Compound-Protein Interaction Prediction with Sparse Perturbation-Aware Attention

Qiwen Wang[1,2], Chen Lin[2,3](✉), Wei Su[3], Liang Xiao[3], and Xiangxiang Zeng[4]

[1] Institute of Artificial Intelligence, School of Informatics,
Xiamen University, Xiamen 361104, China

[2] National Institute for Data Science in Health and Medicine, Xiamen University,
Xiamen 361104, China
chenlin@xmu.edu.cn

[3] School of Informatics, Xiamen University, Xiamen 361104, China

[4] College of Computer Science and Electronic Engineering, Hunan University,
Changsha 410082, China

Abstract. Compound-Protein Interaction (CPI) prediction is a crucial task in drug discovery. Modern CPI prediction models are mostly based on the attention mechanism. However, the attention scores are often inaccurate, i.e., functionally irrelevant substructures can still receive moderate attention scores, and attention scores can not distinguish compounds with similar structural topology but different pharmacological properties. We propose SPACPI to address this problem from three perspectives, i.e., (1) identifies important compound substructures by integrating auxiliary information from molecular fingerprints, (2) determines important compound atoms by learning each atom's tolerance to different perturbation amplitudes, (3) obtains more robust model parameters by focusing on the topK important atoms. Experiments on two benchmark datasets and two label-reversal datasets show that SPACPI outperforms the state-of-the-art CPI prediction model with an average increase of 5.02% across different datasets and evaluation metrics. Visualization verifies that SPACPI can produce more accurate and explainable predictions.

Keywords: Compound-Protein Interaction Prediction · Drug Discovery · Perturbation-Aware Attention

1 Introduction

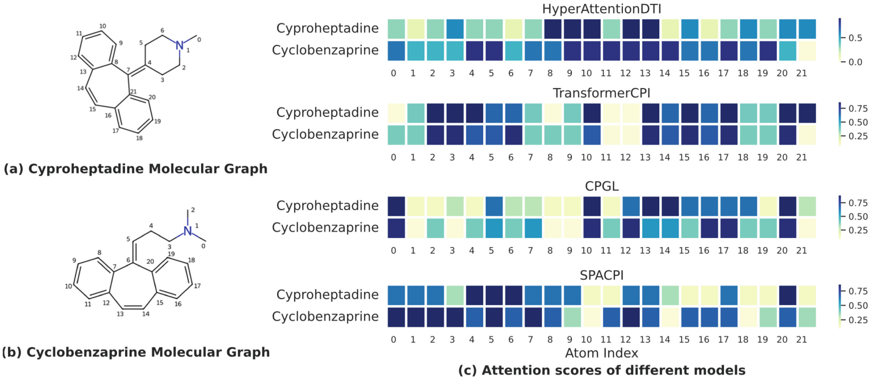
Compound-Protein Interaction (CPI) prediction is a crucial task in drug discovery, which seeks to accurately predict the existence of compound-protein interaction without costly vital trials and time-consuming development cycles. Recent CPI prediction methods mostly adopt the attention mechanisms to capture the inherent relationships between compounds and proteins [1–4].

However, previous attention-based approaches risk the overfitting problem because the compound and protein data have a complex structure and the CPI labels are sparse

D.-S. Huang et al. (Eds.): ICIC 2024, LNBI 14882, pp. 72–83, 2024.
https://doi.org/10.1007/978-981-97-5692-6_7

[5, 6]. Therefore, it is challenging for the models to learn accurate attention scores. Specifically, functionally irrelevant substructures can still receive moderate attention scores, which is undesirable because the interactions only involve pharmacophores in drug compounds and binding sites in protein sequences [7]. Attention scores for unseen compounds in the testing set are indistinguishable from structurally similar compounds in the training set, which can be biologically unreasonable because compounds with similar structural topology may have dissimilar pharmacological properties.

