately, modern diseases are showing a trend of rapid growth as viruses is mutating more and more faster. Those imbalances motivate pharmacist to explore more efficient new drug discovery processes in recent years. One essential preliminary part is to sift out all alternative chemical compounds which might well interact or combine with target protein. In the past it relies on countless hand-crafted experiments. Based on the fact that similar compounds are often interacted with similar target proteins, we could perform drug-target interaction(DTI) prediction from existing interactive-pairs. Three categories of methods have been applied to DTI problem, including docking method, similarity-based method, text processing based method and deep neural network.

Docking method[8] It utilizes the extracted 3D structure of protein and molecular. Each time 3D structures of a drug D and a target T are prepared and drug D is docked into the binding pocket of target T. If the drug D binds well within target T, T is predicted as a potential target of D; otherwise,

[1] https://github.com/houshengyuan/EI314-Drug_Target_Interaction

T is not considered as a potential target of D(as in Fig 1). By visualization interpretability of the result is enhanced consequently. However, for large scale dataset, it is difficult to extract 3D structure efficiently, which limits its scalability. Otherwise, it consumes quantities of storage space.

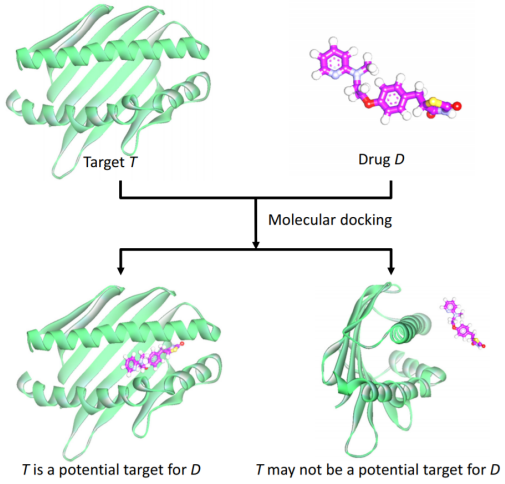


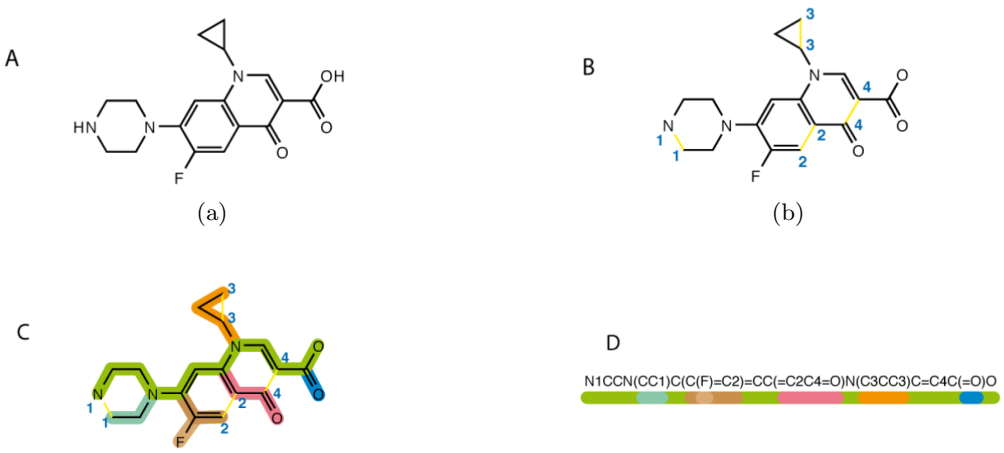
Fig. 1: Docking method

Similarity-based method[3] It borrows the idea and algorithm of Recommender System that taking drugs as the items while targets as the users. The relation between a drug and a target can be calculated by the normalized sum of the similarity between the given drug and the target's all positive drugs. In short, for a certain target, we can predict the possible drug target interaction by searching for a drug similar to the known effective drugs to this target. Unfortunately, we have found that it doesn't perform well in unseen drugs or unseen targets and demands for huge computation time and space.

Text processing based method[7] It applies some methods in natural language processing or text processing to drugs and proteins and get the embeddings of them. Text processing based method will consider drugs and proteins as text sequences. After that, it tries to discover the relationship between different location or the frequencies of different "words" (like atoms) and phrases (like functional groups) and encode drugs and proteins based on discovered knowledge. This method is just like extracting some critical features of drugs and proteins. After encode both drugs and proteins, it uses some deep learning methods like convolutional neural network or transformer to predict DTI.

1.2 Preliminaries of SMILES and FASTA

SMILES[14] is a string obtained by printing the symbol nodes encountered in a depth-first tree traversal of a chemical graph. The chemical graph is first trimmed to remove hydrogen atoms and cycles are broken to turn it into a spanning tree. Where cycles have been broken, numeric suffix labels are included to indicate the connected nodes. Parentheses are used to indicate points of branching on the tree, as shown in Fig 2.



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

Fig. 2: SMILES Generation Algorithm

In bioinformatics and biochemistry, FASTA[13] is a text-based format for representing either nucleotide sequences or amino acid (protein) sequences by using single-letter codes, which can also contain gaps or alignment characters. The format also allows for sequence names and comments to precede the sequences. The format is originated from FASTA software package, but has now become a near universal standard in the field of bioinformatics. The simplicity of FASTA makes it easy to manipulate and parse using text-processing tools and scripting languages like Python. The amino acid consists of 22 amino acids and 3 special codes.

Binding Affinity represents interaction magnitude between drug and target sequence pair. For simplicity, we omit concrete values of binding affinity by binarizing all the binding labels where 1 represents drug-target pair has an interaction and 0 represents no interaction yet inspected, which means we are undergoing a classification rather a regression problem.

1.3 Contributions

In this project, our contributions could be summarized as follows:

1. We survey for different DTI methods in detail and reproduce three typical methods among them: MolTrans, DeepDTA and DeepCDA and analyzed the reason for their bottlenecks.
2. We propose our DTI method: MPNN-Attentioned CNN (MAC) based on MPNN frame, CNN structure and attention mechanism, with clear idea, easy implementation and high interpretability.
3. We conduct various experiments on MolTrans, DeepCDA, DeepDTA and our MAC model. We also compare peroformance among them and conduct robustness test.
4. We visualize our MAC results based on attention layer's weights to illustrate the potential interaction sites of drug and protein pairs.
5. We verify the effectiveness of MPNN's strong ability to extract spatial features and attention mechanism's improvement for DTI predcition.

2 Related work

2.1 Discovery Studio

Discovery Studio ™ (DS) is a new molecular modeling environment on PCs and a professional molecular simulation software for life sciences whose main applications include protein characterization (including protein-protein interaction), homology modeling, molecular mechanics calculation and molecular dynamics simulation, structure-based drug design tools (including ligand protein interaction, new drug design and molecular docking), small molecule based drug design tools (including quantitative structure-activity relationship, pharmacophore, drug delivery system, drug delivery system, drug delivery system, etc.) The design and analysis of database screening, ADMET and composite library. In this paper, our research group mainly uses DS's Discovery Studio Standalone, DS MODELER and DS Sequence Analysis modules.

2.2 Similarity search-based methods[11]

In this part, we use the Collaborative Filtering method in the Recommendation System to find the possible positive drug-target pairs based on drug-drug's and target-target's similarity.

The core idea of this method is that the more similar two drugs are then the more possible that the both drug can effect one target.

Based on the above mentioned core idea, we can divide the whole algorithm into the following steps:

First, create a Drug-Drug-Similarity Matrix (DDS Matrix). Assuming that there are n non-duplicate drugs in the given dataset, then the generated DDS has the size of n×n while DDS [i][j] represents the similarity of fingerprints of drug i and drug j. Similarity calculation is achieved by directly calling rdkit.DataStructs.FingerprintSimilarity function. While calculating the DDS matrix, a by-product is also generated: a Drug-Index array with the size of n × 2. The first column stores the drug's molecular formula, and the second column stores the drug's ID. By querying this index, IDs can be used to identify different drugs, reducing the spatial complexity of subsequent calculations.[10]

Second, similarly, the directory Target-Index between targets is also generated to distinguish different targets by their IDs.

