Building explainable graph neural network by sparse learning for the drug-protein binding prediction

Anonymous Author(s)
Affiliation, City, State, Country
email

ABSTRACT

Explainable Graph Neural Networks (GNNs) have been developed and applied to drug-protein binding prediction to identify the key chemical structures in a drug that have active interactions with the target proteins. However, the key structures identified by the current explainable GNN models are typically chemically invalid. Furthermore, a threshold needs to be manually selected to pinpoint the key structures from the rest. To overcome the limitations of the current explainable GNN models, we propose our SLGNN, which stands for using Sparse Learning to Graph Neural Networks. Our SLGNN relies on using a chemical-substructure-based graph (where nodes are chemical substructures) to represent a drug molecule. Furthermore, SLGNN incorporates generalized fussed lasso with message-passing algorithms to identify connected subgraphs that are critical for the drug-protein binding prediction. Due to the use of the chemical-substructure-based graph, it is guaranteed that any subgraphs in a drug identified by our SLGNN are chemically valid structures. These structures can be further interpreted as the key chemical structures for the drug to bind to the target protein. We demonstrate the explanatory power of our SLGNN by first showing all the key structures identified by our SLGNN are chemically valid. In addition, we illustrate that the key structures identified by our SLGNN have more predictive power than the key structures identified by the competing methods. At last, we use known drug-protein binding data to show the key structures identified by our SLGNN contains most of the binding sites.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Supervised learning by classification*; *Neural networks*.

KEYWORDS

Graph Neural Networks, Interpretable models, Sparse learning, Drug-protein binding prediction.

1 INTRODUCTION

In recent years, the development of the Graph Neural Networks (GNNs) has revolutionized the drug development process, which includes the determination of molecular properties [6, 11, 24, 25], protein–ligand docking [3, 12, 19, 20], drug discovery and optimization [7, 14, 22, 31], etc. GNNs offer several advantages over traditional machine learning models in handling irregular data and integrating local and global information on graph data, which is the key to learning with molecular data. The outperformance of GNNs over traditional methods would solidify their position as leading methods in drug discovery and molecular chemistry in the following years.

In this paper, we focus on the GNN models that are built for predicting drug-protein bindings. Many GNN models [8, 10, 26, 29] have been developed for the drug-protein binding prediction task. They have demonstrated their prediction accuracy in different metrics for various tasks.

From the drug development point of view, other than the accuracy of the drug-protein binding prediction, we are also interested in knowing which molecular substructures of the candidate drugs have active interactions with the target proteins. In order to provide such information, many explainable GNN models have been designed [13, 32, 34, 35]. XGNN [35] proposes a reinforcement learning based method that utilizes information from the trained GNNs to train a graph generator to explain the GNN. This generator iteratively adds edges to the generated subgraphs to maximize the model's prediction. In the end, the generated subgraphs are subsequently considered as the explanation yielded by the XGNN model. GNNExplainer [34] and PGExplainer [13] are also developed to capture the critical substructures in drug molecules, their fundamental ideas for interpretation are similar by utilizing mask matrix which is learned by optimizing the mutual information of prediction and subgraphs' distribution, where entries in the matrix represent nodes' importance. At last, AttentiveFP [32] uses graph attention to prioritize the importance of the chemical atoms in a molecule.

However, there are limitations in the explainable GNN models mentioned above. First of all, all the current explainable GNN models [13, 32, 34, 36] use atoms-based graphs, where nodes are atoms and edges are bonds between atoms. Although using atoms-based graphs is intuitive and effective for the learning tasks, it might be problematic when we try to interpret their explanatory results. For example, the explainable GNN models might identify specific nodes (atoms) and edges (bonds) that are important for the active interaction between the drug and the target protein. However, these
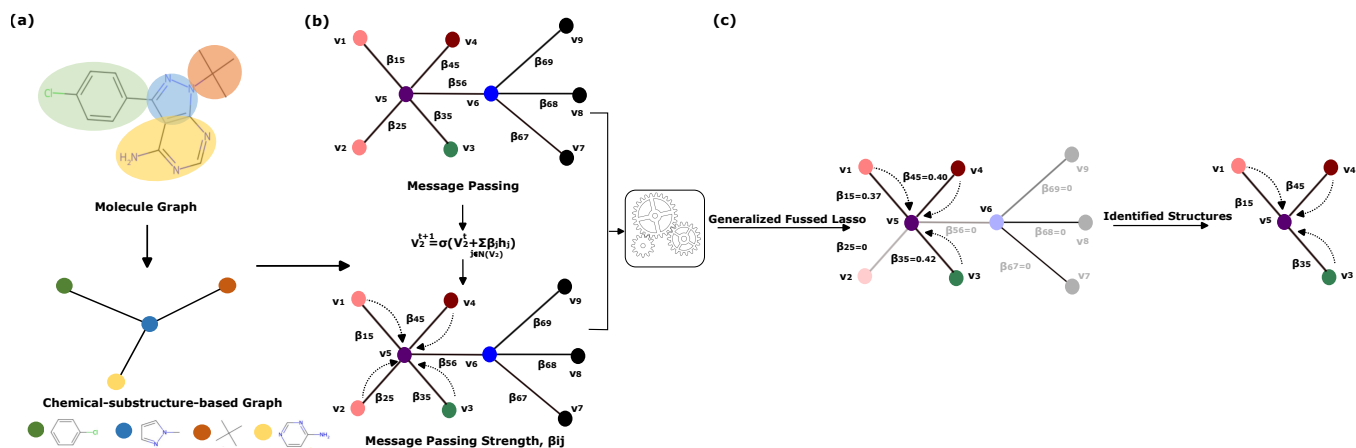
Figure 1: The overview workflow of SLGNN. (a) Convert a drug molecule into a chemical-substructure-based graph. (b) Message passing in the GNN model. Here we introduce β_{ij} to present the message passing strength. (c) Applying generalized fussed lasso on β_{ij} to learn a connected subgraph in the GNN model.

identified nodes and edges might not be connected. Furthermore, even if they are connected, they might not be all valid chemical substructures. Therefore, the explanatory results generated by the current explainable GNN models are not easy to interpret. Secondly, current explainable GNN models are rank-based methods. They use different techniques to weigh the importance of the node (AttentiveFP [32]) or the edge (GNNExplainer [34] and PGExplainer [13]) in a graph. Therefore, they need a cutoff to separate the important nodes or edges from the less important ones. However, for many real-world tasks, how to select an appropriate cutoff remains unknown.

To overcome the above limitations, we proposed an explainable GNN model named SLGNN. Rather than using the atom-based graphs, our SLGNN relies on the chemical-substructures-based graphs, where nodes are valid chemical substructures and edges are their interactions. The key advantage of the chemical-substructures-based graphs over the atom-based graphs is that any subgraph in a chemical-substructures-based graph is chemically valid. SLGNN applies the message passing algorithm on chemical-substructures-based graphs. In order to identify the key chemical substructures with a drug and make sure they are connected, we incorporate the generalized fused lasso into the message passing algorithm. Therefore, after training our SLGNN, we can identify the key connected chemical substructures without any threshold.