For instance, as shown in Fig. 1(a)–(b), Cyproheptadine and Cyclobenzaprine are structurally similar compounds with different pharmacological properties. Cyproheptadine interacts with the Histamine H1 receptor, while Cyclobenzaprine does not. Thus, in predicting interactions with the Hismanine H1 receptor, the attention distribution of atoms in Cyproheptadine and Cyclobenzaprine should focus on the structurally different parts. Nonetheless, if we place the Cyproheptadine-Hismanine H1 receptor pair into the training set, and the Cyclobenzaprine-Hismanine H1 receptor pair into the testing set, after implementing the existing models, i.e., HyperAttentionDTI [8], TransformerCPI [1], and CPGL [4], we can see from Fig. 1(c) that the attention distributions of atoms in these models are similar for Cyproheptadine and Cyclobenzaprine, the 2-dimethylaminoethyl group (atom 0–6) which has the strongest effect is assigned low weights, and functionally irrelevant atoms (atom 17–19) in Cyclobenzaprine are assigned abnormally high weights. As a result, these models mistakenly predict the Cyclobenzaprine-Hismanine H1 receptor pair as positive while the actual label is negative. On the contrary, our model's learned attention distributions can highlight their different parts, and functionally irrelevant substructures are downweighed. The prediction accuracy has been boosted (more details in Sect. 3).



(a) Cyproheptadine Molecular Graph

(b) Cyclobenzaprine Molecular Graph

Fig. 1. Atom-level attention scores for Cyproheptadine and Cyclobenzaprine in different models.

Naturally, to generalize CPI prediction to different datasets, we want the attention values to be high on functional atoms related to compound-protein interactions and low for unrelated atoms. This is challenging because of the insufficient labels. We propose SPACPI (Sparse Perturbation-Aware Attention for Compound–Protein Interaction prediction) that tackles the problem of inaccurate attention scores from three perspectives. Firstly, SPACPI identifies important compound substructures by integrating auxiliary information from molecular fingerprints. To our knowledge, SPACPI is the first CPI prediction model to incorporate chemical information from MACCS and Pharmacophore ErG fingerprints into structural topology information. Secondly, SPACPI proposes a novel perturbation-aware attention mechanism to determine important compound atoms in a self-supervision manner. SPACPI assumes that critical atoms are more sensitive to noise. By applying a random perturbation to each compound atom and minimizing the difference between predictions before and after perturbation, SPACPI learns each atom's tolerance to different perturbation amplitudes. Finally, SPACPI encourages the learned compound attention to focus on a few important atoms to reduce compound features and obtain more robust model parameters.

Experiments on two benchmark datasets and two label-reversal datasets show that SPACPI achieves superior performance. SPACPI outperforms the state-of-the-art CPI prediction model with an average increase of 5.02% across different datasets and evaluation metrics. We also provide visualization to verify that SPACPI can produce more accurate and explainable predictions.

2 Methodology

Inspired by previous work [8–10], we also treat CPI prediction as a binary classification task, which takes the features of compound c and features of protein p as input, feeds the input through three layers, i.e., encoding layer, interaction layer, and prediction layer, and predicts the CPI label (Sect. 2.1). As shown in Fig. 2, SPACPI also adds a perturbation-aware attention mechanism (Sect. 2.2) in the overall framework.

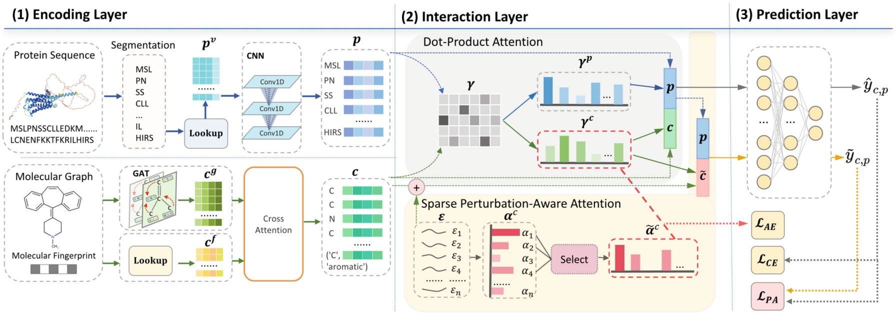


Fig. 2. Framework of SPACPI.

2.1 Prediction Backbone

The encoding layer is responsible for converting the input to the embedding matrix.

Protein Features. Since CPI depends on the protein substructures, substructure segmentation is usually applied to the protein sequences. In order to decompose proteins into substructures driven by domain knowledge, we follow the previous method [11].