Third and the most importantly, create the Target-Drug-Interaction Matrix(TDI Matrix). Through this matrix, we can get the similarity between one target and all drugs. If target i and drug j has known positive DTI relation in the train dataset, then TDI[i][j] equals 1. After recording all positive DTI pairs, then regularizing every row of the TDI matrix.

Finally, the prediction result can be generated. Given target i and drug j, the total similarity equals the sum of target i and drug k's TDI and drug k and drug j's drug-drug-similarity. Then if the similarity overcome a given threshold, usually 0.5, the the prediction result is 1.

2.3 Text Processing Based method - MolTrans

In this part we introduce an efficient text processing based method: MolTrans [7]. It combines text processing and transformer [12] to solve DTI problems. Figure 3 illustrates the whole pipeline.

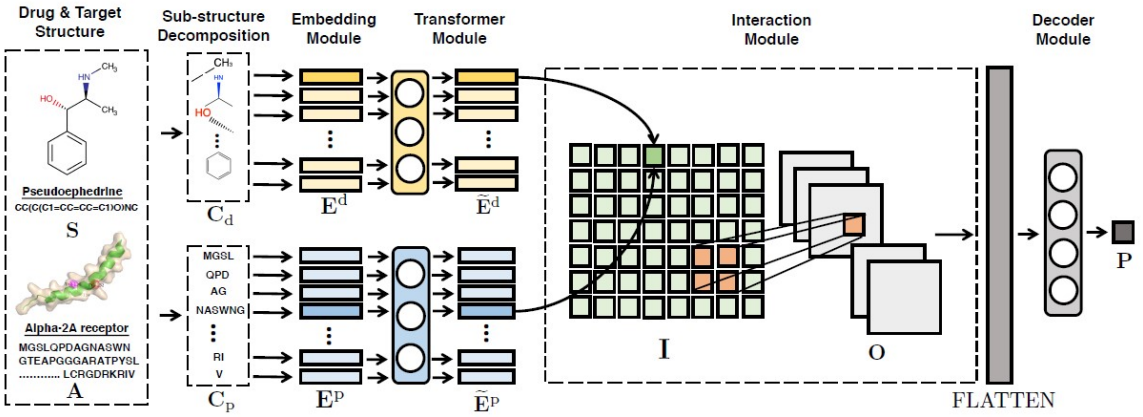


Fig. 3: Illustration of MolTrans

Drugs and Proteins Encoder MolTrans propose a data-driven sequential pattern mining algorithm Frequent Consecutive Sub-sequence Algorithm (FCS) to find recurring sub-sequences. The algorithm is shown in Algorithm 1. It iteratively updates token vocabulary along with the input of drugs and proteins. At the same time, drugs and proteins will be encoded with the help of the vocabulary. FCS aims to generate a set of hierarchy of frequent sub-sequences for sequences. In this way, MolTrans could convert drugs and proteins into a new format representing important features and substructures as well as their locations. Since it is a text processing based method, it needs some prior knowledge. FCS needs initial amino acids/SMILES tokens and we download them from github[2]. After performing FCS, we could get two new encoding matrix $\mathbf{C}_{\mathrm{d}}$ and $\mathbf{C}_{\mathrm{p}}$ containing sub-structure information. each column $\mathbf{C}_i^{\mathrm{d}}$ and $\mathbf{C}_i^{\mathrm{p}}$ is a one-hot vector corresponding to the sub-structure index for the i-th sub-structure of drug sequence and protein sequence.

Algorithm 1 Frequent Consecutive Sub-sequence Mining

Input: The set of all initial amino acids/SMILES tokens $\mathbb{V}$; The set of tokenized proteins/drugs $\mathbb{W}$; The specified frequency threshold θ; the maximum size of $\mathbb{V}$ l.

Output: The updated tokenized proteins/drugs $\mathbb{W}$; The updated token vocabulary set $\mathbb{V}$.

for $t = 1, \cdots, l$ **do**
  (A,B), FREQ $\leftarrow$ scan $\mathbb{W}$
  if FREQ $< \theta$ **then**
    break
  end if
  $\mathbb{W} \leftarrow$ find (A,B)$\in \mathbb{W}$, replace with (AB)
  $\mathbb{V} \leftarrow \mathbb{V} \cup$ (AB)
end for

Transformer Embedding After getting encoded drug and protein representations $\mathbf{C}_{\mathrm{d}}$ and $\mathbf{C}_{\mathrm{p}}$, we further generate content embedding and positional embedding. Let's take encoded drug substructure matrix $\mathbf{C}_{\mathrm{d}}$ as an example, and the procedure of further encoding $\mathbf{C}_{\mathrm{p}}$ is similar.

[2] https://github.com/kexinhuang12345/MolTrans/tree/master/ESPF

First of all, we generate content embedding based on a lookup dictionary matrix $\mathbf{W}^{\mathrm{d}}_{\mathrm{cont}}$ which contains latent embedding for all possible sub-structures as follows:

$$\mathbf{E}^{\mathrm{d}}_{\mathrm{cont}\ i} = \mathbf{W}^{\mathrm{d}}_{\mathrm{cont}}\ \mathbf{C}^{\mathrm{d}}_{i}$$

Second, we generate positional embedding $\mathbf{E}^{\mathrm{d}}_{\mathrm{pos}\ i}$ indicating which sub-structure appears in the i-th position. The final embedding $\mathbf{E}^{\mathrm{d}}_{i}$ are generated via the sum of content and positional embedding:

$$\mathbf{E}^{\mathrm{d}}_{i} = \mathbf{E}^{\mathrm{d}}_{\mathrm{cont}i} + \mathrm{E}^{\mathrm{d}}_{\mathrm{pos}_i}$$

The embedding of whole drug $\mathbf{E}^{\mathrm{d}}$ is just the combination of the embedding of each column $\mathbf{E}^{\mathrm{d}}_{i}$. We use transformer layers to further encode $\mathbf{E}^{\mathrm{d}}$:

$$\widetilde{\mathbf{E}}^{\mathrm{d}} = \mathrm{Transformer}_{\mathrm{Drug}}\ \left(\mathbf{E}^{\mathrm{d}}\right)$$

CNN for DTI Prediction We use $\mathbf{I}_{i,j}$ to model the pair-wise interaction of sub-sequence i in drug and sub-sequence j in protein and add a CNN layer to model the interaction of nearby regions:

$$\mathbf{I}_{i,j} = \mathrm{F}\left(\widetilde{\mathbf{E}}^{\mathrm{d}}_{i}, \widetilde{\mathbf{E}}^{\mathrm{p}}_{j}\right)$$

$$\mathbf{O} = \mathrm{CNN}(\mathbf{I})$$

$$\mathbf{P} = \sigma\left(\mathbf{W}_{\mathrm{o}}\mathrm{FLATTEN}(\mathbf{O}) + \mathbf{b}_{\mathrm{o}}\right)$$

$$l = \mathbf{Y}\log\mathbf{P} + (1-\mathbf{Y})\log(1-\mathbf{P})$$

Where F is a function that measures the interaction (it uses dot product actually), $\sigma(\mathbf{x}) = 1/(1+\exp(-\mathbf{x}))$, and $\mathbf{Y}$ is the label.

2.4 Deep neural network

DeepDTA: deep drug-target binding affinity prediction[9] As the first attempt to incorporate deep learning techniques into drug-target binding affinity prediction without any handcrafted feature engineering preprocessing, DeepDTA model tries to employ CNN blocks to learn representations from the raw protein sequences and SMILES strings and combine these representations to feed into a fully connected layer block, as shown in Fig 4.