Additionally, the aim of this work is to find drug substructures that play critical roles in binding to a certain protein. This is important since in drug discovery tasks if we have a specific protein as our target, with the help of our method, we can find these critical substructures and provides insight into how to design new drugs with such substructures. As a result, in our method, it is not necessary to include protein learning for only one protein we focused on. Thus, we don't include the protein learning part in the scope of this work.

We evaluate our interpretation power of SLGNN using real-world data. We extract drug-protein binding data from the ChEMBL [4] database. We focus on three proteins that are Lyn, Lck, and Src. We first show that the key structures identified by our SLGNN for

each drug are chemically valid. In contrast, the competing methods cannot guarantee the key structures they identified are all chemically valid. Furthermore, we demonstrate that the key structures identified by our SLGNN has more predictive power than the key structures identified by the competing methods. Last but not least, we conduct case studies. We use the known drug-protein data to check whether the identified key structures contain the binding sites. We find that our SLGNN outperforms the competing methods for most of the cases.

2 METHODS

The overview of our SLGNN method is illustrated in Fig. 1. Overall, SLGNN incorporates sparse learning techniques into the message passing algorithm in GNN to identify essential connected chemical substructures in a drug for binding to a target protein. SLGNN consists of two major steps. The first step (as shown in Fig. 1) is to represent a drug using a chemical-substructure-based graph, where each node in the graph denotes a chemical substructure, as shown in Fig. 1(a). Compared to the widely used atom-based group [26, 32, 36] (nodes in the graph are chemical atoms), the chemical-substructure-based graph ensures that SLGNN identifies connected chemical substructures rather than connected atoms, which might not be chemically valid. More details about this step will be elaborated in section 2.1. The second step (as shown in Fig. 1(b-c)) applies the generalized fused lasso learning method to the message passing algorithm in GNN to pinpoint subgraphs in the chemical-substructure-based graph. Computationally, SLGNN uses those subgraphs to determine whether a drug would bind or not bind to the target protein. Therefore, the subgraphs in a drug identified by SLGNN can be used to explain why the drug bind or does not bind to the target protein. The details of the second step will be described in section 2.3.

2.1 Building the chemical-substructure-based graphs

Our SLGNN relies on using the chemical-substructure-based graph. The advantage of using the chemical-substructure-based graph over
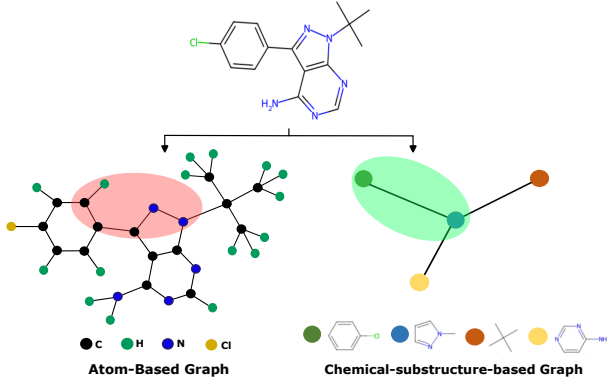
Figure 2: Difference between atom-based graph and chemical-substructure-based graphs. A subgraph in the atom-based graph might not be chemically valid (as shown in the red shade). In contrast, any subgraph in the chemical-substructure-based Graph is chemically valid (green shade is an example).

the atom-based graph (widely used in previous work [26, 32, 36]) is that any subgraph in the chemical-substructure-based graph is chemically valid. In contrast, any subgraph in the atom-based graph might not be chemically valid, which is the key limitation of previous explainable methods [26, 32, 36]. Fig. 2 illustrates the advantage of using the chemical-substructure-based graph.

We use the junction tree algorithm [9] to convert a set of drugs $\mathcal{D}$ used in the learning task to a set of corresponding chemical-substructure-based graphs $\mathcal{G}$. Given a drug $D \in \mathcal{D}$ and its corresponding chemical-substructure-based graph $G = (V, E) \in \mathcal{G}$, we know V is the set of chemical substructures appeared in drug D and E is the edges set including the connectivity between the chemical substructures in drug D. In short, the junction tree algorithm first screen all drugs in $\mathcal{D}$ to construct a vocabulary $\mathcal{V}$ of all chemically valid substructures. Then any drug $D \in \mathcal{D}$ is represented by a graph $\hat{G}$, where nodes in $\hat{G}$ are valid chemical substructures in $\mathcal{V}$ and edges are weighted by the overlaps between the valid chemical substructures. In the end, $G \in \mathcal{G}$ is obtained by finding the maximum weighted spanning tree on $\hat{G}$. For more details of the junction tree algorithm, we refer the audience to [9].

2.2 Message passing in GNN

2.2.1 Input: vertex embedding for the chemical-substructure-based graphs.
For a drug $D \in \mathcal{D}$ and its corresponding chemical-substructure-based graph $G = (V, E) \in \mathcal{G}$, $v_i \in V$ is the ith chemical substructure in D and $e_{ij} \in E$ represents the direct connection between the ith chemical substructure and the jth chemical substructure. $\mathcal{V}$ is the vocabulary of all chemically valid substructures appearing in graphs in $\mathcal{G}$.

In our GNN model, we only consider vertex embedding. We assign a d-dimensional embedding vector for each chemically valid substructure in $\mathcal{V}$. For simplicity, we use the vertex embedding $\mathbf{v}_i = \mathbf{v}_i^0 \in \mathbb{R}^d$ to describe the message passing procedure in our GNN model in the following.

2.2.2 Message passing function in GNN.
Given a graph $G = (V, E)$, we first randomly initialize the embedding for its vertices. Then we can compute $\mathbf{v}_i^{(t)} \in \mathbb{R}^d$, the ith vertex embedding at time step t, by the following message passing function:

$$\mathbf{v}_i^{(t+1)} = \sigma\left(\mathbf{v}_i^{(t)} + \sum_{j \in \mathcal{N}(i)} \beta_{ij}\mathbf{h}_j^{(t)}\right), \tag{1}$$

where σ is the element-wise activation function (here we use the sigmoid function), $\mathcal{N}(i)$ is the set of neighboring indices of the ith vertex, and $\mathbf{h}_j^{(t)} \in \mathbb{R}^d$ is the hidden neighborhood vector. The hidden vector can be computed using the vertex embedding $\mathbf{v}_j^{(t)}$ using the following neural network:

$$\mathbf{h}_i^{(t)} = f\left(\mathbf{W}_n\mathbf{v}_j^{(t)}\right), \tag{2}$$

where f is a non-linear activation function and $\mathbf{W}_n \in \mathbb{R}^{d \times d}$ is a learnable weight matrix.

Notably, we introduce a set of leanable parameter $\boldsymbol{\beta} = \{\beta_{ij}|(i, j) \in E\}$ in the message passing function (1), which has not been used by the traditional GNN model [21]. We require $\beta_{ij} = \beta_{ji}$. The goal of introducing β_{ij} is to quantify the contribution made by the jth neighbor to the embedding of the ith vertex. The larger the magnitude of β_{ij}, the more contribution the jth neighbor makes to the embedding of vertex i. More importantly, $\beta_{ij} = 0$ indicates that the jth neighbor makes no contribution to the embedding of vertex i, meaning in the message passing (1), the edge e_{ij} can be ignored. Therefore, by enforcing $\beta_{ij} = 0$, we could remove less important edges in G in the GNN message passing. Then the structure of G would become sparse. And such sparse structures can be interpreted as the key chemical structures for deciding the binding of drugs. We will introduce how to learn such chemical valid sparse structures in drugs in section 2.3.

