Self-Attention Attribution: Proximity Molecule Information Interactions Inside Graph Neural Network

Zhenkang Lei[1], **Xuelian Yang**[1], **Zhuoqi Jiang**[1], **Zhipeng Li**[1], **Yeyun Chen**[1]
[1]Institute of Artificial Intelligence, Xiamen University
422 Siming South Road
Siming District Xiamen, China 361000
36920221153089, 36920221153136, 36920221153087, 36920221153097, 36920220156893
863811326, 1272111762, 870925440, 374941490@qq.com
yeyunchen2022@stu.xmu.edu.cn

Abstract

The advantages of automatic end-to-end feature extraction by graph neural networks help analyze property caused by different chemical structure. In particular, different lengths of alkane chains, despite their similar chemical composition, their different carbon chain lengths, functional groups in branched chains cause differences in solubility, surface tension and other chemical properties. Therefore, it is of great significance to use graph neural networks to study the interaction between chemical molecule. Here, we introduced an interpretable representation based on graph neural networks for the prediction of molecule interaction to explain physicochemical phenomena. We propose a novel neural network based approach to address this classic yet challenging graph problem, aiming to alleviate the computational burden while preserving a good performance.

Keywords: GNN; molecule interaction; GAT; graph similarity; interpretation

Introduction

Graph-structured data is a ubiquitous structure that can be find everywhere in the real-world, such as social networks, chemical molecule, and protein-target interactions. Recently, Graph Neural Networks have received considerable attention on a wide variety of tasks. The Categories of data that can be processed also expand from Euclidean distance to hyperbolic data. The interaction between molecules determines the physical and chemical properties of many substances and helps to explain the common physicochemical phenomena in life. But chemical modeling, theoretical calculations, and other methods for analyzing the interactions between molecules is a very time-consuming task.

Chemical molecular structure diagram is a typical type of non-Euclidean data, graph neural networks used to solve the analysis of non-Euclidean distance has long been studied. This method can effectively guide the synthesis and development of new drug molecules, study the specific effect of drugs-targeted molecules, and find new catalysts for chemical production processes. With the deepening of research problems, the research on graph neural networks has also developed, and graph attention neural networks and graph convolutional networks have also been developed to solve different problems. GAT (Veličković et al., 2018) is a type of GNN in spatial domain that introduces an attention mechanism to solve the disadvantage that GCN cannot be applied to dynamic graphs. Using the interaction of the drug with the targeted molecule, SkipGNN (Sun et al, 2020) specifies a neural architecture in which neural information is transmitted not only through direct interactions, but also through similarity in secondary interactions, known as jump similarity. Importantly, while the principle of jumping similarity governs many types of molecular interaction networks, GNN methods fail to capture this principle.

Graph neural networks are of great significance for feature extraction of non-Euclidean data. The core of graph neural networks lies in locality, aggregation, and composition. The expression learning of graph node features is the core of graph neural network feature extraction. The similarity of two vectors can be expressed in terms of cosine similarity, etc. Due to the disordered and mobile nature of graph nodes, the graph is required to maintain isomorphism-perserving translation, and there are multiple sequential order plans for the same graph. The node features in the graph neural network can be calculated and updated through multiple rounds of neighbor message diffusion, that is, Message Passing, and after a certain round of message passing/update, the feature values of the nodes will tend to converge, and then update, the feature values will remain unchanged and enter a steady state. GNNs are divided into two categories: Spatial, which performs node embedding updates through message exchange, and Spectrum, which is processed into the frequency domain through Fourier transforms and then inversely transformed.

A deep graph similarity learning model SimGNN is proposed in (Bai et al. 2019a) which also aims to learn similarity for chemical compounds as one of the tasks. Instead of using sub-graphs or other explicit features, the model adopts GCNs to learn node-level embeddings, which are fed into an attention module after multiple layers of GCNs to generate the graph-level embeddings. Then a neural tensor network (NTN) (Socher et al. 2013) is used to model the relation between two graph-level embeddings, and the output of the NTN is used together with the pairwise node embedding comparison output in the fully connected layers for predict-

ing the graph edit distance between the two graphs.