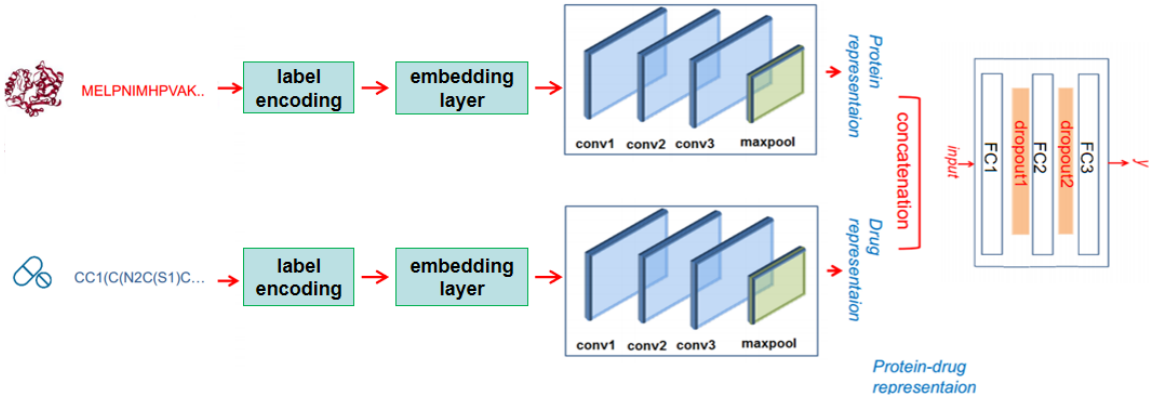


Fig. 4: DeepDTA Model Overview

Firstly, it represent SMILES(FASTA) sequence as a list of integer labels from a mapping dictionary which corresponds every chemical sign(amino acid) into an integer. For simplicity padding ,we define integer 0 as padding sign $< PAD >$. Next, each integer sequence is projected by an embedding layer which is automatically learned during training. This is much better than simple one-hot encoding since it could capture the similarity between different atoms, bonds or amino acids while the latter one directly treat them as independent. Then both drugs and targets representation matrix are fed into three one-dimensional convolutional blocks with different kernel sizes followed by a pooling layer downsampling results of extracted convolutional feature. Downsampling essentially enables kernel to not only detect interaction feature from a larger view point but also intergrates low-level feature into high-lavel ones including local dependencies. The longest interaction distance CNN can capture is determined by the size of filter kernels, making it crucial. Now both SMILES and FASTA sequence are encoded into a feature vector with 96 dimensions.

To combine the represented feature vector, the model directly concatenates them as 192-dim global representation and feed them into a multi-layer perceptron(MLP). The MLP is comprised of three linear connected layers each followed by a ReLU activate function. To alleviate overfitting two dropout layers are inserted into the gap of fully connected layers and ultimately we add an activation layer of sigmoid to transform the output into probability.

DeepCDA: deep cross-domain compound–protein affinity prediction through LSTM and convolutional neural networks[1] In DTI tasks, one common challenge is that there are always some unseen proteins or unseen proteins existing in the validation dataset resulting to the degradation of performance. To solve this, DeepCDA raised the corresponding resolution that using Adversarial Domain Adaptation technique to make training domain and test domain more similar.

DeepCDA model can be roughly divided into two parts: (1) Feature Encoder Feature Encoder's main function is to achieve the representation vector of drugs and proteins and predict their interaction label. First, using an alphabet dictionary to encode the drug's SMILE and protein's FASTA into arrays with the shape of 1x100 and 1x1000. Then making the encoded array go through the network of CNN and LSTM layers to get the representation vector. And finally concatenating the drug's and protein's representation vectors and using Two-Sided Attention Machanism to predict the interaction label. In the nutshell, the single Feature Encoder part can get a relatively nice DTI prediction performance.

(2) Domain Adaptation Domain Adaptation part works as the supplement and upgrade of Feature Encoder. Here this paper borrows the idea of GAN technique that creating a Discriminator Network and two encoders for training dataset and test dataset respectively. The goal of Discriminator Network is to distinguish the difference between to encoders while the goal of encoder for test dataset is to confuse the Discriminator. By optimazing Discriminator and test dataset's encoder alternately, Domain Adaptation network can finally converting the test dataset representation vector space quite like that of training dataset which making nearly no difference between training and test dataset so that the prediction performance of unseen drugs or proteins can have a great performance.

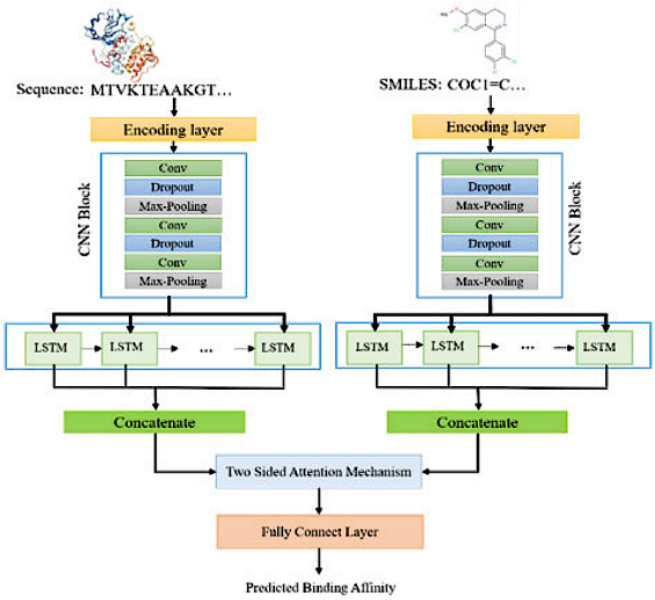


Fig. 5: DeepCDA Feature Encoder Model

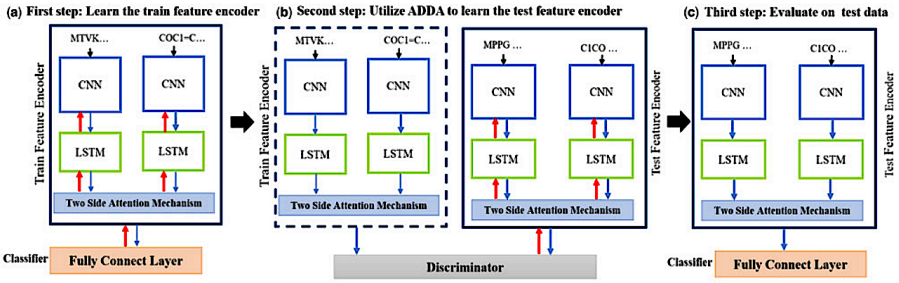


Fig. 6: DeepCDA Domain Adaptation Model

One thing to be mentioned is that this Domain Adaptation technique can be applied to nearly all kinds of networks to improve their DTI prediction performance. Assuming a DTI prediction network $N = N_{representation} + N_{classifier}$, then we can copy $N_{representation}$ as test domain's encoder and replace $N_{classifier}$ by the Discriminator mentioned above. The universality is its biggest advantage.

3 Proposed Method: MAC (MPNN-Attentioned CNN)

After reproducing MolTrans, DeepDTA and DeepCDA mentioned above and analyzing their merit and demerit, We propose our own model for DTI prediction named MAC inspired by DeepPurpose Library[6], which means the combination of MPNN(Message Passing Neural Network), Attention Mechanism and CNN. To describe more clear, we use MPNN to fetch drugs' representation vector and attentioned CNN to that of proteins. Then we combine two representation vectors and use fully connected layers and dropout layers to output our results.

3.1 The Frame of MAC

The frame of MAC is illustrated in Figure 7.