2.2.3 Output: drug binding prediction.
After going through the message passing function, the chemical-substructure-based graph G will be represented by a feature vector $\mathbf{y}_{\text{mol}} \in \mathbb{R}^d$ from the set of vertex embedding vector, i.e. $V = \{\mathbf{v}_1^{(t)}, \mathbf{v}_2^{(t)}, ..., \mathbf{v}_{|V|}^{(t)}\}$. Mathematically, $\mathbf{y}_{\text{mol}} \in \mathbb{R}^d$ can be computed by

$$\mathbf{y}_{\text{mol}} = \frac{1}{|V|}\sum_{i=1}^{|V|} w_i\mathbf{v}_i^{(t)}, \tag{3}$$

where w_i is a learnable parameter for the ith vertex embedding. We set $w_i = 0$ if $\beta_{ij} = 0, \forall j \in \mathcal{N}(i)$, meaning we neglect $\mathbf{v}_i$ in (3) if all its neighbors ($\beta_{ij} = 0, \forall j \in \mathcal{N}(i)$) does not pass message to it in (1).

The final output of the GNN model can be computed by applying a fully connected single-layer neural network to $\mathbf{y}_{\text{mol}}$.

$$p = \sigma\left(\mathbf{W}_p\mathbf{y}_{mol} + \mathbf{b}\right), \tag{4}$$

where σ is the sigmoid function: $\sigma(x) = 1/(1 + e^{-x})$, p is the probability of binding, and $\mathbf{W}_p$ and $\mathbf{b}$ are parameters for the single layer neural network.

2.2.4 Learning objective.
The learning objective of the GNN model introduced above can be written as the cross entropy loss with the weight decay:

$$\min_{\boldsymbol{\beta},\Theta} : \mathcal{L}(\boldsymbol{\beta}, \Theta) + \eta\|\Theta\|^2, \tag{5}$$

where $\mathcal{L}(\beta, \Theta) = \sum_i y_i \log(p_i)$ ($y_i \in \{0, 1\}$ is the label and p_i is the prediction from (4)) is the cross entropy loss parameterized by β and Θ. $\boldsymbol{\beta} = \{\beta_{ij} | (i, j) \in E\}$ is the collection of β_{ij} introduced in (1), and Θ is the set of the rest of model parameters other than $\boldsymbol{\beta}$ in our GNN model, such as $\mathbf{W}_n$, $\mathbf{W}_p$, etc.

2.3 Identify key chemical structures in a drug using generalized fussed lasso

Our GNN model can be trained by solving (5). However, learning the GNN model from (5) cannot provide the information about which chemical structures in a drug is critical for binding to the target protein. Therefore, in this section, we introduce how we use the generalized fussed lasso to identify the key chemical structures in a drug for binding.

Thanks to the introduction of β_{ij} in (1), we are able to measure the message passing strength between vertices i and j. We require $\beta_{ij} = \beta_{ji}$ to impose the message passing strength from i to j equals to the strength from j to i. We observe that $\beta_{ij} = 0$ denotes that in the message passing function (1) the connection between vertices i and j has not been used, which implies that the connection between vertices i and j in graph $G = (V, E)$ is not important in our GNN model. Based on the observation, we propose to apply generalized fussed lasso to β_{ij}. Then the learning objective of our GNN model becomes

$$\min_{\boldsymbol{\beta}, \Theta} : \mathcal{L}(\boldsymbol{\beta}, \Theta) + \eta\|\Theta\|^2 + \Omega(\boldsymbol{\beta}), \tag{6}$$

where $\Omega(\boldsymbol{\beta})$ is the the generalized fussed lasso term that applies to $\boldsymbol{\beta}$. Mathematically, $\Omega(\boldsymbol{\beta})$ is defined as

$$\Omega(\boldsymbol{\beta}) := \lambda_1 \sum_{(i,j) \in E} |\beta_{ij}| + \lambda_2 \sum_{\substack{(i,j) \in E, \\ (j,k) \in E}} |\beta_{ij} - \beta_{jk}|. \tag{7}$$

The $\sum_{i,j} |\beta_{ij}|$ term aims to remove less important edges in G by enforcing β_{ij} to be zero. The $\sum_{(i,j),(j,k) \in E} |\beta_{ij} - \beta_{jk}|$ term promotes smoothness between β_{ij} and β_{jk} if they have a share vertex j. This term makes sure that after removing edges from G, the remaining edges can still form connected subgraphs in G. The effect of the generalized fussed lasso term is illustrated in Fig. 1(c).

For a vertex i, if all $\beta_{i,j} = 0, \forall (i, j) \in E$, we will remove its vertex embedding in (3) by setting the corresponding $w_i = 0$.

2.4 Prioritizing shared structures in positive samples

In positive samples (drugs that are known to bind to the target protein), if certain connections between the same chemical substructures frequently appear, it suggests that such connections are important for the positive samples. We would enforce such connections to be preserved rather than removed by our GNN model. Therefore, we add additional terms to our learning objective.

$$\min_{\boldsymbol{\beta}, \Theta} : \mathcal{L}(\boldsymbol{\beta}, \Theta) + \eta\|\Theta\|^2 + \Omega(\boldsymbol{\beta}) + \Gamma(\boldsymbol{\beta}), \tag{8}$$

where $\Gamma(\boldsymbol{\beta})$ is defined as

$$\Gamma(\boldsymbol{\beta}) := -\lambda_3 \sum_{(i,j) \in E} m_{ij} \beta_{ij}^2. \tag{9}$$

m_{ij} is the frequency of the connections between chemical substructure i and chemical substructure j appeared in the positive samples.

We note that the larger m_{ij} is, the more penalty is applied to β_{ij} to make it away from zero. Therefore, adding $\Gamma(\boldsymbol{\beta})$ in (8) would preserve connections between chemical substructures that frequently appeared in the positive samples.

2.5 Optimization

The final learning objective of our GNN model is written in (8). We first rearrange the terms in (8) and rewrite it as

$$\min_{\boldsymbol{\beta}, \Theta} : \mathcal{F}(\boldsymbol{\beta}, \Theta) + \eta\|\Theta\|^2 + \Omega(\boldsymbol{\beta}), \tag{10}$$

where $\mathcal{F}(\boldsymbol{\beta}, \Theta) = \mathcal{L}(\boldsymbol{\beta}, \Theta) + \Gamma(\boldsymbol{\beta})$ is a smooth function for both $\boldsymbol{\beta}$ and Θ. The reason we write our final learning objective as (10) is that it is in the format for the proximal alternative linearized minimization (PALM) algorithm [2]. The PALM algorithm applies the proximal forward–backward algorithm [2] to optimize both $\boldsymbol{\beta}$ and Θ in an alternative manner. Specifically, at iteration k, the proximal forward-backward mappings of $\eta\|\Theta\|^2$ and $\Omega(\boldsymbol{\beta})$ for given Θ^k and $\boldsymbol{\beta}^k$ are