Specifically, we initialize a vocabulary to convert protein sequences to a set of predefined sub-sequences and then translate all sub-sequences to real-valued embeddings $p^v \in \{R\}^{m \times d}$ via a learnable dictionary lookup matrix, where m is the number of substructure sequences for protein. Then, each real-valued embedding p_j^v is passed into the Convolutional Networks with Conv1D layers to obtain the embedding for protein substructures:

$$\mathrm{p_j} = Conv\left(\mathrm{p_j^v}\right), 1 \leq j \leq m \tag{1}$$

where $p_j^v \in \{R\}^d$ is the embedding for the j-th substructure, $Conv$ is the CNN function.

Compound Features. Graph Neural Networks are generally chosen to extract features from molecular graphs. We adopt Graph Attention Networks (GAT) because they are superior in learning the strength of the connection (i.e., chemical bonds) between different nodes (i.e., atoms) [12]. Specifically, we transform the SMILES formula of a compound into an undirected graph by the RDKit toolkit [13], where the nodes represent atoms, and the edges represent the chemical bonds between nodes. Then, we utilize GAT to update each node embedding by aggregating the information of itself and its neighbors iteratively. The output is $c_i^g \in \{R\}^d,\ 1 \leq i \leq n$, where n is the number of atoms in the compound, and d is the embedding dimension.

Unlike previous studies, SPACPI incorporates molecular fingerprints in CPI because they can holistically express molecular characteristics and supplement structural information [14]. SPACPI incorporates two complementary fingerprints, namely the MACCS fingerprint [15] and the Pharmacophore ErG fingerprint [16]. The MACCS (Molecular ACCess System) fingerprint contains most atomic properties, bond properties, and atomic neighborhoods at diverse topological separations. The Pharmacophore ErG fingerprint applies pharmacophore-type node descriptions to encode molecular properties. We use RDKit to obtain the two molecular fingerprints, concatenate them, and then use a learnable dictionary lookup matrix to translate all atomic properties to real-valued embeddings, which can be represented as $c^f \in \{R\}^{l \times d}$, where l is the number of atomic properties extracted from the fingerprint.

Next, we incorporate chemical properties from molecular fingerprints into structural information by utilizing cross-attention:

$$\mathrm{c_i} = \mathrm{c_i^g} + softmax\,(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d}}})V \tag{2}$$

where $Q = c_i^g W_Q$; $K = c^f W_K$; $V = c^f W_v$, W_Q, W_k and W_V are learnable projection matrices.

Interaction Layer. The encoding step outputs a set of atom embeddings $c = \{c_1, c_2, ..., c_n\}$ of compound c and sub-structure embeddings $p = \{p_1, p_2, ..., p_m\}$ of protein p. To capture the interaction activeness of each compound and protein, we apply the dot-product attention,

$$\gamma^{\mathrm{c}} = Softmax\left(\frac{1}{\mathrm{m}}\sum_{\mathrm{j=1}}^{\mathrm{m}} \gamma_{\mathrm{i,j}}\right), \gamma^{\mathrm{p}} = Softmax\left(\frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{n}} \gamma_{\mathrm{i,j}}\right)$$
$$\gamma_{\mathrm{i,j}} = \mathrm{ReLu}(\mathrm{c_i W_c})\mathrm{ReLu}\left(\mathrm{p_j W_p}\right)^{\mathrm{T}}, \tag{3}$$

where W_c and $W_p \in \{R\}^{d \times d}$ are the learnable projection matrices. $\gamma_{i,j}$ measures the interaction between each compound atom c_i and protein substructure p_j, γ_i^c is the atom c_i's activeness aggregated over all protein substructures, and γ_j^p is the substructure p_j's activeness aggregated over all compound atoms.

To represent the information of a compound, we should weigh each atom c_i by its interaction activeness. Similarly, we should combine the embeddings of each substructure of a protein by its interaction activeness. This gives us the final representations of a compound and a protein:

$$c = \sum_{\mathrm{i}=1}^{\mathrm{n}} \gamma_{\mathrm{i}}^{\mathrm{c}} \mathrm{c_i}, p = \sum_{\mathrm{j}=1}^{\mathrm{m}} \gamma_{\mathrm{j}}^{\mathrm{p}} \mathrm{p_j}, \tag{4}$$