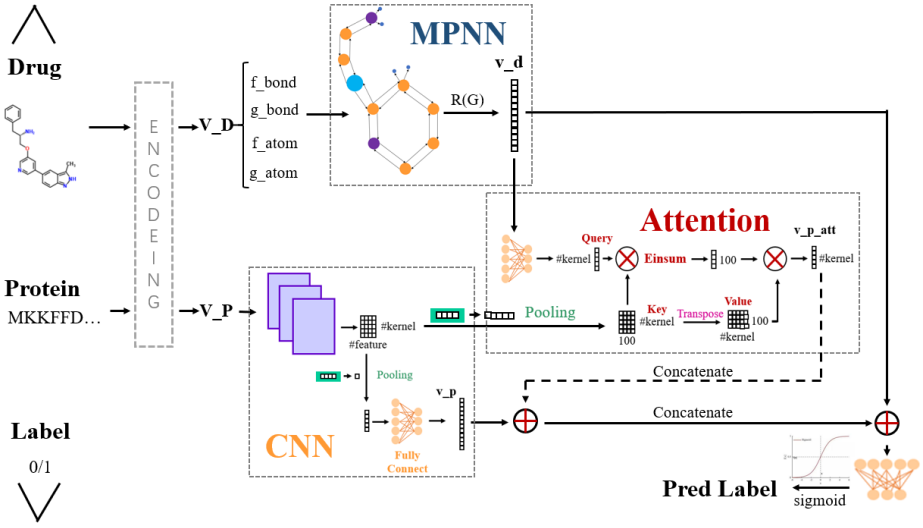


Fig. 7: MAC Frame

3.2 MPNN

MPNN(Message Passing Neural Networks, [4]) is used here to acqiure the representation vector of Drug's SMILE. To reduce the noise caused by two nodes' alternatively passing message, we used Directional-MPNN here that edges hidden states are learned and represent the whole graph. As a Graph Neural Network framework instead of a GNN, it's necessary to self-define Message Passing function, Hidden State function and Readout function to specify our own model. Here's our assumption:

$$M_t(x_v, x_w, h_{vw}^t) = \tau(h_{vw}^t)$$

$$U_t(h_{vw}^t, m_{vw}^{t+1}) = \tau(h_{vw}^t, W_g \cdot m_{vw}^{t+1})$$

$$R(G) = \sum_{v \in V(G)} \tau(W_o \cdot cat(f_{bond}[v], \sum_{w \in N(v)} h_{vw}^T))$$

Message Passing function defines what information one edge can pass to all its neighbour edges. It's the core to the spread of information in MPNN and gives MPNN ability to capture Series's space feature. Assuming there exists two atoms have strong connection while far apart, then their hidden states which are initialized by their feature vectors can pass information to each other after certain epochs training and has similar hidden states meaning that these two atoms has strong connection in the structure of the drug. In our self defined MPNN, for the simplicity of calculation, we directly use edges's hidden states as information.

Update function is applied after every epoch's information passing phase that every edge will update themselves by received information. First, every edge needs to sum up all the information

that it receives which is $m_{vw}^{t+1}) = \sum_{p \in N(v)} M_t(x_p, x_v, h_{pv}^t)$. Then the summed information will be left-dotted by a learned Matrix M_g. Then considering the hidden state of last epoch, we need to add h_{vw}^t. Finally, to further increase the nonlinearity of the MPNN, we use activation of Relu.

Readout function is applied after the whole training process of MPNN and used to sum up all the information learned to output a representation vector. First, we need to get the representation of every node in the whole graph that every node can be expressed as the concatenation of feature vector of atom and the sum of all related edges' hidden states, which is equivalent to $cat(f_{bond}[v], \sum_{w \in N(v)} h_{vw}^T)$ in the definition. Then to increase the expression ability of the whole network, we used the previous result to be left-dotted by a learned Matrix M_o and activated by Relu function. Finally, by summing up all nodes' vector we can get the representation vector of the whole graph.

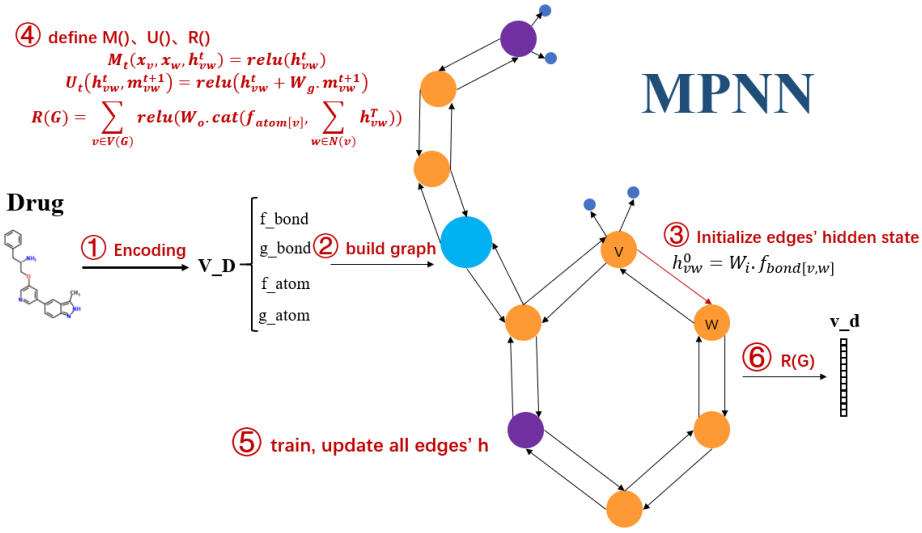


Fig. 8: MPNNFrame

3.3 CNN

CNN (Convolutional Neural Networks[5]), is a kind of feedforward deep neural network using convolution kernels to conduct convolution computation. In our MAC network, we design a CNN network module to learn the representation vector of proteins' FASTAs. First, we used one-hot to encode the FASTA sequences in the datatype of String to get the encoded vector in that of matrix. Then, we used three convolutional kernels with the shape of 16,32 and 48 to conduct convolutional computation then used Maxpooling method to compress matrix into vector. And finally, we used a structure consists of fully connected layers and dropouts to learn the representation vectors.

3.4 Attention Mechanism

Attention Mechanism[2] here is used as the supplement to the CNN for CNN's comparatively poor ability to learn the space feature. The core idea of Attention Mechanism is that by multiplying an weighted array with same shape of target array, we can shift the model's "attention" to the

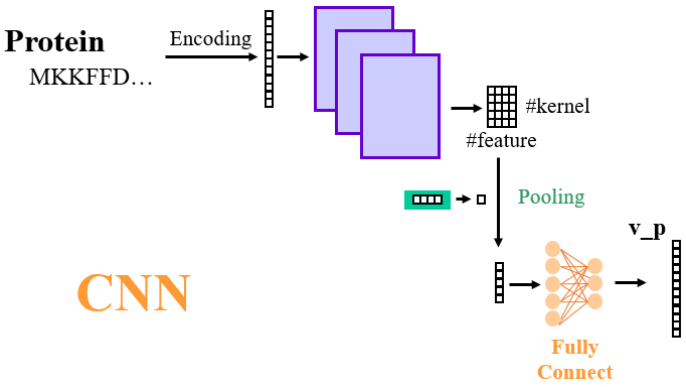


Fig. 9: CNNFrame

part in target array with higher weight by giving them higher contribution to the loss function. To increase the learning effect of FASTA's space feature, we take the learned representation vectors of drugs with fully gained space features in MPNN as the weighted array to help enhance the CNN's space feature learning ability. First, Attention still needs the process of convolution and maxpooling. Then we take drugs' representation vectors which have been reshaped by fully connected layers as the Query, the poolinged result as key and the transpose result as value to get the attentioned protein vectors. The attentioned vector will be concatenated with CNN's vector to become a longer one.