$$\Theta^{k+1} \in \min_{\Theta} : \left\{ \frac{a^k}{2} \left\| \Theta - U^k \right\|_F^2 + \eta\|\Theta\|^2 \right\}; \tag{11a}$$

$$\boldsymbol{\beta}^{k+1} \in \min_{\boldsymbol{\beta}} : \left\{ \frac{b^k}{2} \left\| \boldsymbol{\beta} - V^k \right\|_F^2 + \Omega(\boldsymbol{\beta}) \right\}, \tag{11b}$$

where $U^k = \Theta^k - \frac{1}{a^k} \nabla_\Theta \mathcal{F}(\Theta^k, \boldsymbol{\beta}^k)$ and $V^k = \boldsymbol{\beta}^k - \frac{1}{b^k} \nabla_{\boldsymbol{\beta}} F(\Theta^{k+1}, \boldsymbol{\beta}^k)$. a^k and b^k are positive real numbers and $\nabla_\Theta \mathcal{F}(\Theta^k, \boldsymbol{\beta}^k)$ is the derivative of $\mathcal{F}(\Theta, \boldsymbol{\beta}^k)$ with respect to Θ at point Θ^k for fixed $\boldsymbol{\beta}^k$ and $\nabla_{\boldsymbol{\beta}} \mathcal{F}(\Theta^{k+1}, \boldsymbol{\beta}^k)$ is the derivative of $\mathcal{F}(\Theta^{k+1}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ at point $\boldsymbol{\beta}^k$ for fixed Θ^{k+1}.

For the first proximal forward–backward mapping (11a), we can easily find the closed-form solution (leave the derivation to the audience). For the second proximal forward–backward mapping (11b), we use the algorithm proposed in [30] to find its optimal solution. We iteratively solve (11a) and (11b) to obtain a sequence $\{(\Theta^k, \boldsymbol{\beta}^k)\}_{k \in \mathbb{N}}$. It has been proven that the sequence $\{(\Theta^k, \boldsymbol{\beta}^k)\}_{k \in \mathbb{N}}$ generated by PALM converges to a critical point [2].

2.6 Related works

We briefly discuss some current efforts for constructing explainable GNN models. Despite using different computational strategies, the following methods can be categorized into one kind: the weight ranking-based interpretation method. They [13, 32, 34] have been tested on BBBP, MUTAG, Tox21, synthetic datasets, etc.

AttentiveFP [32] stacks Graph Attention (GAT) layer [23] on message passing neural network. Its interpretation comes from the attention weight of the GAT layer. GNNExplainer [34] and PG-Explainer [34] explain the model using a learnable binary feature selector. After training, the interpretation can be obtained by analyzing the values in the binary feature selector (values toward 1 means more important).

The output of the current explainable GNN models is the ranking of the atoms in drug molecules. Therefore, a cutoff is needed to distinguish between important atoms and less important atoms. For tasks like aromatic detection, we can easily select the cutoff [32]. However, for the drug-protein binding task that we focus on in this

(a) SLGNN **(b) AttentiveFP** **(c) GNNExplainer** **(d) PGExplainer**
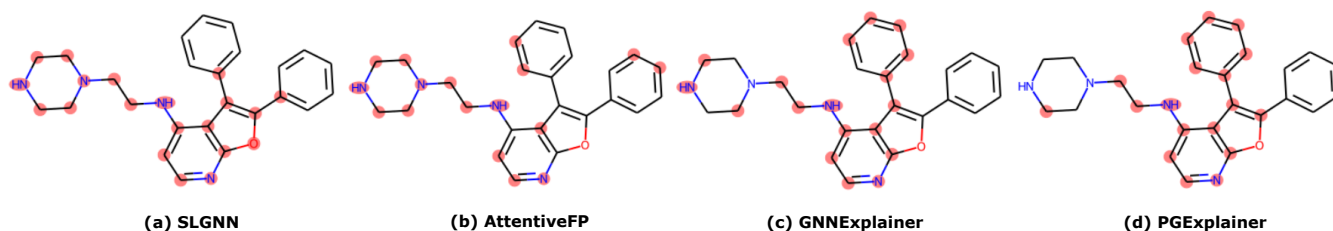
Figure 3: The key structures identified by different methods on 2OF2(PDB ID). (a) The atoms in the structure identified by SLGNN are connected and chemically valid. (b) The atoms selected by AttentiveFP are not connected. (c) The atoms selected by GNNExplainer are not connected. (d) The atoms selected by PGExplainer are not connected.

paper, it is very challenging to pick the cutoff because how many substructures in drug molecules play critical roles in forming the bindings is unknown before chemical experiments.

3 EXPERIMENTAL RESULTS

In this section, we evaluate our SLGNN and the competing methods on the drug-protein binding task, which categorize drugs into binding and not binding classes for the target protein. We use SLGNN and the competing methods to build binary classifiers for Lyn, Lck, and Src proteins. We focus on comparing the explanatory power of the competing methods. We first check the predictive power of the key structures identified by the competing methods. Last but not least, for some known drug-protein bindings, we check whether the chemical substructures identified by the competing methods contain known binding sites.

3.1 Datasets

We extract drugs/molecules and their binding information for proteins Lyn, Lck, and Src from the ChEMBL database [4]. We use the datasets of these three proteins in our experiments. We use Ki values provided in ChEMBL to label the drugs/molecules. We consider a drug/molecule with Ki < 1000 nm as a binding class and a drug/molecule with Ki ≥ 1000 nm as a not binding class. In sum, for protein Lyn, we extract 649 drugs/molecules, where there are 219 positive samples (binders) and 430 negative samples (non-binders). For protein Lck and Src, we extract data with the same method and get 239 drugs/molecules (93 positive samples and 146 negative samples) and 499 drugs/molecules (218 positive samples and 281 negative samples), respectively.

3.2 Competing methods

We consider the drug-protein binding prediction task as a binary classification task. We aim to build models to categorize drugs into binding and not binding classes. Therefore, we use all the competing methods to build binary classifiers.

The competing methods we consider include our SLGNNand three state-of-the-art methods: AttentiveFP [32], GNNExplainer [34] and PGExplainer [13]. AttentiveFP is a graph attention-based method whose interpretation is derived from attention weights. GNNExplainer is a model-agnostic interpretation method based on mean field variational approximation. Given a trained GNN model, it will

generate a graph mask for the explanation of this instance. PGExplainer is built based on a generative probabilistic model, which generates model explanations for new instances without learning from scratch. We use all the models to build binary classifiers for the drug-protein binding task.

3.3 Hyper-parameters

For AttentiveFP [32], GNNExplainer [34] and PGExplainer [13], we use the hyper-parameters suggested in the orginal papers. For our SLGNN, we set $\lambda_1 = 1$, $\lambda_2 = 9$, and $\lambda_3 = 0.01$ for all experiments.

3.4 Select the key chemical structures

For identification of the chemical structures in drugs, our SLGNN can directly output the key chemical substructures in a drug. However, AttentiveFP [32], GNNExplainer [34] and PGExplainer [13] only generate the rank of the atoms in a drug. To make sure the key structures identified by different methods are comparable, for each drug, we count the number of atoms in the output of SLGNN and use the ranks generated in AttentiveFP [32], GNNExplainer [34] and PGExplainer [13] to pick the same number of atoms in the same drug. Therefore, for any drug used in the learning task, the structures identified by all the competing methods have the same number of atoms.