The Prediction Layer Is a three-layer fully connected network to predict the interaction state:

$$\hat{y}_{\mathrm{c,p}} = \mathrm{FFN}([\mathrm{c}; \mathrm{p}]), \tag{5}$$

where $[c; p]$ is the concatenation operation, the activation function is *ReLu*. The parameters are optimized via the cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}} = \sum_{\mathrm{c,p}} \mathrm{y_{c,p}} log \hat{\mathrm{y}}_{\mathrm{c,p}} + \left(1 - \mathrm{y_{c,p}}\right) log \left(1 - \hat{\mathrm{y}}_{\mathrm{c,p}}\right), \tag{6}$$

2.2 Perturbation-Aware Attention

Although γ^c measures the interaction activeness of a compound, as mentioned in Fig. 1, it can be deficient in identifying the decisive substructures within compound sequences.

Intuitively, less critical atoms have a higher tolerance for noise, i.e., if we add a large perturbation on less critical atoms, the prediction is unaffected. Thus, our goal is to learn a perturbation vector ε_i^c and add ε_i^c to each compound atom c_i and keep the predictions before and after perturbation unchanged.

In practice, we first initialize $\varepsilon_i^c, \forall_i$ following a Gaussian distribution and optimize $\varepsilon_i^c, \forall_i$ within a finite number of iterations (three iterations) to minimize $\mathcal{L}_{\mathrm{PA}}$. . There are two advantages: (1) we use a small number of update steps to speed up the training process without expensive adversarial training, and (2) the random initialization introduces randomness to avoid over-fitting.

$$\mathcal{L}_{\mathrm{PA}} = \sum_{\mathrm{c,p}} \left(\hat{\mathrm{y}}_{\mathrm{c,p}} - \tilde{\mathrm{y}}_{\mathrm{c,p}}\right)^2, \tag{7}$$

where $\hat{\mathrm{y}}_{\mathrm{c,p}}$ and $\overset{\sim}{\mathrm{y}}_{\mathrm{c,p}}$ represent the prediction between compound c and protein p before and after perturbation. $\hat{\mathrm{y}}_{\mathrm{c,p}}$ is obtained by Eq. 5, $\tilde{y}_{c,p} = FFN([\tilde{c}; p])$, and $\tilde{c}_i = c_i + \varepsilon_i^c$ is the atom representation after perturbation.

After learning the perturbation ϵ, we can calculate the perturbation-aware atom-level importance score by:

$$\alpha_{\mathrm{i}}^{\mathrm{c}} = 1 - \frac{\left\| \varepsilon_{\mathrm{i2}}^{\mathrm{c2}} \right\|}{\max_{\mathrm{j}} \left\| \varepsilon_{\mathrm{j2}}^{\mathrm{c2}} \right\|}, \tag{8}$$

It is well-regarded that feature selection can improve the generalization capability of machine learning models [17–19]. Intuitively, to reduce the features, we should select the most influential features. Thus, we sparsify the features by selecting the top $K = n \times \beta$ influential atoms with the largest perturbation-aware attention, where β is the sparse ratio. To downweigh the scores of irrelevant atoms, we set the atoms not in $top - K(\alpha^c)$ to 0:

$$\tilde{\alpha}_{\mathrm{i}}^{\mathrm{c}} = \begin{cases} \alpha_{\mathrm{i}}^{\mathrm{c}}, \; if \;\; \alpha_{\mathrm{i}}^{\mathrm{c}} \in top - K(\alpha^{\mathrm{c}}) \\ 0, \quad otherwise \end{cases}, \tag{9}$$

We do not directly use $\tilde{\alpha}^c$ as the final attention. Instead, as we want the model to autonomously learn appropriate attention scores, we pull γ^c closer to $\tilde{\alpha}^c$, which gives the following objective:

$$\mathcal{L}_{\mathrm{AE}} = \sum_{c} KL(\gamma^{\mathrm{c}} || Softmax(\tilde{\alpha}^c)), \tag{10}$$

The overall optimization function is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{AE}} + \mathcal{L}_{\mathrm{CE}}, \tag{11}$$

The optimization process is to alternatively update γ^{c}, γ^{p} and $\tilde{\alpha}^c$. In each epoch, we first fix the values of γ^{c}, γ^{p} to obtain $\tilde{\alpha}^c$. To do so, we initialize ε_i^c, $\forall i$, $\forall c$ following a Gaussian distribution and optimize ε_i^c within a finite number of iterations $T = 3$ by Eq. 7. Then, we fix $\tilde{\alpha}^c$ and update γ^c by Eq. 11.