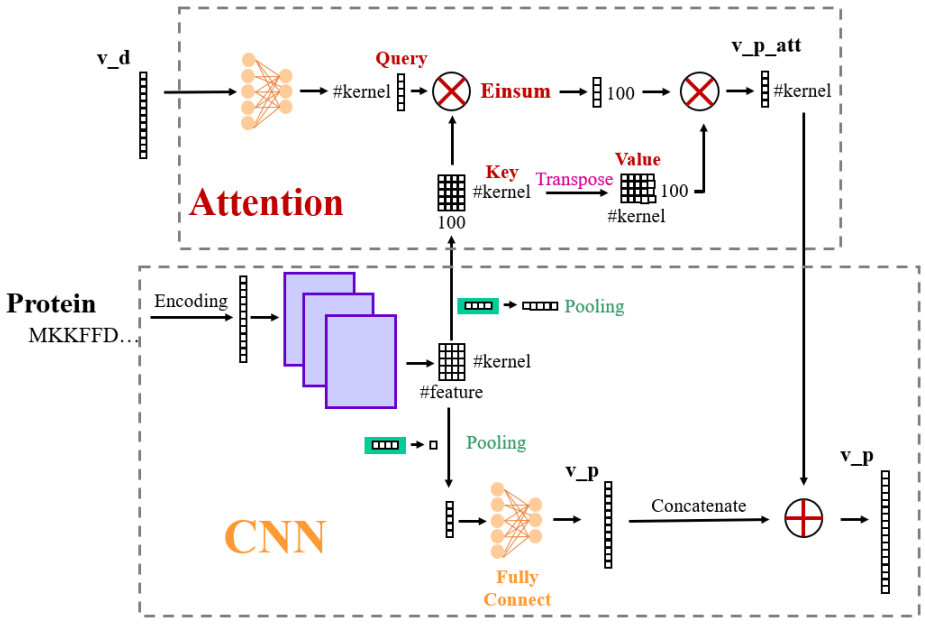


Fig. 10: AttentionFrame

4 Experiments

4.1 Dataset

In this project, we deal with the dataset obtained from TA and at first do some necessary statistical analysis. The result is shown in table . The dataset consists of totally 25303992 data pair samples in which 15856 are positive instances and 25288126 are negative ones. With a ratio of nearly 1600, it undergoes a severve class imbalance. Due to limited resources and time, it's rarely possible to augment positive samples as large as negative samples. Therefore, we take some sampling strategy to remain all positive samples and sample equal negative ones.

	# of samples	# of SMILE categories	# of FASTA categories
positive sample	15856	6771	3766
negative sample	25288126	5504	4599
Overall	25303992	6860	4603

Table 1: Statistics of Dataset

After inspection of data, we find that the organization of drug-target pair appears in a continuous way(same drug or same target). This reminds us that we could design two simple sampling strategies to improve the model performance.

Continuous sampling.

Since the data samples are stored continuously, which means identical drug or target will be placed one by one in the .csv file, the simplest way is to fetch all positive samples and specific amount of negative samples(such as 100,000). Experimental results have shown that it will greatly improve the training efficiency due to sufficient samples for every kind of drug and target. However, since only a proportion of samples are seen in training process, the generalization capability of model is worse than ramdom sampling.

Interval sampling

To tackle generalization problem mentioned above, we could perform interval sampling so that nearly every drug or target could be seen once at least and still although sample amount for specific drug/target is not so much as one from continuous sampling, it could still remain a scale(about 500) for effective training. It appears to be better generalized but requires more training time with low efficiency.

Since the original dataset is too large, we use two different sampled dataset in our experiments:

- **Continuous sampled small dataset**: We sample all positive pairs and the first 100,000 negative pairs. This dataset is used to reduce the training time and compare the performance of different models.
- **Interval sampled large dataset**: We sample all positive pairs and 1,000,000 negative pairs in an interval of 25. It is used in robustness test part.

Note that the eventual version of model we hand in is trained on interval sampled large dataset for better generalization performance.

4.2 Experimental setup

Since normalized frequency value is extremely close to 0, exponential operation is approximately equal to identity operation and softmax will degenerate to L1 normalization, we use the count value instead. According to paper..., prediction model performance varies on different sequence pairs depending on whether their components have been up in training dataset. According to this, we could design a sampling strategy solely for both-seen pair by frequency of occurrence in positive dataset and another one for all remaining pairs.

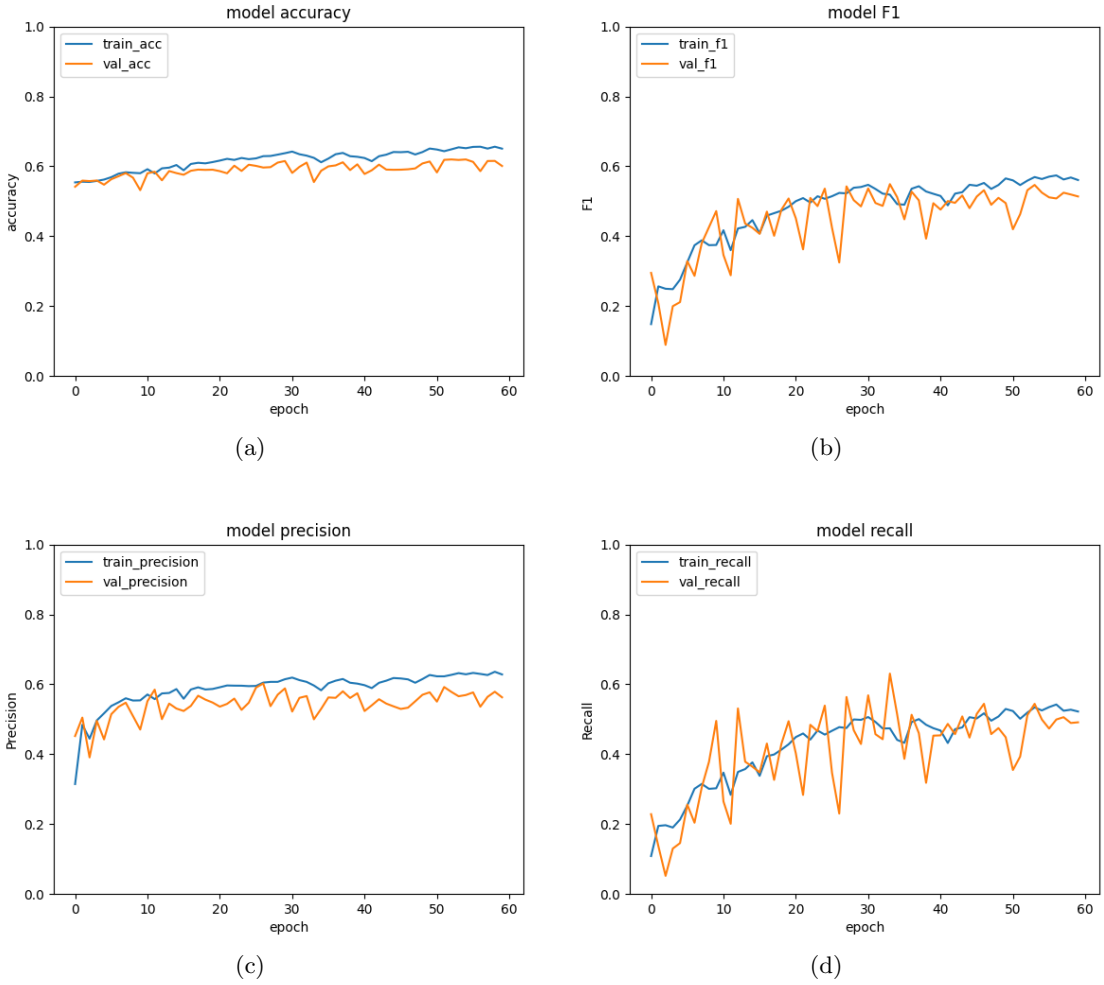


Fig. 11: Performance of MolTrans model

4.3 MolTrans

We first train and validate our text-processing-based model: MolTrans on continuous sampled small dataset described above. After text-based encoding (FCS) described in the theory of MolTrans,

we will get (1,205) vector for drug encoding and (1,545) for protein encoding. We then put these encoding results to transformer networks to get there representations. Finally, we perform pair-wise interaction and predict the interaction results. The results are shown in Figure 11.

We could discover that the accuracy grows slightly and then converge quickly to around 0.64. The final F1 is 0.58, precision is 0.62, recall is 0.52, which are not very good. We also draw the loss curve in Figure 12. The train loss decreases during training and the validation loss fluctuates. We conclude two reasons that possibly result in poor performance of Moltrans.