3.5 Evaluation of whether the identified structures are chemically valid

In this section, we check whether the structures identified by the competing methods are chemically valid.

First, we check whether the structures identified by the competing methods are connected. Fig. 3 shows an example of the structures identified by different methods for the same drug 2OF2[16] in the Lyn experiment. As illustrated, only the structure identified by our SLGNN is connected. Furthermore, to quantify the connectivity of the structures identified by the competing methods, we compute the average number of connected components (standard deviation in the parenthesis of Table. 1) in the structures identified by the competing methods in each drug. Table.1 exhibits such statistics. Clearly, our SLGNN achieves the smallest average connected components, indicating the structures identified by our SLGNN are more connected than the competing methods.

In addition, we checked whether the structures identified by the competing methods are chemically valid. Table. 1 illustrates,
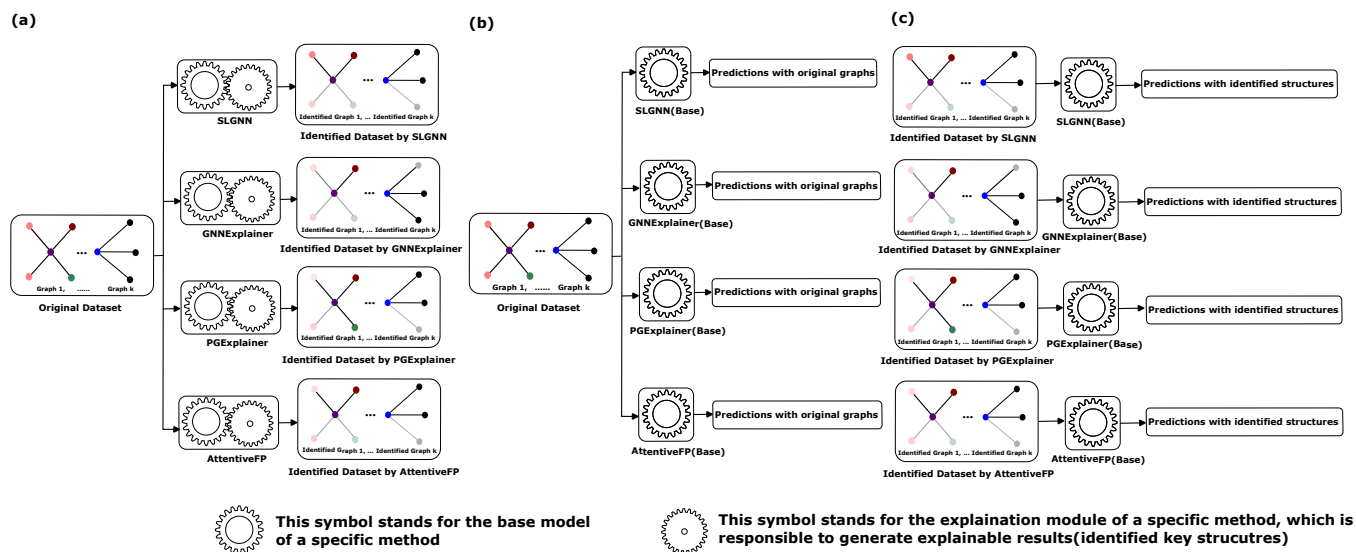
Figure 4: In this figure, we illustrate the workflow of experiments in section 3.6. In subfigure(a), we illustrate how to generate the dataset with identified structures for each competing method. In subfigure(b), we introduce how to generate the results for predictions on the original dataset, which correspond to the results on the left panel of Table 2. In subfigure(c), we illustrate how to generate the results for predictions on the identified datasets, which correspond to the results on the right panel of Table 2.

for each competing method, the number of drugs whose identified structures are chemically valid. Due to the usage of the chemical-substructure-based graph, all structures identified by our SLGNN are chemically valid. However, for other competing methods, they barely identify chemically valid structures. This experiment demonstrates that the explanatory results generated by the current explainable GNN models are not chemically interpretable. In contrast, our SLGNN overcomes the problem by identifying key structures that consist of chemically valid chemical substructures.

Dataset	Model	Avg.CC	Valid Molecules
Lyn	SLGNN	1.22(0.65)	649/649
	GNNExplainer	1.61(0.53)	0/649
	PGExplainer	1.76(0.78)	0/649
	AttentiveFP	2.42(0.51)	0/649
Lck	SLGNN	1.48(0.78)	239/239
	GNNExplainer	1.59(0.67)	12/239
	PGExplainer	1.77(0.80)	13/239
	AttentiveFP	2.85(1.40)	14/239
Src	SLGNN	1.26(0.76)	499/499
	GNNExplainer	1.62(0.42)	4/499
	PGExplainer	1.74(0.64)	8/499
	AttentiveFP	2.13(0.21)	6/499

Table 1: Comparison of the average number of connected components, and molecules whose identified structures are all valid. Avg.CC stands for the average number of connected components, and the number after the slash is the total number of molecules in each dataset.

3.6 Evaluation of the predictive power of the identified structures

In this section, we evaluate the predictive power of the key structures identified by the competing methods. We check whether applying the basic message-passing algorithm on the key structures identified by the competing methods could still achieve decent learning accuracy. We use AUPR score [18] and F1 score [18] to evaluate the performance of the drug-protein binding prediction task. To help our audience have a better understanding of the procedure, we draw the workflow of this experiment in Figure. 4.

First, for each competing method, we switch off their function to generate explainable results (identify the key structures) and test their performance on the original data, which is shown in Figure. 2(b). For our SLGNN, it becomes applying the basic message passing algorithm on the chemical-substructure-based graphs and we name it SLGNN(Base). For GNNExplainer [34] and PGExplainer [13], they become applying the basic message passing algorithm on the atom-based graphs and we name them GNNExplainer(Base) and PGExplainer(Base), respectively. And for AttentiveFP [32], it becomes applying the basic message passing algorithm and the graph attention to the atom-based graphs and we name it AttentiveFP(Base). We apply SLGNN(Base), GNNExplainer(Base), PGExplainer(Base), and AttentiveFP(Base) to the original Lyn, Lck, and Src data and show their performance in Table. 2 in the left part. Obviously, our SLGNN(Base) achieves the highest learning accuracy for all three learning tasks due to the use of chemical-substructure-based graphs. Surprisingly, using chemical-substructure-based graphs can significantly improve learning accuracy over using atom-based graphs.