3 Experiments

3.1 Experimental Setup

Datasets. We follow [1] to evaluate SPACPI on two public datasets: i.e., the Human dataset and the Caenorhabditis elegans dataset [20] and two label reversal datasets, namely GPCR and Kinase. In the label reversal datasets, a ligand in the training set appears in only one type of interaction (i.e., positive or negative pairs), while the same ligand appears in samples of opposite categories in the test set. Statistics of the datasets are summarized in Table 1.

Competitors. We compare SPACPI with five traditional approaches, such as K-Nearest Neighbours (KNN), Random Forest (RF), L2-logistic (L2), Support Vector Machine (SVM) and Graph Convolution Networks (GCN) [21]. We also compare SPACPI with five recent CPI prediction models, including CPI-GNN [20], GraphDTA [22], DeepConvDTI [5], TransformerCPI [1], HyperAttentionDTI [8] and CPGL [4]. For KNN,

Table 1. Summary of the datasets.

	Label Reversal	Partition	#Proteins	#Compounds	#Positives	#Negatives
Human	No	Random-split	852	1,052	3,369	3,359
C.elegans	No	Random-split	2,504	1,434	4,000	3,786
GPCR	Yes	Well-designed	356	5,359	7,989	7,354
Kinase	Yes	Well-designed	229	1,644	23,190	88,047

RF, L2, SVM, CPI-GNN, GCN and GraphDTA, we copy the results [1]. For other competitors, we use their original implementations. On Human and C.elegans datasets, we randomly split the training/testing set, and conduct five runs with different random seeds. On GPCR and Kinase datasets, we use the default training/testing split.

3.2 Implementation Details

We set the number of GAT layers to 3, the number of CNN layers to 3, the CNN kernel size to 3, the vector dimension to 32, and the sparse ratio to 0.5. We use Adam [23] optimizer. The learning rate and weight decay are set to 1e$-$4 for all the datasets. The dropout rates are set to 0.2, 0.2, 0.2, and 0.5 for Humans, C.elegans, GPCR, and Kinase, respectively. The number of training epochs is set to 200. Our codes are available at[1].

3.3 Comparative Performance

From Table 2 and Table 3, we have the following observations: (1) SPACPI outperforms all competitors, achieving an improvement of 1.13%, 0.92%, 0.23% and 4.92% over the best competitor in terms of AUC in Human, C.elegans, GPCR and Kinase datasets, respectively. These results demonstrate the effectiveness and generalization ability of SPACPI on different datasets. (2) On GPCR dataset, SPACPI achieves slightly better performance than the second best model, which is TransformerCPI, in terms of AUC, while SPACPI's AUPR result is significantly higher than TransformerCPI. For classification tasks, AUC considers the overall performance of positive and negative examples, while AUPR focuses more on identifying positive examples. Given the importance of recognizing positive CPI, SPACPI is significantly more effective than TransformerCPI. (3) SPACPI achieves a remarkable improvement of 30.00% in terms of AUPR and 4.92% in terms of AUC on Kinase dataset, compared with the SOTA CPGL. Most ligands in the Kinase dataset possess nearly ten times more non-interacting than interacting pairs. The high imbalance between positive and negative samples increases the risk of models memorizing ligand patterns and overfitting the training set. SPACPI's remarkable performance on the Kinase dataset demonstrates its ability to extract a limited amount of influential features and generalize well to imbalanced datasets.

[1] https://github.com/xmudm/spacpi

Table 2. Comparative performance of different methods in Human, C.elegans datasets.