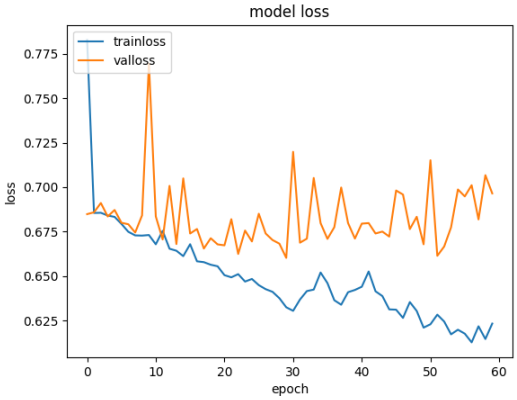


Fig. 12: Loss Curve of MolTrans

1. The model of Moltrans is too complicated by adding embedding layer, attention layer, convolution layer and interaction layer together, which is harder for training. In this experiment, the loss is decreasing but the accuracy improves slowly, which means that the loss could not help update all of the parameters well due to the complicated structure of model.
2. MolTrans is a text processing based model, which relies highly on prior knowledge. In MolTrans, we use ESPF knowledge mentioned in the source paper, which describes the properties and possible interaction between acids and molecules. If prior knowledge is inaccurate or do not fit the data set well, we could hardly learn correct interactions based on it. In this experiment, the prior knowledge may not fit our data well and it may not be able to extract useful properties of drugs and proteins in encoding process.

4.4 DeepDTA

As the first model to incorporate deep learning technique into affinity prediction of drug-target pair, DeepDTA model features by its own simplicity and efficiency. According to parameters described in paper ...., we implement it with tensorflow and record its metrics performance on (a) binary accuracy (b) F1 score (c) precision and (d) recall and the result is shown in Fig 13. To transform the regression problem into classification, we add a sigmoid activation layer before the final output. Detailed parameters are listed in Table 2. The final precision is 0.92, recall is 0.77, F1 score is 0.84 and accuracy is 0.77. We train DeepDTA on continuous sampled small dataset.

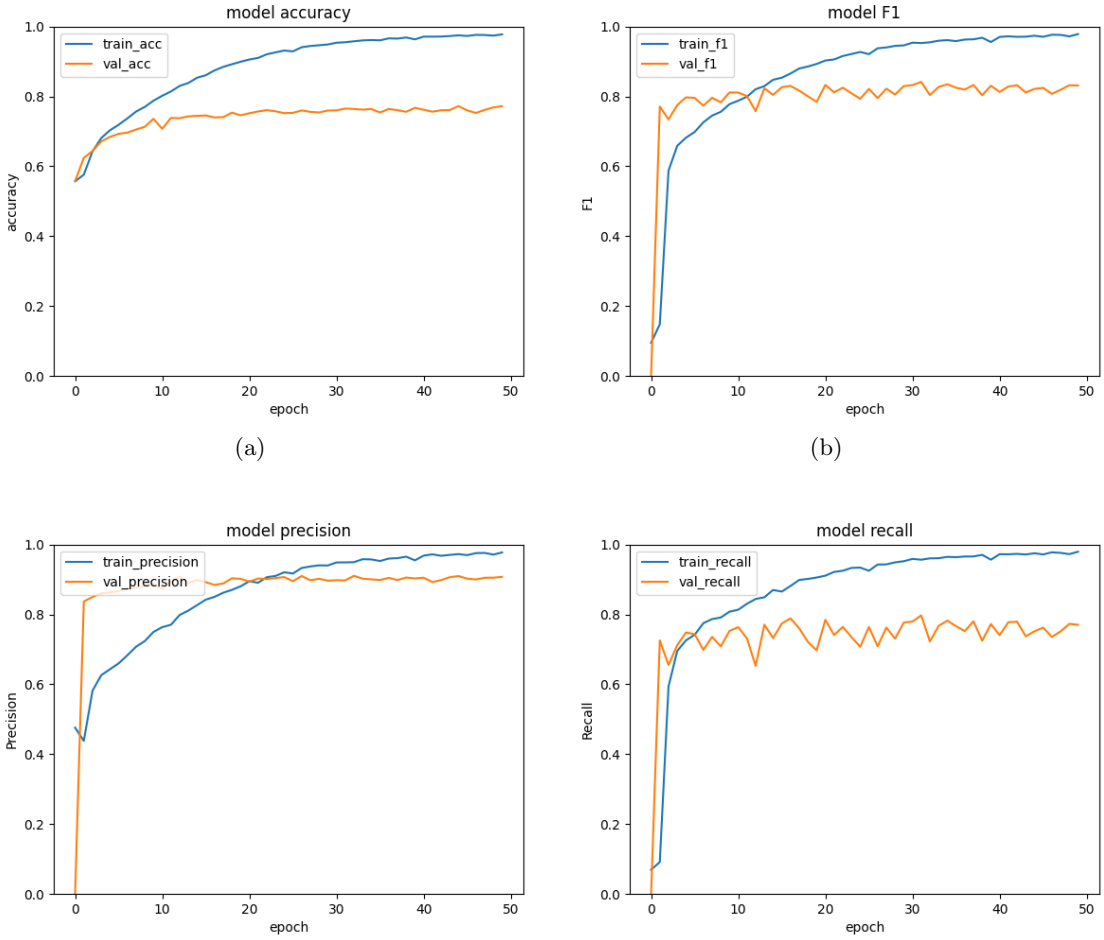


Fig. 13: Performance of DeepDTA model

Parameter	Settings
Kernel Size	32*1;32*2;32*3
filter length(protein)	[4,8,12]
filter length(compound)	[4,6,8]
hidden neuron	1024;1024;512
dropout	0.2
epoch	50
batch size	128
optimizer	Adam
learning rate	2.5e-4

Table 2: Parameter settings of DeepDTA

By analyzing the bottleneck of this model, we propose some essential defects which severely limits DeepDTA architecture's performance.

(1) In embeddding layer, the model utilizes token-level information and tries to learn distributed representation of every single sign, such as a, C and so on. However, it's well known that in many cases only combinations of signs will show their real chemical meanings. For example, $Ca(calcium)$, $Na(sodium)$, $C(carbon)$ and $N(nitrogen)$ enjoy a high similarity w.r.t their token. Unfortunately however, they repectively lie in IB, IA, IVA, VA region and owns significantly different properties. Therefore, Word-level(Cu,Ga,...) or even group level(Ga+,(=O)) information should be incorporated to capture properties more precisely.
(2) CNN model suffers from limited distance detection capability, and if two distant points are correlated with each other they have to pass through deeper layers which might be influenced by gradient vanishing.

In our final model, we will try to tackle with these obstacles.

4.5 DeepCDA

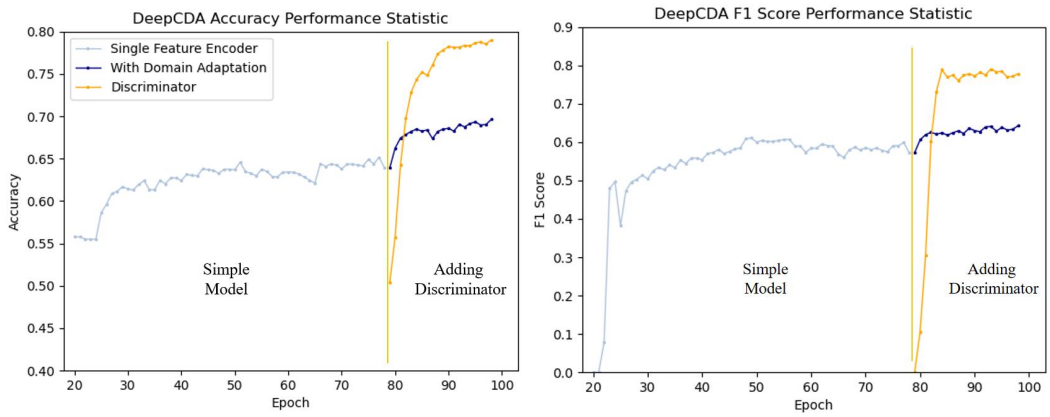
As the same to DeepDTA, DeepCDA's initial model was designed to predict the binding affinity of the given drug and protein pair, which is a continuous value. So to convert it into a classifier not a predictor, we add a sigmoid function after the initial model's final output layer. As required by the TA, in the training process, we mainly statistic the performance of accuracy and F1 score, which are shown by the following two pictures. One thing to be mentioned is that because of the limited computing resources, the training process was not enough that the number of epochs with discriminator was just 20 not 50, leading to the performance was actually not the upper bound. We train DeepCDA on continuous sampled small dataset.