Original Data				Data with identified key structures			
Data Used	Model	AUPR	F1	Data Used	Model	AUPR	F1
Lyn(Original)	SLGNN(Base)	**0.969(0.045)**	**0.973(0.027)**	Lyn(SLGNN Identified)	SLGNN(Base)	**0.902(0.022)**	**0.943(0.027)**
Lyn(Original)	GNNExplainer(Base)	0.843(0.009)	0.918(0.004)	Lyn(GNNExplainer Identified)	GNNExplainer(Base)	0.733(0.021)	0.786(0.009)
Lyn(Original)	PGExplainer(Base)	0.843(0.009)	0.918(0.004)	Lyn(PGExplainer Identified)	PGExplainer(Base)	0.756(0.018)	0.799(0.0084)
Lyn(Original)	AttentiveFP(Base)	0.833(0.007)	0.863(0.008)	Lyn(AttentiveFP Identified)	AttentiveFP(Base)	0.695(0.007)	0.773(0.002)
Lck(Original)	SLGNN(Base)	**0.936(0.047)**	**0.962(0.009)**	Lck(SLGNN Identified)	SLGNN(Base)	**0.865(0.024)**	**0.911(0.033)**
Lck(Original)	GNNExplainer(Base)	0.772(0.041)	0.836(0.043)	Lck(GNNExplainer Identified)	GNNExplainer(Base)	0.700(0.029)	0.758(0.063)
Lck(Original)	PGExplainer(Base)	0.772(0.041)	0.836(0.043)	Lck(PGExplainer Identified)	PGExplainer(Base)	0.721(0.043)	0.771(0.041)
Lck(Original)	AttentiveFP(Base)	0.685(0.006)	0.7472(0.017)	Lck(AttentiveFP Identified)	AttentiveFP(Base)	0.632(0.003)	0.696(0.027)
Src(Original)	SLGNN(Base)	**0.974(0.023)**	**0.992(0.012)**	Src(SLGNN Identified)	SLGNN(Base)	**0.957(0.026)**	**0.921(0.035)**
Src(Original)	GNNExplainer(Base)	0.637(0.016)	0.693(0.007)	Src(GNNExplainer Identified)	GNNExplainer(Base)	0.525(0.019)	0.563(0.019)
Src(Original)	PGExplainer(Base)	0.637(0.016)	0.693(0.007)	Src(PGExplainer Identified)	PGExplainer(Base)	0.622(0.004)	0.643(0.032)
Src(Original)	AttentiveFP(Base)	0.794(0.010)	0.842(0.006)	Src(AttentiveFP) Identified)	AttentiveFP(Base)	0.694(0.0215)	0.733(0.006)

Table 2: The comparison of the predictive power of the key identified structures. Lyn(Original), Lck(Original), and Src(Original) are the data described in section 3.1. Lyn(SLGNN Identified), Lyn(GNNExplainer Identified), Lyn(PGExplainer Identified), and Lyn(AttentiveFP Identified) are data with the key identified structures identified by each method (described in the 3rd paragraph in section 3.6). SLGNN(Base), GNNExplainer(Base), PGExplainer(Base), and AttentiveFP(Base) are the basic models of each method that switches off the explanatory function (described in the 2nd paragraph in section 3.6).

Then, for each competing method, we switch back on their function to generate explainable results, therefore, they can identify the key structures in a drug. And this procedure is illustrated in Figure. 2(a). We apply all the competing methods to identify key structures for each drug in the original data. Then, for each protein (Lyn, Lck, and Src), we construct four new data sets where we replace the original structures of the drugs with the corresponding key structures identified by the competing methods. For example, after applying SLGNN to the Lyn data, we construct a new dataset, in which we replace the drug with the corresponding key structures of the drug identified by SLGNN and we call this new data Lyn(SLGNN Identified). Similarly, we can construct Lyn(GNNExplainer Identified), Lyn(PGExplainer Identified), and Lyn(AttentiveFP Identified), respectively. For Lck and Src proteins, we construct four new datasets as for Lyn, respectively.

Last, to test whether the identified key structures have predictive power, for each competing method, we switch off their function to generate explainable results again and apply them to the identified key structures only. This procedure is shown in Figure. 2(c). Specifically, for Lyn, we apply SLGNN(Base), GNNExplainer(Base), PG-Explainer(Base), and AttentiveFP(Base) to Lyn(SLGNN Identified), Lyn(GNNExplainer Identified), Lyn(PGExplainer Identified), and Lyn(AttentiveFP Identified), respectively. We test whether applying the basic message passing algorithm on the key structures identified by the competing methods could still achieve decent learning accuracy. Table. 2 the right part illustrates the results. As shown, applying the message-passing algorithm to the key structures identified by SLGNN could still achieve high learning accuracy (above 0.85 for AUPR and 0.9 for F1). Compared to applying the message-passing algorithm to the original data (left part of Table. 2), there is not much accuracy loss. However, when applying the message passing algorithm to the key structures identified by other competing methods, the learning accuracy has become much lower (less than 0.8 for both AUPR and F1). Therefore, the results imply that the key structures identified by SLGNN have more predictive power than the key structures identified by other competing methods.

3.7 Case studies

In this section, we use the known drug-protein binding data as the ground truth to evaluate the explanatory power of each method. The ground truth data we have included is the binding data: For Lyn, the binding between ANP and Lyn (PDB ID 2ZV8 [27]), PP2 and Lyn (PDB ID 2ZV9 [27]), 1N1 and Lyn (PDB ID 2ZVA [27]), STU and Lyn (PDB ID 3A4O [17]). For Src, NBS and Src (PDB ID 1KSW [28]), ANP and Src (PDB ID 2SRC [33]), 0J9 and Src (PDB ID 4K11 [15]). For Lck, the binding between 547 and Lck (PDB ID 2OF2 [16]), 979 and Lck (PDB ID 2OF4 [16]), 9NH and Lck (PDB ID 3B2W [5]). The binding sites between the drugs and the proteins mentioned above are calculated and collected from PLIP [1].

We check whether the chemical substructures identified by each method contain the known binding sites. Given a drug (for example ANP) and a protein (for example Lyn), we use the known binding data to classify all the chemical substructures in the drug into two classes: chemical substructures that contain at least a binding site are assigned in the positive class and chemical substructures that do not contain a binding site are assigned in the negative class. The labels of these two classes serve as our ground truth. We consider each method as a binary classifier, which identifies key structures in the drug. For each method, the chemical substructures in the identified key structures are categorized in the positive class and the rest are categorized in the negative class. We then evaluate each method by comparing its classification results against the ground truth. For each drug-protein binding pair, we will use AUPR score [18] and F1 score [18] to evaluate each competing method.

We want to mention that the key structures identified by AttentiveFP, GNNExplainer, and PGExplainer are not always valid chemical substructures. Therefore, we remove the atoms in the key structures identified by these methods if they do not form a valid chemical substructure. However, for our SLGNN, we guarantee that the identified key structures consist of valid chemical substructures.