	Human			C.elegans		
	AUC	Precision	Recall	AUC	Precision	Recall
KNN	0.860	0.927	0.798	0.858	0.801	0.827
RF	0.940	0.897	0.861	0.902	0.821	0.844
L2	0.911	0.913	0.867	0.892	0.890	0.877
SVM	0.910	0.966	0.969	0.894	0.785	0.818
CPI-GNN	0.970	0.918	0.923	0.978	0.938	0.929
GCN	0.956 ± 0.004	0.862 ± 0.006	0.928 ± 0.010	0.975 ± 0.004	0.921 ± 0.008	0.927 ± 0.006
GraphDTA	0.961 ± 0.004	0.879 ± 0.040	0.910 ± 0.020	0.970 ± 0.006	0.930 ± 0.010	0.921 ± 0.010
TransformerCPI	0.971 ± 0.002	0.913 ± 0.003	0.923 ± 0.004	0.978 ± 0.004	0.933 ± 0.005	0.935 ± 0.005
HyperAttentionDTI	0.970 ± 0.004	0.907 ± 0.010	0.922 ± 0.010	0.974 ± 0.005	0.937 ± 0.004	0.933 ± 0.010
CPGL	0.973 ± 0.003	0.911 ± 0.010	0.923 ± 0.006	0.982 ± 0.003	0.939 ± 0.005	0.940 ± 0.005
SPACPI(w/o fingerprint)	0.970 ± 0.004	0.911 ± 0.004	0.939 ± 0.005	0.979 ± 0.001	0.923 ± 0.003	0.941 ± 0.005
SPACPI(w/o SPA attention)	0.979 ± 0.004	0.919 ± 0.004	0.961 ± 0.004	0.987 ± 0.002	0.946 ± 0.003	0.949 ± 0.005
SPACPI	**0.984 ± 0.002**	**0.921 ± 0.004**	**0.964 ± 0.003**	**0.991 ± 0.002**	**0.957 ± 0.003**	**0.965 ± 0.006**

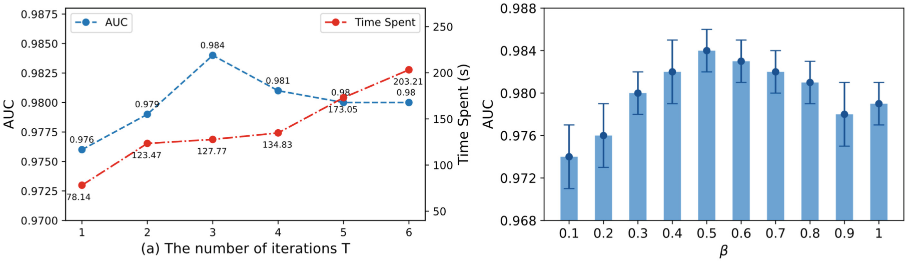
Table 3. Comparative performance of different methods in GPCR, Kinase datasets.

	GPCR		Kinase	
	AUC	AUPR	AUC	AUPR
CPI-GNN	0.490	0.524	0.434	0.173
GCN	0.820	0.809	0.447	0.186
GraphDTA	0.817	0.817	0.421	0.184
TransformerCPI	0.855	0.836	0.571	0.284
HyperAttentionDTI	0.825	0.827	0.488	0.311
CPGL	0.839	0.824	0.690	0.320
SPACPI(w/o fingerprint)	0.834	0.849	0.590	0.319
SPACPI(w/o SPA attention)	0.784	0.779	0.696	0.349
SPACPI	**0.857**	**0.862**	**0.724**	**0.416**

3.4 Impacts of Modules and Parameters

In Table 2 and Table 3, we also compare SPACPI with two variants: (1) w/o fingerprints and (2) w/o the sparse Perturbation-Aware mechanism (w/o SPA attention). Based on the outcomes, we have the following observations: (1) SPACPI can benefit from the integration of auxiliary molecular fingerprints; (2) Removing perturbation-aware attention mechanism decreases SPACPI's performance, which suggests that the sparsified, perturbation-aware attention weights are effective in predicting CPI.

Furthermore, we compare the performances under different values of the number of iterations T of perturbation attention and sparse radio β on the Human dataset. Firstly, we optimize ε_i^c within each epoch with a finite number of iterations T. We set the number of iterations T from 1 to 6 and compare the experimental results as shown in Fig. 3(a). We find that the optimal number of iterations T is 3. To achieve the highest AUC value, T is insufficient for the model to be guided by the perturbation, while a large T will significantly increase the time consumed to train one epoch. Nonetheless, even with $T = 1$, SPACPI beats recent competitors such as GraphDTA and TransformerCPI. A smaller number of iterations When T is too small, the model is unable to learn appropriate perturbation attention to guide the original attention; when T is too large, the time consumed to train one epoch increases significantly. Secondly, sparse radio β determines the number of top $K = n \times \beta$ atoms. We change β in the range [0.1, 1.0] and report the performances in Fig. 3(b). The best performance is obtained when $\beta = 0.5$. Retaining too many features (i.e., $\beta > 0.8$) or removing too many features (i.e., $\beta < 0.2$) will degrade the performance.