Though GNN-based methods are theoretically superior to SMILES-based methods in learning molecule structure, they are limited to designing fresh and delicate GNN architectures while ignoring the essence of molecule representation learning, which is generalization ability.

We propose a new network framework based on the Graph attention mechanism, which uses the multi-attention to extract the intermolecular interactions, hoping that the model can learn the difference between the three types of forces based on the same attention mechanism, and then combine different forces as knowledge, and then predict the interaction between the two molecules through similarity score. We conduct extensive experimental studies on both tasks, and the results demonstrate the outstanding performance of our Prox-GNN by comparison with baseline method. Our work will help reduce the computational cost compared to the previously widely used density functional theory calculation methods, thus effectively advancing the study of intermolecular interactions, molecular property prediction, and contributing to experimental and theoretical studies.

Related work

For the prediction of chemical properties, many ML methods had been developed. The majority of the studies have made Graph Neural Networks the tool of choice. Three main categorization based on which model architecture has been used: (1) graph embedding based methods; (2) graph neural networks based models; (3) deep graph kernels

Actually, the message exchange of spatial GNN is a simple matrix multiplication of node embeddings and adjacency matrices. GCN (Welling et al., 2016) uses the Laplace operator and Fourier transform, sharing the connection relationship between nodes and effectively extracting the characteristics of the graph. GNN are often shallow to avoid oversmooting. GraphSAGE (Hamilton et al., 2017) implements inductive learning by learning the aggregate function: mean, LSTM, and pooling aggregator. GAT proposes to take attention to make neighbors play different roles in aggregation (Casanova A et al., 2017). Introducing adaptive Knowledge Distillation in GNN (Guo et al., 2022) can boosting knowledge transfer. The Graph Matching Networks (GMNs) compute a similarity score through a cross-graph attention mechanism to associate nodes across graphs and identify differences.

Graph similarity learning. The nodes of a graph can be divided into two disjoint sets, i.e., labeled and unlabeled. The goal of node classification is to predict unlabeled labels based on the learning of labeled nodes. Node features learned in graph classification task for node classification. A graphical representation of chemical molecules can lighten the need for feature engineering. Mapping nodes into an embedding space is potentially used as a downstream prediction. Node similarity can be considered from the following aspects: whether two nodes are connected or not, whether there are shared neighbor nodes, and whether there are similar structural representations. Similarity can be evaluated by K-L divergence, cosine, distance, et al method. Siamese (Lecun et el., 2005) embeds pairs of data into the same vector space through weight sharing, and calculates similarity by calculating vector distances. The representation of each graph by this method is calculated separately, and the interaction information between the graphs is lacking. SimGNN (Bai et al. 2019a) first formulates graph similarity learning as a regression task, where its GCN and attention layers are supervised by GED scores solved by A*. They extends their previous work by processing a multi-scale node-wise similarity map using CNNs. It has been proposed that the cross-graph attention mechanism is used to interact with graph information in the graph embedding process, to improve the representation ability of embedding. SkipGNN has jump similarity, which not only by aggregating information from direct interactions, but also by information from secondary interactions.

Embedding method. Generating embeddings for every node in the network to capture the node's local network topology. A popular approach is to use random walks with a skip-gram model, such as DeepWalk, node2vec, and LINE. The use of the random walk algorithm enables flexible and randomly defined integration of local and global information of nodes. Node2vector makes the neighbors of adjacent networks after node embedding also have similar coordinates in the feature space. The other popular approach leverages the spectral graph theory to generate a spectral embedding such as spectral clustering. The generated node embeddings are then fed into a decoder classifier to predict the link existing probability. SimGNN using mutual information to enhancing the interaction of two graph. The generic framework for the graphs embedding is:

- Mapping function to map nodes from graph domain to embedding domain
- Information extractor for extracting the key information I to be retained in the graph domain
- Reconstructor that reconstructs the extracted graph information I using embeddings in the embedding domain
- Learning the parameters involved in the mapping and reconstructors by optimizing the objectives based on the extracted information I and reconstructed information I'.