Table. 3 illustrates the comparison between the competing methods. As shown in the left part of Table. 3, our SLGNN outperforms other methods in terms of AUPR and F1 for finding the chemical substructures that contain binding sites for protein Lyn except for the AUPR for 3A4O. The outperformance of our SLGNN denotes

Lyn				Lck				Src			
PDB ID	Model	AUPR	F1	PDB ID	Model	AUPR	F1	PDB ID	Model	AUPR	F1
2ZV8	SLGNN	**0.621**	**0.268**	2OF2	SLGNN	**0.626**	**0.667**	1KSW	SLGNN	0.612	0.211
	GNNExplainer	0.587	0.266		GNNExplainer	0.474	0.333		GNNExplainer	0.55	0.5
	PGExplainer	0.607	0.154		PGExplainer	0	0		PGExplainer	**0.766**	**0.588**
	AttentiveFP	0.565	0.143		AttentiveFP	0.530	0.4		AttentiveFP	0.633	0.153
2ZV9	SLGNN	0.482	**0.545**	2OF4	SLGNN	0.237	0.222	2SRC	SLGNN	0.572	0.222
	GNNExplainer	0.466	0.250		GNNExplainer	0	0		GNNExplainer	**0.649**	0.352
	PGExplainer	**0.599**	0.333		PGExplainer	0	0		PGExplainer	0.647	**0.444**
	AttentiveFP	0	0		AttentiveFP	**0.265**	**0.285**		AttentiveFP	0	0
2ZVA	SLGNN	**0.556**	**0.572**	3B2W	SLGNN	**0.544**	**0.286**	4K11	SLGNN	**0.624**	**0.727**
	GNNExplainer	0.536	0.222		GNNExplainer	0.530	0.273		GNNExplainer	0.45	0.224
	PGExplainer	0.474	0.200		PGExplainer	0.534	0.2		PGExplainer	0.60	0.333
	AttentiveFP	0	0		AttentiveFP	0	0		AttentiveFP	0.433	0.363
3A4O	SLGNN	0.630	**0.533**								
	GNNExplainer	**0.726**	0.200								
	PGExplainer	**0.726**	0.200								
	AttentiveFP	0.671	0.182								

Table 3: Results for case studies. The comparison on finding the chemical substructures that contain binding sites. We have 10 drug-protein binding tests.

that most of the chemical substructures identified by SLGNN bind to the protein Lyn. As shown in the middle part of Table. 3, we have the comparison on finding the chemical substructures that contain binding sites for protein Lck. For 2OF2, SLGNN outperforms other methods. For 2OF4, SLGNN and AttentiveFP have competitive performance. AttentiveFP obtains slightly better AUPR and F1 scores. For 3B2W, GNNExplainer attains the best AUPR score but SLGNN achieves the best F1 score. As shown in the right part of Table. 3, we have the comparison on finding the chemical substructures that contain binding sites for protein Src. We notice that for drug 4K11, our SLGNN outperforms other methods. However, for 1KSW and 2SRC, PGExplainer and GNNExplainer achieve better scores.

Model	AUPR	F1
SLGNN	**0.550**(0.114)	**0.425**(0.192)
GNNExplainer	0.497(0.184)	0.262(0.121)
PGExplainer	0.495(0.260)	0.245(0.175)
AttentiveFP	0.310(0.274)	0.153(0.147)

Table 4: Comparison of the average and standard deviation of AUPR and F1 scores of the 10 test cases shown in Table 3.

In sum, we remark that our SLGNN is the most robust method among the competing methods. We have 10 drug-protein binding tests in Table. 3. SLGNN achieves the best accuracy in 7 out of 10. Even for the ones SLGNN does have the best performance, its performance is close to the best ones. However, for the competing methods, they might perform well on one dataset but their overall performance is not good. As shown in Table. 4, we compute the average and standard deviation of the AUPR and F1 scores obtained on the 10 drug-protein binding tests (listed in Table. 3) for each method. Clearly, SLGNN achieves the best average AUPR and F1 scores.

4 CONCLUSION

In this paper, we propose an explainable GNN model called SLGNN. SLGNN is a GNN model built on the chemical-substructure-based graphs, where nodes represent chemically valid substructures. Using chemical-substructure-based graphs is superior to using the widely used atom-based graphs. Because any subgraph in a chemical-substructure-based graph guarantees to be chemically valid. However, the atom-based graphs do not have such nice properties. In addition, SLGNN applies generalized fussed lasso to the message-passing algorithm to identify key structures of each drug used in the GNN model. The generalized fussed lasso makes sure to the identified structures yielded by SLGNN are connected. We test SLGNN and the state-of-the-art competing methods on three real-world drug-protein binding datasets. We have demonstrated that the key structures identified by our SLGNN are chemically valid and have more predictive power.

Our SLGNN overcomes the current explainable GNN models' limitations for generating interpretation for the drug-protein binding prediction. These limitations include the explanations generated by explainable GNN models are not always chemically valid and they rely on manual selection of the cutoff threshold that is hard to pick in practice. Our SLGNN resolves the limitations. Therefore, we believe that our SLGNN has the potential to yield a meaningful explanation for drug-protein binding, which would lead to promising outcomes in drug discovery.

REFERENCES

[1] Melissa F Adasme, Katja L Linnemann, Sarah Naomi Bolz, Florian Kaiser, Sebastian Salentin, V Joachim Haupt, and Michael Schroeder. 2021. PLIP 2021: expanding the scope of the protein–ligand interaction profiler to DNA and RNA. *Nucleic Acids Research* 49, W1 (May. 2021), W530–W534. https://doi.org/10.1093/nar/gkab294

[2] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming* 146, 1-2 (Aug. 2014), 459–494. https://doi.org/10.1007/s10107-013-0701-9

[3] Hyeoncheol Cho, Eok Kyun Lee, and Insung S. Choi. 2020. Layer-wise relevance propagation of InteractionNet explains protein–ligand interactions at the atom level. *Scientific Reports* 10, 1 (Dec. 2020). https://doi.org/10.1038/s41598-020-78169-6

[4] Mark Davies, Michał Nowotka, George Papadatos, Nathan Dedman, Anna Gaulton, Francis Atkinson, Louisa Bellis, and John P Overington. 2015. ChEMBL web services: streamlining access to drug discovery data and utilities. *Nucleic Acids Res.* 43, W1 (Jul. 2015), W612–W620.

[5] Holly L. Deak, John R. Newcomb, Joseph J. Nunes, Christina Boucher, Alan C. Cheng, Erin F. DiMauro, Linda F. Epstein, Paul Gallant, Brian L. Hodous, Xin Huang, Josie H. Lee, Vinod F. Patel, Stephen Schneider, Susan M. Turci, and Xiaotian Zhu. 2008. N-(3-(Phenylcarbamoyl)arylpyrimidine)-5-carboxamides as potent and selective inhibitors of Lck: Structure, synthesis and SAR. *Bioorganic & Medicinal Chemistry Letters* 18, 3 (Feb. 2008), 1172–1176. https://doi.org/10.1016/j.bmcl.2007.11.123

[6] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. 2022. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence* 4, 2 (Feb. 2022), 127–134. https://doi.org/10.1038/s42256-021-00438-4

[7] Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy B R Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L Blundell, Michael M Bronstein, and Jake P Taylor-King. 2021. Utilizing graph machine learning within drug discovery and development. *Briefings in Bioinformatics* 22, 6 (May. 2021). https://doi.org/10.1093/bib/bbab159

[8] Huaipan Jiang, Jian Wang, Weilin Cong, Yihe Huang, Morteza Ramezani, Anup Sarma, Nikolay V. Dokholyan, Mehrdad Mahdavi, and Mahmut T. Kandemir. 2022. Predicting Protein–Ligand Docking Structure with Graph Neural Network. *Journal of Chemical Information and Modeling* 62, 12 (Jun. 2022), 2923–2932. https://doi.org/10.1021/acs.jcim.2c00127

[9] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*. PMLR, 2323–2332.