By analyzing the performance and the bottleneck of the DeepCDA model, we obtain some useful conclusions:

- Domain Adaptation technique does have the ability to improve a model's performance when training dataset and test dataset have different domain distribution that in DeepCDA, the accuracy has risen about 6 percents while F1 score risen about 7 percents, which can be even higher with adequate training epochs.
- The Feature Encoder's performance was not ideal enough, severely limiting the total model's performance, whose accuracy was just about 64 percents. Possible reasons are that: 1.6 million parameters need more epochs to be training; the network architecture was so single that cannot capture enough spatial feature of drugs and proteins; or so on.

Model Name	Encoder	Discriminator
dropout	0.1	0.3
epoch	50	20
batch size	256	1024
optimizer	Adam	Adam
learning rate	1e-5	4e-5

Table 3: Parameter settings of DeepCDA



(a) (b)

Fig. 14: Performance of DeepCDA

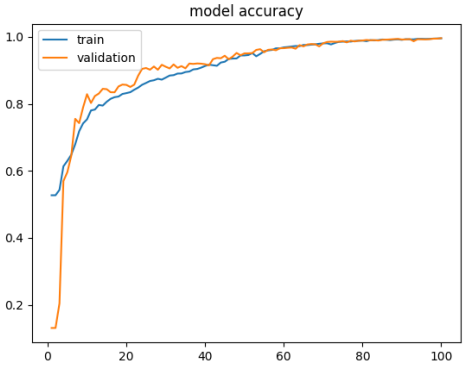
4.6 MAC:MPNN - Attentioned CNN

We finally conduct experiments on our MAC model on continuous sampled small dataset. We consider three settings below:

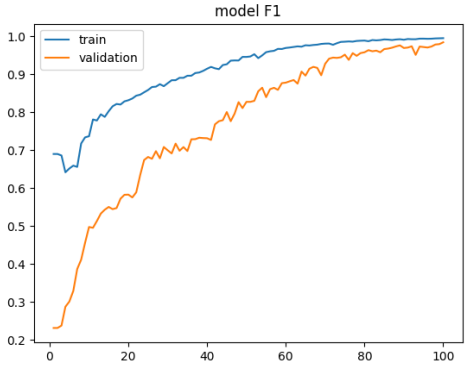
- **Concatenation and attention**: Concatenate attention results with convolution results of proteins as classifier input.
- **Attention**: Do not concatenate and use attention results alone as classifier input. This is the basic setting of our MAC model.
- **No attention**: Do not use attention. This setting will help us determine the influence of attention module.

We only plot accuracy and F1 score of three settings in Figure 15, since they are two main evaluation metrics. We now analyze these curves based on accuracy and F1 score:

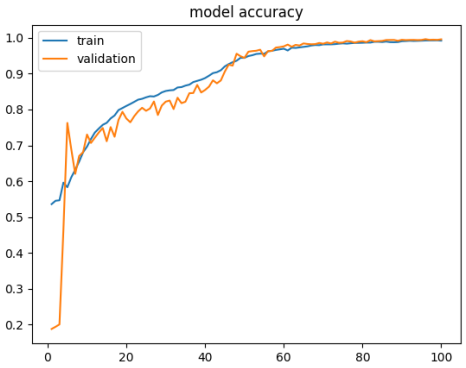
1. As for accuracy, three different settings have similar performance, reaching nearly 0.98 classification accuracy in both training and validation set. It indicates that our MAC model could identify drug target interactions very well.
2. For F1, all of three settings could converge to a high level (around 0.95) in the end. However, our model with an attention mode converges faster than other two settings. According to the properties of our sampled data set, we have a large number of negative samples but only a few positive samples. Therefore, many models may easily classify positive samples to negative side mistakenly, resulting in low recall. In this case, higher F1 indicates that the model could identify positive samples accurately. Based on this analysis, we could conclude that our MAC model is able to classify positive and negative samples precisely and quickly.
3. Adding attention mode to protein vectors using drug information is useful and could serve as the protein encoding results without the concatenation of original convolution output of protein. Besides, attention layer enjoy great interpretability. With attention added, we could interpret the detailed interaction sites of protein and drug.



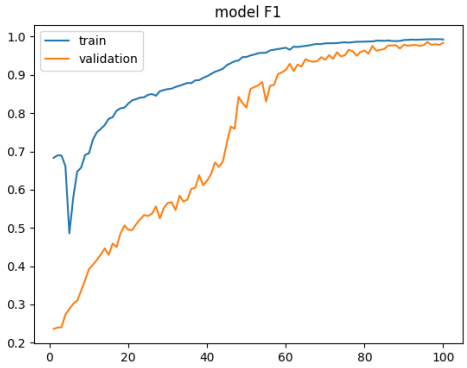
(a) Concatenation and Attention Accuracy



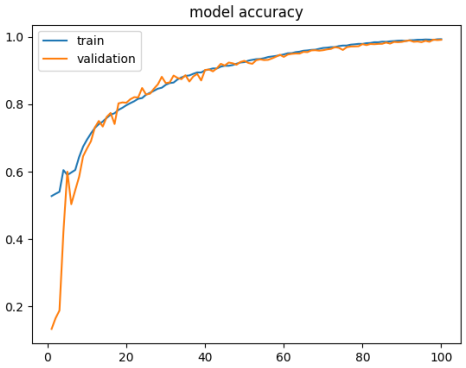
(b) Concatenation and Attention F1



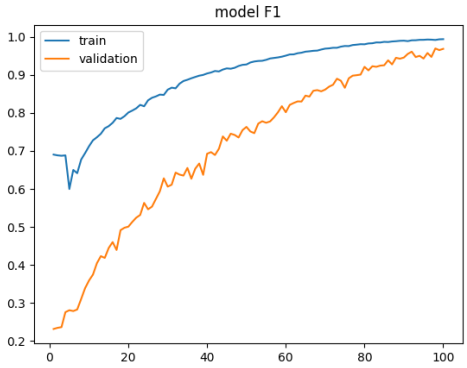
(c) Attention Accuracy



(d) Attention F1



(e) No Attention Accuracy



(f) No Attention F1

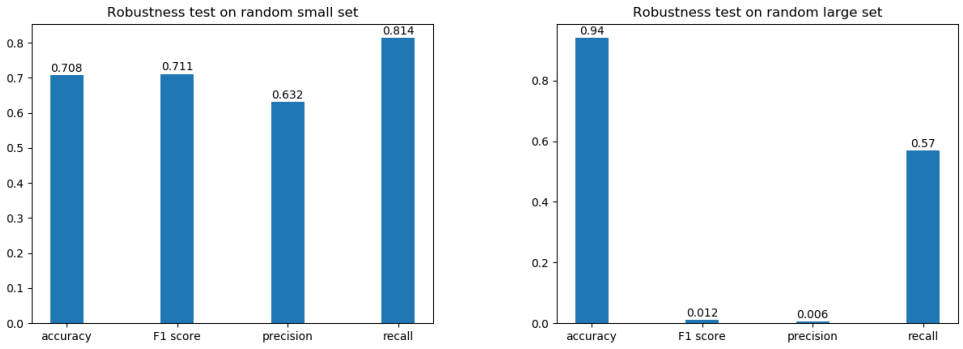
Fig. 15: Performance of our MAC model

4.7 Robustness Test

Since the distribution of data is accompanied with severe classing imbalance, especially from the real-world data. We simulate two kinds of environments and perform test on our MAC model. In this part, our models are trained on interval sampled large dataset. Two test sets are:

- **Random small**: 15,000 positive pairs+20,000 randomly sampled negative pairs.
- **Random large**: 1,500 randomly sampled positive pairs+ 2,500,000 randomly sampled negative pairs.