The time complexity is usually still polynomial or even subexponential in the number of nodes in the graphs, such as A*-Beamsearch (Beam), Hungarian,VJ, etc.

Aggregation function. The symmetry of the aggregation function ensures that the neural network model can be trained and applied to an arbitrary sequence of vertex neighboring feature sets. The symmetry of the aggregation function ensures that the neural network model can be trained and applied to an arbitrary sequence of vertex neighboring feature sets. Compared to mean aggregators, LSTMs have a stronger expressive power. However, LSTMs are not pairwise, i.e., they do not have permutation invariant. In skipGNN, an iterative fusion scheme with aggregation gates

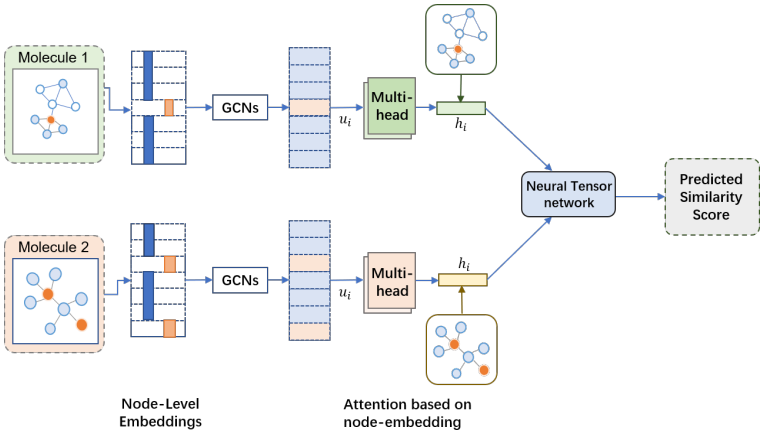

Figure 1: Flowchart of the our framework.

is designed to combine the similarity information of both. Instead of simply concatenating the output node embedding from the GNN output of the original graph G that captures direct similarity and the GNN output of the skip graph G that captures skip similarity, the two GNNs on G and G are allowed to interact with each other iteratively through propagation rules.

Dynamic interaction. Recent related works on graph-based methods for human motion prediction include(Alahi et al., 2016) where the graph is not learned but is based on proximity and (Le et al., 2017) tries to cluster agents into roles. A number of recent works (Monti et al., 2017;Duan et al., 2017; Hoshen, 2017;Veličković et al., 2018) parameterize messages in GNNs with a soft attention mechanism (Luong et al., 2015). This equips these models with the ability to focus on specific interactions with neighbors when aggregating messages.

Proposed Solution

Our task is to explore two problems: estimating the properties of molecule and predicting future states when different two molecules interact.

Here, we devised a simple method to calculate the similarity of two graphs in order to initially determine the interaction between two chemical molecules from their solubility data. This model in Figure 1 is more interpretable and generalizable than the SimGNN model for calculating intermolecular forces of proteins.

Now, we introduce our proposed approach **Prox-GNN** in detail, which is a new neural network architecture capable of directly modeling molecule as graph, and show that this approach outperforms state-of-the-art deep learning models on two molecules similarity prediction benchmarks. Using SMILES code, it is transformed into our model and further extracts the features of the nodes into one-hot vectors. SMILES based on the chemical molecule in Prox-GNN is used as input to construct the molecular graph of the molecule and extract atomic features by open source code RDKit, and this graph structure data is input to the GCN layer learn the potential patterns in the graph feature representation. The intermolecular interaction prediction problem is then converted to a regression task where the input is a pair of molecule representations, and the output is a continuous value reflecting the similarity score of the pair.

After that the graph level embedding is updated by the attention module. Note that here we introduce more attention layers to avoid GCN over-smoothing the molecular node features.

Vanilla GNN. The trivial way of handling the dynamic condition is that when the graph is modified, a complete feed-forward pass is called for all nodes in the new graph. However, such practice involves redundant computation, which is discussed as follows. We denote n as the number of nodes, F as embedding dimensions, and K as the number of GNN layers.