[10] Carter Knutson, Mridula Bontha, Jenna A. Bilbrey, and Neeraj Kumar. 2022. Decoding the protein–ligand interactions using parallel graph neural networks. *Scientific Reports* 12, 1 (May. 2022). https://doi.org/10.1038/s41598-022-10418-2

[11] Shuangli Li, Jingbo Zhou, Tong Xu, Dejing Dou, and Hui Xiong. 2022. GeomGCL: Geometric Graph Contrastive Learning for Molecular Property Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (Jun. 2022), 4541–4549. https://doi.org/10.1609/aaai.v36i4.20377

[12] Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. 2021. Structure-aware Interactive Graph Neural Networks for the Prediction of Protein-Ligand Binding Affinity. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*. ACM. https://doi.org/10.1145/3447548.3467311

[13] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized Explainer for Graph Neural Network. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 1646, 12 pages.

[14] Qiujie Lv, Guanxing Chen, Ziduo Yang, Weihe Zhong, and Calvin Yu-Chian Chen. 2023. Meta Learning With Graph Attention Networks for Low-Data Drug Discovery. *IEEE Transactions on Neural Networks and Learning Systems* (Mar. 2023), 1–13. https://doi.org/10.1109/tnnls.2023.3250324

[15] Thomas Madej, Christopher J. Lanczycki, Dachuan Zhang, Paul A. Thiessen, Renata C. Geer, Aron Marchler-Bauer, and Stephen H. Bryant. 2013. MMDB and VAST+: tracking structural similarities between macromolecular complexes. *Nucleic Acids Research* 42, D1 (Dec. 2013), D297–D303. https://doi.org/10.1093/nar/gkt1208

[16] Matthew W. Martin, John Newcomb, Joseph J. Nunes, Jean E. Bemis, David C. McGowan, Ryan D. White, John L. Buchanan, Erin F. DiMauro, Christina Boucher, Theodore Faust, Faye Hsieh, Xin Huang, Josie H. Lee, Stephen Schneider, Susan M. Turci, and Xiaotian Zhu. 2007. Discovery of novel 2, 3-diarylfuro[2, 3-b]pyridin-4-amines as potent and selective inhibitors of Lck: Synthesis, SAR, and pharmacokinetic properties. *Bioorganic & Medicinal Chemistry Letters* 17, 8 (Apr. 2007), 2299–2304. https://doi.org/10.1016/j.bmcl.2007.01.048

[17] Nao Miyano, Takayoshi Kinoshita, Ryoko Nakai, Yasuyuki Kirii, Koichi Yokota, and Toshiji Tada. 2009. Structural basis for the inhibitor recognition of human Lyn kinase domain. *Bioorg. Med. Chem. Lett.* 19, 23 (Dec. 2009), 6557–6560. https://doi.org/10.1016/j.bmcl.2009.10.038

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (Nov. 2011), 2825–2830. https://doi.org/10.5555/1953048.2078195

[19] Manon Réau, Nicolas Renaud, Li C Xue, and Alexandre M J J Bonvin. 2022. DeepRank-GNN: a graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics* 39, 1 (Nov. 2022). https://doi.org/10.1093/bioinformatics/btac759

[20] Chao Shen, Junjie Ding, Zhe Wang, Dongsheng Cao, Xiaoqin Ding, and Tingjun Hou. 2019. From machine learning to deep learning: Advances in scoring functions for protein–ligand docking. *WIREs Computational Molecular Science* 10, 1 (Jun. 2019). https://doi.org/10.1002/wcms.1429

[21] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. 2018. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics* 35, 2 (Jul. 2018), 309–318. https://doi.org/10.1093/bioinformatics/bty535

[22] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, and Shanrong Zhao. 2019. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery* 18, 6 (Apr. 2019), 463–477. https://doi.org/10.1038/s41573-019-0024-5

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. (2018). https://doi.org/10.48550/ARXIV.1710.10903 arXiv:1710.10903

[24] Stephen W. C. Walker, Ahdia Anwar, Jarrod M. Psutka, Jeff Crouse, Chang Liu, J. C. Yves Le Blanc, Justin Montgomery, Gilles H. Goetz, John S. Janiszewski, J. Larry Campbell, and W. Scott Hopkins. 2018. Determining molecular properties with differential mobility spectrometry and machine learning. *Nature Communications* 9, 1 (Nov. 2018). https://doi.org/10.1038/s41467-018-07616-w

[25] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. 2022. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence* 4, 3 (Mar. 2022), 279–287. https://doi.org/10.1038/s42256-022-00447-x

[26] Prageeth Wijewardhane, Krupal Jethava, Jonathan Fine, and Gaurav Chopra. 2021. Graph Neural Networks Bootstrapped for Synthetic Selection and Validation of Small Molecule Immunomodulators. (Oct. 2021). https://doi.org/10.33774/chemrxiv-2021-r4xnx-v2

[27] Neal K. Williams, Isabelle S. Lucet, S. Peter Klinken, Evan Ingley, and Jamie Rossjohn. 2009. Crystal Structures of the Lyn Protein Tyrosine Kinase Domain in Its Apo- and Inhibitor-bound State. *Journal of Biological Chemistry* 284, 1 (Jan. 2009), 284–291. https://doi.org/10.1074/jbc.m807850200

[28] Laurie A. Witucki, Xin Huang, Kavita Shah, Yi Liu, Saw Kyin, Michael J. Eck, and Kevan M. Shokat. 2002. Mutant Tyrosine Kinases with Unnatural Nucleotide Specificity Retain the Structure and Phospho-Acceptor Specificity of the Wild-Type Enzyme. *Chemistry & Biology* 9, 1 (Jan. 2002), 25–33. https://doi.org/10.1016/s1074-5521(02)00091-1

[29] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. 2018. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* 9, 2 (Jan. 2018), 513–530. https://doi.org/10.1039/c7sc02664a

[30] Bo Xin, Yoshinobu Kawahara, Yizhou Wang, Lingjing Hu, and Wen Gao. 2016. Efficient Generalized Fused Lasso and Its Applications. *ACM Trans. Intell. Syst. Technol.* 7, 4, Article 60 (Jul. 2016), 22 pages. https://doi.org/10.1145/2847421

[31] Jiacheng Xiong, Zhaoping Xiong, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. 2021. Graph neural networks for automated de novo drug design. *Drug Discovery Today* 26, 6 (Jun. 2021), 1382–1393. https://doi.org/10.1016/j.drudis.2021.02.011

[32] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. 2019. Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism. *Journal of Medicinal Chemistry* 63, 16 (Aug. 2019), 8749–8760. https://doi.org/10.1021/acs.jmedchem.9b00959

[33] Wenqing Xu, Amish Doshi, Ming Lei, Michael J Eck, and Stephen C Harrison. 1999. Crystal Structures of c-Src Reveal Features of Its Autoinhibitory Mechanism. *Molecular Cell* 3, 5 (May. 1999), 629–638. https://doi.org/10.1016/s1097-2765(00)80356-1

[34] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating explanations for graph Neural Networks. *Adv. Neural Inf. Process. Syst.* 32 (Dec. 2019), 9240–9251. pmcid:PMC7138248

[35] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 430–438. https://doi.org/10.1145/3394486.3403085

[36] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. 2021. ProtGNN: Towards Self-Explaining Graph Neural Networks. (2021). https://doi.org/10.48550/arXiv.2112.00911 arXiv:2112.00911