Experimental results are shown in Figure 16. Since it's time-consuming, we only display results of MAC here.



(a) (b)

Fig. 16: Robustness test on random small and random large test set

4.8 Visualization

By outputing Attention Layer's weight distribution, our research group achieved the visualization of the action sites of specific protein and drug with the power of Discovery Studio. We draw the interaction illustration figures and the heat map of attention layer's weights of protein in Figure 18.

The output of Attention Layer has the same size as the embedding of proteins. And the higher weight means certain part of the protein and the given drug has greater interactive force, which can be taken as action site. In Figure 18 (a), (b), yellow part of protein denotes potential interaction site and blue molecule is the chosen drug. We also plot the heat map according to attention layer's weights and map weights to original protein sequence. Therefore, x-axis in Figure 17 denotes the sequence length, and different colors denote different weights. From the heatmap distribution, we could find that nearly all attention is contentrated on 189-207 area, and thus could trace back corresponding segment of protein.

With the visualization function of Discovery Studio by inputing the original FASTA of the protein and highlight function, our research group achieved the visualization of action site.

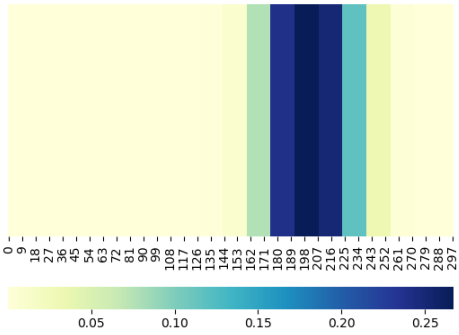
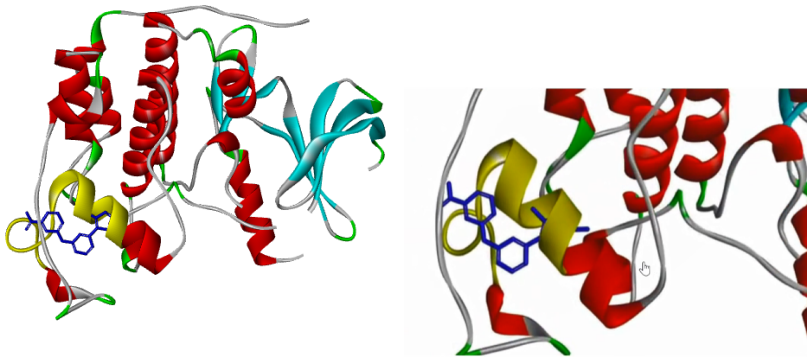


Fig. 17: Protein Weight Heat Map



(a) Global visualization (b) Local visualization

Fig. 18: Visualization results of drug target interactions

4.9 Summary of Performance

To give a clear overview of how different models perform on our data set, we summary our results of different models on **validation set** and robustness test in Table 4.

5 Conclusion

By reappearing the work of Moltrans, DeepCDA and DeepDTA and analyzing the experiment results of them, our research group jumped to the conclusion that existing methods has the disadvantages including poor ability to extract the spatial features of drugs' SMILEs and proteins' FASTAs, the beginning procedure of embedding based on words leading to poor ability to express complex structures such as functional groups and weak generalization ability for unseen drugs or proteins in test dataset, all of which led to the bottleneck of DTI prediction mission.

Table 4: A summary of performance of different models (validation set)

Model	accuracy	F1 score	precision	recall
Moltrans	0.601	0.546	0.570	0.525
DeepDTA	0.748	0.822	0.903	0.758
DeepCDA	0.696	0.643	0.590	0.601
MAC (concatenation and attention)	0.996	0.985	0.993	0.977
MAC (attention)	0.996	0.984	0.995	0.973
MAC (no attention)	0.992	0.968	0.970	0.966
Robustness test (random small)	0.708	0.711	0.632	0.814
Robustness test (random large)	0.940	0.012	0.006	0.570

The distribution of positive and negative samples in dataset has a great influence on the overall experiment results which can be expressed by robustness test. We can see an obvious decrease on F1 score, precision and recall while an increase on accuracy when the number of negative samples is far greater than that of positive samples. Too many negative samples make the network tend to output all zeros to obtain higher accuracy. So weighted randomly sampling is quite necessary. By the way, to many unseen drugs and proteins can also lower the experiment result.

The introduction of attention mechanism makes the whole model get a slight performance increasement while greatly enhances the interpretability of MAC, which is not available in the previous DTI models. By outputing the result of attention mechanism layer, it's easily to be found that the distribution of attention weight is more and more focused on a certain segment of protein, where is the action site.

Message Passing Neural Network does has the ability to extract the spatial feature of drugs' SMILEs. With fine-tuned parameters and timesteps, one bond's feature vector can be passed to a distant one with very little loss and update its hidden state so that the spatial connection of these two bonds can be extracted and learned by MPNN. Compared with traditional methods like CNN or LSTM, MPNN framework does achieved an obvious performance improvement.

6 Division of labor and Contribution

Table 5: The Division of Total Labor

Member	Work Load		
Chun Chi (33%)	DeepCDA	PPT	Report
Yifan Liu (33%)	MolTrans	MAC Training	Report
Shengyuan Hou (33%)	DeepDTA	MAC Coding	Report

References

1. Abbasi, K., Razzaghi, P., Poso, A., Amanlou, M., Ghasemi, J.B., Masoudi-Nejad, A.: Deepcda: deep cross-domain compound-protein affinity prediction through LSTM and convolutional neural networks. Bioinform. **36**(17), 4633–4642 (2020). https://doi.org/10.1093/bioinformatics/btaa544, `https://doi.org/10.1093/bioinformatics/btaa544`
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

3. Ding, H., Takigawa, I., Mamitsuka, H., Zhu, S.: Similarity-based machine learning methods for predicting drug–target interactions: a brief review. Briefings in bioinformatics **15**(5), 734–747 (2014)
4. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning. pp. 1263–1272. PMLR (2017)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
6. Huang, K., Fu, T., Glass, L.M., Zitnik, M., Xiao, C., Sun, J.: Deeppurpose: a deep learning library for drug–target interaction prediction. Bioinformatics (2020)
7. Huang, K., Xiao, C., Glass, L.M., Sun, J.: MolTrans: Molecular Interaction Transformer for drug–target interaction prediction. Bioinformatics **37**(6), 830–836 (10 2020). https://doi.org/10.1093/bioinformatics/btaa880, `https://doi.org/10.1093/bioinformatics/btaa880`
8. Luo, H., Mattes, W., Mendrick, D.L., Hong, H.: Molecular docking for identification of potential targets for drug repurposing. Current topics in medicinal chemistry **16**(30), 3636–3645 (2016)
9. Öztürk, H., Özgür, A., Ozkirimli, E.: Deepdta: deep drug–target binding affinity prediction. Bioinformatics **34**(17), i821–i829 (2018)
10. Öztürk, H., Ozkirimli, E., Özgür, A.: A comparative study of smiles-based compound similarity functions for drug-target interaction prediction. BMC bioinformatics **17**(1), 1–11 (2016)
11. Thafar, M.A., Olayan, R.S., Ashoor, H., Albaradei, S., Bajic, V.B., Gao, X., Gojobori, T., Essack, M.: Dtigems+: drug–target interaction prediction using graph embedding, graph mining, and similarity-based techniques. Journal of Cheminformatics **12**(1), 1–17 (2020)
12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
13. Wikipedia contributors: Fasta — Wikipedia, the free encyclopedia (2020), `https://en.jinzhao.wiki/w/index.php?title=FASTA&oldid=989327866`, [Online; accessed 19-June-2021]
14. Wikipedia contributors: Simplified molecular-input line-entry system — Wikipedia, the free encyclopedia (2021), `https://en.jinzhao.wiki/w/index.php?title=Simplified_molecular-input_line-entry_system&oldid=1022582088`, [Online; accessed 19-June-2021]