Graph convolutional networks. GCN is a inductive method for inference, as it is graph representation-invariant. Graph convolution operates on the representation of a node, which is denoted as $u_n \in \mathbb{R}^D$, and is defined as follows: $conv(u_n) = f1(\sum m \in N(n) \frac{1}{\sqrt{d_n d_m}} u_m W_1^{(l)} + b_1^{(l)})$ where N(n) is the set of the first-order neighbors of node n plus n itself, d_n is the degree of node n plus 1,$W_1^{l \in D^l \times D^l + 1}$ is the weight matrix associated with the l-th GCN layer, $b_1^l \in \mathbb{R}^{D^{l+1}}$ is the bias, and $f_1(\cdot)$ is an activation function such as $ReLU(x) = max(0, x)$. Intuitively, the graph convolution operation aggregates the features from the first-order neighbors of the node.

The potential issue of using a deep GCN architecture is that the embeddings may lose subtle patterns in local neighborhood after aggregating neighbors multiple times. The issue is especially severe when the two graphs are very similar, and the differences mainly lie in small local substructures.

Therefore, in our model, only one layer of GCN is retained for generalizing molecular information at graph level.

Attention block for graph-level embedding. Given node-level embeddings, the graph-level embedding is obtained through attention mechanism. To extract more feature from graph-structured data, we add double-layer attention for keep the local substructure for molecule.

Graph attention mechanism. Both GCN and GAT aggregate the features of neighboring vertices to the central vertex (an aggregate operation) and use the local stationary on the graph to learn the new vertex feature representation. The difference is that GCN uses Laplace matrix and GAT uses attention coefficients. To some extent, GAT is stronger because the correlation between vertex features is better integrated into the model. As with all attention mechanisms, the GAT is calculated in two steps.

Firstly, calculate the attention coefficient (attention coefficient). For vertex i, compute the similarity coefficient between its neighbors and itself one by one

$$\alpha_{ij} = \frac{exp(LeakyReLU(e_{ij}))}{\sum_{k\in\mathcal{N}_i} exp(LeakyReLU(e_{ik}))}$$

In the second step, the features are weighted and aggregated according to the computed attention coefficients.

$$h_i^{'} = \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij} W h_j)$$
$$h_i^{'}(K) = \Big\Vert_{k=1}^{K} \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k W^k h_j)$$

Neural Tensor Networks. Graph-graph interaction is model by Neural Tensor Networks(NTN) as follows:

$$g(h_i, h_j) = f_3(h_i^T W_3^{[1:K]} h_j + V\begin{bmatrix}h_i\\h_j\end{bmatrix} + b_3)$$

where$W_3^{[1:K]} \in \mathbb{R}^K$ is a bias vector, and $f_3(\cdot)$ is an activation function. K is a hyperameter controlling the number of interaction (similarity) scores produced by the model for each graph embedding pair. Module frame can be find in figure2. Introducing NTN module that performance can be improved when entities are represented as an average of their constituting word vectors. This allows sharing of statistical strength.

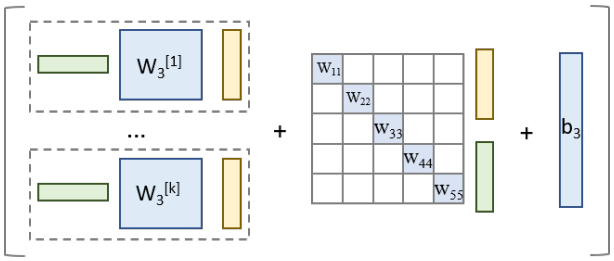

Figure 2: Neural Tensor Network block.

Experiments

Datasets

We evaluate our Prox-GNN model on three challenging real-world datasets: ESOL, LIPO, and AIDS. Each set contains thousands of molecule SMILES as well as binary labels indicating the property of interest.