(a) AUC and Time Spent w.r.t. T on Human dataset. **(b) AUC w.r.t. β on Human dataset.**

Fig. 3. AUC w.r.t. T and β on Human dataset.

3.5 Case Study

To exemplify the effectiveness of SPACPI, we visualize the molecular graphs of Cyproheptadine, Cyclobenzaprine and Citalopram in Fig. 4**.** The visualization of attention weights of the compounds, Cyproheptadine, Cyclobenzaprine and Citalopram. The highlighted atoms are the areas that have a more significant impact on the CPI interaction prediction. We can see that, before perturbation-aware attention mechanism, the highlighted atoms (i.e., with large attention weights) are concentrated in areas with similar structures between the three molecules. The labels beneath each molecular graph represent their *actual label/predicted label* for whether they interact with Hismanine H1 receptor, for example, *pos/pos* indicates that the actual label is positive, and the predicted label is also positive. For molecules Cyclobenzaprine and Citalopram, which do not interact with the Hismanine H1 receptor, the model focuses on the similar key atoms as in Cyproheptadine, resulting in them being mistankenly assigned the same predicted label as Cyproheptadine. After the attention enhancement, the highlighted atoms tend to be distributed in areas with differences. This showcases the model's ability to capture genuine interaction features and identify key atoms involved in compound-protein interactions.

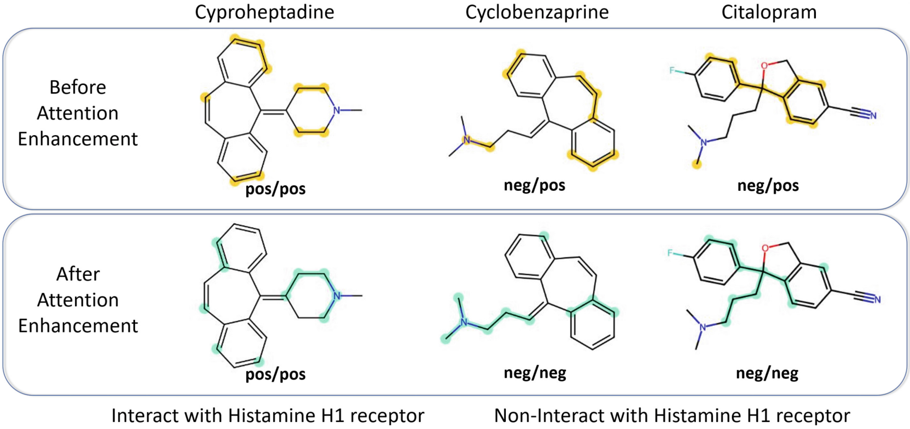

Cyproheptadine
Cyclobenzaprine
Citalopram
Before Attention Enhancement
pos/pos
neg/pos
neg/pos
After Attention Enhancement
pos/pos
neg/neg
neg/neg
Interact with Histamine H1 receptor
Non-Interact with Histamine H1 receptor

Fig. 4. The visualization of attention weights of the compounds, Cyproheptadine, Cyclobenzaprine and Citalopram. The highlighted atoms are the areas that have a more significant impact on the CPI interaction prediction.

4 Related Work

Most related work is based on the attention mechanism. TransformerCPI [1] is the first work that adapts the Transformer architecture with a self-attention mechanism to address sequence-based CPI classification tasks. DrugBAN [2] applies a bilinear attention network to explicitly learn CPI. CPGL [4] adopts a two-sided attention mechanism to give different weights to different parts of the compound and protein. MolTrans [11] adopts dot-product attention to measure the pair's interaction. HyperAttentionDTI [8] designs a hyperAttention module to generate an attention matrix. PerceiverCPI [3] uses the cross-attention mechanism to model the semantic relevance between the protein and compound.

5 Conclusion

This paper proposes SPACPI to predict compound-protein interaction, which includes three novel strategies: (1) molecular fingerprints are integrated to supplement molecular graphs, (2) atom importance is determined by learning each atom's tolerance to different perturbation amplitudes, (3) redundant features are filtered by sparsifying attention scores. Extensive experimental results on two benchmark datasets and two label-reversal datasets verify the effectiveness of SPACPI.