ESOL is a small dataset containing water solubility data for 1128 compounds. This dataset can be used to train models to predict solubility based on chemical molecular structures (encoded in SMILES strings) that do not contain 3D coordinates of atoms.

Lipophilicity is an important characteristic of drug molecules that affects membrane permeability and solubility. This dataset, from the ChEMBL database, provides experimental results on the octanol/water partition coefficient (logD at pH 7.4) for 4200 compounds.

AIDS is a collection of antivirus screen chemical compounds from the Developmental Therapeutics Program at NCI/NIH7, and has been used in several existing works on graph similarity search. It contains 42,687 chemical compound structures with Hydrogen atoms omitted. We select 700 graphs, each of which has 10 or less than 10 nodes.

Implementation Details

To demonstrate our model, we test it on three graph datasets and compare it with state-of-the-art methods on graph neural networks. We convert the SMILES code to its corresponding molecular graph and extract atomic features using the open-source chemical informatics software RDKit (Landrum, 2006). For the model architecture, we set the number of GCN layers to 1, and use ReLU as the activation function. For the initial node representations, we adopt the onehot encoding scheme for the input data reflecting the node type. The output dimensions for the layer of GCN are 128. We add 2 GAT module with two-head, the output dimensions for the 1st and 2nd layer of GAT are 64 and 32, respectively. For the NTN layer, We set K to 16. We use 2 fully connected layers to reduce the dimension of the concatenated results from the NTN module, from 16 to 16, and 16 to 1. More parameter details can be find in the table 1. We conduct all the experiments on a single machine with an Intel i7-12700h CPU and one Nvidia RTX 3070Ti GPU.

Table 1: **Hyper-parameters for different graph neural network variants used in our models.**

Hyper-parameters	Setting
Learning rate	0.001
Batch size	128
Optimizer	Adam
GCN layers	1
GAT layers	2
Multi-head	2
FC layers	2

Evaluation Metrics

The following metrics are used to evaluate all the models: The mean squared error ($MSE = \frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$) measures the average squared difference between the computed similarities and the ground-truth similarities. We also adopt the following metrics to evaluate the ranking results. Spearman's Rank Correlation Coefficient and Kendall's Rank

Correlation Coefficient measure how well the predicted ranking results match the true ranking results. ρ and τ evaluates the global ranking results instead of focusing on the top k results.

Spearman's Rank Correlation Coefficient:

$$\rho = \frac{\sum_i (x_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (y_i - \overline{y})^2}}$$

Kendall's Rank Correlation Coefficient:

$$\tau = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}$$

where n_c is the number of concordant pairs in a two-comparison pair, n_d is the number of discordant pairs in two-comparison pairs, n_0 is the total number of pairs in two-comparison pairs, which is $n_0(n_0-1)/2$, and n is the sample size, n_1 is the number of invariant pairs in which the value of X is invariant, n_2 is the number of invariant pairs in which the value of y is invariant.

The results of the correlation coefficient evaluation are shown in Table 2, where it can be seen that our model has a higher Spearman coefficient compared to the other models and shows a higher intermolecular interaction.

Table 2: **Evalution of Similarity relation based on Spearman and Kendalltau**

Model	ρ	τ
GMN	0.672±0.036	0.497±0.032
GraphSim	0.849±0.008	0.693±0.010
SimGNN	0.824±0.009	0.665±0.011
Proposed model	0.8967±0.007	0.600±0.010

Ablation Study: How well Attention works?

To analyze our method in depth, ablation studies are conducted to evaluate the effect of each individual component and the results on the ESOL, LIPO, AIDS datasets are reported in Table 3.

Effect of histogram. To analyze the effect of histogram, we compare our method with the model with histogram. The experimental results prove that the histogram makes the model worse, which may be due to the fact that this statistical method tends to ignore local features although it can extract the features of the layers.

Effect of attention. We also conducted additional experiments on SMILES string dataset with ablation consideration. We analyzed the attribution of the attention layer, and here we used a double layer of attention. The experimental results show that adding GAT block can effectively prevent the over-smoothing caused by GCN, extract molecular detail features, and improve the model performance.

Comparison with State-Of-The-Art Methods

In this section, we compare the proposed method with five state-of-art methods on AIDS databases in table 4. We consider both the state-of-the-art similarity computation methods and baselines using neural networks.

Table 3: **Predicition performance on the three datasets, sorted by MSE(10^{-3})**

Methods	ESOL	LIPO	AIDS
SimGNN	3.29293	1.04566	0.33244
SimGNN-h	2.44216	1.00012	0.27548
Prox-GNN+h	2.12896	1.02549	0.30678
Prox-GNN	2.03074	0.99405	0.27211
Baseline error	3.59432	1.0839	0.54748

To ensure consistency, all neural network models use three layer GCN for node embeddings except for Prox-GNN, and to demonstrate the flexibility of our framework, we show the performance improvement of Prox-GNN by removing historm and adding with the more powerful Attention's node embedding methods.

Table 4: **Comparison experiment conducted on AIDS dataset**

Methods	MSE (10^{-3})
SGNN(FCMax)	3.114 ± 0.114
SGNN(BiLSTM)	1.422 ± 0.044
GMN (Li, et all.,2019)	4.610 ± 0.365
GSimGNN (Bai, et al. 2020)	1.919 ± 0.060
Baseline (Bai, et al. 2019)	1.376 ± 0.066
Proposed model	0.797 ± 0.112

Results and Analysis

The evaluation of ESOL, LIPO, and AIDS dataset in line with is presented in Dataset section, where the problem is defined as evaluate the similarity of the graph from all graphs in the training set. Our regression model Prox-GNN has comparable performance against state-of-the-art with a simplified pipeline, and better performs among other models.

Conclusion

In this paper, we have presented a Prox-GNN model that consists of three main block: GCN, GAT, NTN module. Driven by the attention meachanism that is employed in the backbone network, our model is able to extract more features. Thanks to the multi-headed attention, more finer molecular structures can be find. However, due to time constraints, we did not analyze attention at different layers, attention at the graph level, and visualization of attention weights. In the future, we expect the model to extract more chemical information to make graph neural networks according to interpretability.

Reference

Veličković, P; Cucurull, G, Casanova, A. 2017. Graph attention networks. *arXiv* preprint arXiv:1710.10903.

Huang, K.; Xiao, C.; Glass, L. M.; Zitnik, M.; Sun, J. 2020. SkipGNN: predicting molecular interactions with skip-graph networks. *Sci Rep*, 10 (1), 21092.

Bai Y , Ding H , Gu K , et al. 2020. Learning-Based Efficient Graph Similarity Computation via Multi-Scale Convolutional Set Matching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4):3219-3226.

Socher R., Chen D., Manning C., et al. 2013. Reasoning with neural tensor networks for knowledge base completion. *Curran Associates Inc.*.

Kipf, T. N., & Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv* preprint arXiv:1609.02907.

Hamilton, W., Ying, Z., & Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Guo, Z., Zhang, C., Fan, Y., Tian, Y., Zhang, C., & Chawla, N. 2022. Boosting Graph Neural Networks via Adaptive Knowledge Distillation. *arXiv* preprint arXiv:2210.05920.

Chopra, S., Hadsell, R., & LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In 2005 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 539-546). IEEE.

Le, H. M., Yue, Y., Carr, P., & Lucey, P. Coordinated multiagent imitation learning. 2017 *In International Conference on Machine Learning, (ICML).*

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Duan, Y., Andrychowicz, M., Stadie, B. C., Ho, J., Schneider, J., Sutskever, I., Abbeel, P., & Zaremba, W. 2017. One-shot imitation learning. *In Advances in Neural Information Processing Systems (NIPS).*

Monti, F., Boscaini, D., & Masci, J. 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Hoshen, & Y. Vain: Attentional multi-agent predictive modeling. *In Advances in Neural Information Processing Systems (NIPS).*

Luong, M.-T., Pham, H., & Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. *In Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Zhu, Y.; Ouyang, Z.; Chen, W.; Feng, R.; Chen, D. Z.; Cao, J.; and Wu, J. 2021. TGSA: Protein-Protein Association-Based Twin Graph Neural Networks for Drug Response Prediction with Similarity Augmentation. *Bioinformatics*, btab650.