References

1. Chen, L., et al.: TransformerCPI: improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. Bioinform. **36**(16), 4406–4414 (2020)
2. Bai, P., Miljković, F., John, B., Lu, H.: Interpretable bilinear attention network with domain adaptation improves drug–target prediction. Nat. Mach. Intell. **5**(2), 126–136 (2023)
3. Nguyen, N.-Q., Jang, G., Kim, H., Kang, J.: Perceiver CPI: a nested cross-attention network for compound–protein interaction prediction. Bioinformatics **39**(1), btac731 (2023)
4. Zhao, M., Yuan, M., Yang, Y., Xu, S.X.: CPGL: prediction of compound-protein interaction by integrating graph attention network with long short-term memory neural network. IEEE/ACM Trans. Comput. Biol. Bioinf.Comput. Biol. Bioinf. **20**(3), 1935–1942 (2023)
5. Lee, I., Keum, J., Nam, H.: DeepConv-DTI: prediction of drug-target interactions via deep learning with convolution on protein sequences. PLoS Comput. Biol. Comput. Biol. **15**(6), e1007129 (2019)
6. Öztürk, H., Ozkirimli, E., Özgür, A.: WideDTA: prediction of drug-target binding affinity. arXiv preprint arXiv:1902.04166 (2019)
7. Schenone, M., Dančík, V., Wagner, B.K., Clemons, P.A.: Target identification and mechanism of action in chemical biology and drug discovery. Nat. Chem. Biol. **9**(4), 232–240 (2013)
8. Zhao, Q., Zhao, H., Zheng, K., Wang, J.: HyperAttentionDTI: improving drug–protein interaction prediction by sequence-based deep learning with attention mechanism. Bioinformatics **38**(3), 655–662 (2022)
9. Zheng, S., Li, Y., Chen, S., Xu, J., Yang, Y.: Predicting drug–protein interaction using quasi-visual question answering system. Nat. Mach. Intell. **2**(2), 134–140 (2020)
10. Huang, L., et al.: CoaDTI: multi-modal co-attention based framework for drug–target interaction annotation. Brief. Bioinform. **23**(6), bbac446 (2022)
11. Huang, K., Xiao, C., Glass, L.M., Sun, J.: MolTrans: molecular interaction transformer for drug–target interaction prediction. Bioinformatics **37**(6), 830–836 (2021)
12. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Graph attention networks. stat **1050**(20), 10–48550 (2017)
13. Landrum, G., Sforna, G., Winter, H.D., deric4: RDKit: open-source cheminformatics (2006). https://github.com/rdkit/rdkit
14. Shen, W.X., et al.: Out-of-the-box deep learning prediction of pharmaceutical properties by broadly learned knowledge-based molecular representations. Nat. Mach. Intell. **3**(4), 334–343 (2021)
15. Durant, J.L., Leland, B.A., Henry, D.R., Nourse, J.G.: Reoptimization of MDL keys for use in drug discovery. J. Chem. Inf. Comput. Sci.Comput. Sci. **42**(6), 1273–1280 (2002)
16. Stiefl, N., Watson, I.A., Baumann, K., Zaliani, A.: ErG: 2D pharmacophore descriptions for scaffold hopping. J. Chem. Inf. Model. **46**(1), 208–220 (2006)
17. Chen, Q., Zhang, M., Xue, B.: Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. IEEE Trans. Evol. Comput.Evol. Comput. **21**(5), 792–806 (2017)
18. Jiao, R., Nguyen, B.H., Xue, B., Zhang, M.: A survey on evolutionary multiobjective feature selection in classification: approaches, applications, and challenges. IEEE Trans. Evol. Comput. (2023)
19. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
20. Tsubaki, M., Tomii, K., Sese, J.: Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. Bioinformatics **35**(2), 309–318 (2019)

21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017)
22. Nguyen, T., Le, H., Quinn, T.P., Nguyen, T., Le, T.D., Venkatesh, S.: GraphDTA: predicting drug–target binding affinity with graph neural networks. Bioinformatics **37**(8), 1140–1147 (2021)
23. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2017)