Shao K.; Zhang Y.; Wen Y.; Zhang Z.; He S.; Bo X. 2022. DTI-HETA: prediction of drug-target interactions based on GCN and GAT on heterogeneous graph. *Briefings in Bioinformatics*,bbac109.

Liu, Z.; Chen, Q.; Lan, W.; Liang, J.; Chen, Y. P.; and Chen, B. 2020. A Survey of Network Embedding for Drug Analysis and Prediction. *Current protein and peptide science*.

Schwarz, K.; Pliego-Mendieta, A.; Planas-Paz, L.; Pauli, C.; Allam, A.; and Krauthammer, M. 2022. DDoS: A Graph Neural Network based Drug Synergy Prediction Algorithm.*arXiv* preprint arXiv:2210.00802.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE transactions on neural networks and learning systems*, 32(1), 4–24.

Yan,X.; You,Z.; Wang,L.; and Chen,P. 2021. CNNEMS: Using Convolutional Neural Networks to Predict Drug-Target Interactions by Combining Protein Evolution and Molecular Structures Information. *In Intelligent Computing Theories and Application: 17th International Conference, ICIC,Proceedings, Part III. Springer-Verlag*, Berlin, Heidelberg, 570–579.

Nikolentzos, G.; Meladianos, P.; and Vazirgiannis, M. 2017. Matching Node Embeddings for Graph Similarity. *In AAAI*, 31(1).

Kang, C.; Zhang, H.; Liu, Z.; Huang, S.; and Yin, Y. 2022. LR-GNN: a graph neural network based on link representation for predicting molecular associations. *Briefings in bioinformatics*, 23(1), bbab513.

Shen, Z. A.; Luo, T.; Zhou, Y. K.; Yu, H.; and Du, P. F. 2021. NPI-GNN: Predicting ncRNA-protein interactions with deep graph neural networks. *Briefings in bioinformatics*, 22(5), bbab051.

Zhu,W.; Zhang, Y.; Zhao, D.; Xu, J.; and Wang, L. 2022. HiGNN: Hierarchical Informative Graph Neural Networks for Molecular Property Prediction Equipped with Feature-Wise Attention. *arXiv* preprint arXiv:2208.13994.

Chen, J.; Chen, S.; Bai, M.; Gao, J.; Zhang, J.; and Pu, J. 2022. SA-MLP: Distilling Graph Knowledge from GNNs into Structure-Aware MLP. *arXiv* preprint arXiv:2210.09609.

Ryu, S.; and Lee, S. 2022. Accurate, reliable and interpretable solubility prediction of druglike molecules with attention pooling and Bayesian learning. *arXiv* preprint arXiv:2210.07145.

Spalevi'c, S.; Velivckovi'c, P.; Kovavcevi'c, J.; and Nikolic, M. 2020. Hierarchical Protein Function Prediction with Tail-GNNs. *arXiv* preprint arXiv:2007.12804.

Verma, V.; Qu, M.; Kawaguchi, K.; Lamb, A.; Bengio, Y.; Kannala, J.; and Tang, J. 2021. GraphMix: Improved Training of GNNs for Semi-Supervised Learning. *In AAAI* , 35(11), 10024-10032.

Webb, M. A.; Delannoy, J. Y.; and de Pablo, J. J. 2019. Graph-Based Approach to Systematic Molecular Coarse-Graining. *Journal of chemical theory and computation*, 15(2), 1199–1208.

Li, Y.; Gu, C.; Dullien, T.; VinyalsO., Kohli P. 2019. Graph Matching Networks for Learning the Similarity of Graph Structured Objects. *ICML*.

Nguyen T., Le H , Quinn T P , et al. 2020. GraphDTA: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*.

Landrum, G. 2006. RDKit: Open-source cheminformatics.