GLAM: An adaptive graph learning method for automated molecular interactions and properties predictions

Yuquan Li
Lanzhou University https://orcid.org/0000-0003-2756-0449
Chang-Yu Hsieh
Tencent
Ruiqiang Lu
Lanzhou University
Xiaoqing Gong
Lanzhou University
Xiaorui Wang
Lanzhou University
Pengyong li
Tsinghua University https://orcid.org/0000-0001-5971-046X
Shuo Liu
Lanzhou University
Yanan Tian
Lanzhou University
Dejun Jiang
tingjunhou@zju.edu.cn https://orcid.org/0000-0002-2035-5074
Jiaxian Yan
University of Science and Technology of China
Qifeng Bai
Lanzhou University https://orcid.org/0000-0001-7296-6187
Huanxiang Liu
Lanzhou University
Shengyu Zhang
Tencent
Xiaojun Yao (✉ xjyao@lzu.edu.cn)
Lanzhou University

Article

Keywords:

Posted Date: January 6th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1172418/v1

Version of Record: A version of this preprint was published at Nature Machine Intelligence on June 23rd, 2022. See the published version at https://doi.org/10.1038/s42256-022-00501-8.

GLAM：An adaptive graph learning method for automated molecular interactions and properties predictions

Yuquan Li [1,2,†], Chang-Yu Hsieh [2,†], Ruiqiang Lu [1], Xiaoqing Gong [1], Xiaorui Wang [3], Pengyong Li [4], Shuo Liu [5], Yanan Tian [5], Dejun Jiang [6], Jiaxian Yan [7], Qifeng Bai [8], Huanxiang Liu[5], Shengyu Zhang [2], & Xiaojun Yao [1,*]

[1] College of chemistry and chemical engineering, Lanzhou University, Lanzhou, China.

[2] Tencent Quantum Laboratory, Tencent, Shenzhen, China.

[3] State Key Laboratory of Quality Research in Chinese Medicines, Macau University of Science and Technology, Macau, China.

[4] School of computer science and technology, Xidian University, Xian, China.

[5] School of pharmacy, Lanzhou University, Lanzhou, China.

[6] College of Computer Science and Technology, Zhejiang University, Hangzhou, China.

[7] School of Data Science, University of Science and Technology of China, Hefei, China.

[8] School of Basic Medical Sciences, Lanzhou University, Lanzhou, China.

[†]These authors contributed equally to this work.

[*]Corresponding authors. E-mail: xjyao@lzu.edu.cn

Abstract

Improving drug discovery efficiency is a core and long-standing challenge in drug discovery. For this purpose, many graph learning methods have been developed to search potential drug candidates with fast speed and low cost. In fact, the pursuit of high prediction performance on a limited number of datasets has crystallized them, making them lose advantage in repurposing to new data generated in drug discovery. Here we propose a flexible method that can adapt to any dataset and make accurate predictions. The proposed method employs an adaptive pipeline to learn from a dataset and output a predictor. Without any manual intervention, the method achieves far better prediction performance on all tested datasets than traditional methods, which are based on hand-designed neural architectures and other fixed items. In addition, we found that the proposed method is more robust than traditional methods and can provide meaningful interpretability. Given the above, the proposed method can serve as a reliable method to predict molecular interactions and properties with high adaptability, performance, robustness and interpretability. This work would take a solid step forward to the purpose of aiding researchers to design better drugs with high efficiency.

1 Introduction

Drug discovery is a lengthy, costly, and complex process that plays a crucial role in human health and well-being[1,2]. At present, experimental assays[3] remain the most reliable approach to screen compounds but cost too much. Although many computational methods[4] are proposed to estimate molecular interactions and properties and improve drug discovery efficiency, it is still a tricky process.

Graph learning methods have the potential to improve drug discovery efficiency dramatically for their ability to amplify insights available from existing drug-related datasets[5]. Using the insights to predict molecular interactions and properties[6,7] is a key to finding potential drug candidates from the vast chemical space with extremely fast speed and low cost. On the other hand, molecular generation[8,9] based on the insights can more efficiently traverse the vast chemical space to find potential drug candidates. Accordingly, graph learning is becoming a rising area of interest within the field of drug discovery[10].

However, the pursuit of high prediction performance on a limited number of existing datasets has crystallized them[11–13], making them lose advantage in repurposing to new data generated in drug discovery. In practice, a graph learning method tends to be made with a fixed architecture and a fixed hyperparameter set to get the best performance on a dataset[14]. This heavily limits them in repurposing to new data generated in drug discovery, which tend to be increasingly complex[15]. Besides, most graph learning methods rely heavily on expert knowledge of deep learning to achieve their claimed state-of-the-art results, or even cannot be reproduced[16]. Therefore, they are not flexible and reliable enough.

More recently, a few works have been reported to address these problems above. MolMapNet introduced an out-of-the-box deep learning method based on broadly learning knowledge-based representations to get reliable prediction performance on more datasets[7]. A recent work introduced a neural architecture search based method to design neural architecture for molecular property predictions[11].

In this work, we proposed **G**raph **L**earning based **A**daptive **M**achine (**GLAM**), a flexible method that can adapt to any dataset and make accurate predictions without human intervention. We fairly compared our proposed method with previously reported methods in terms of adaptability and prediction performance. The results show that our proposed method can adapt to all tested datasets exceptionally well and obtain far better prediction performance than reported methods. On the other hand, we also investigated the robustness and interpretability of our proposed method. We found that the proposed method is more roubst than tested methods and can provide meaningful intepretability, making the proposed method a more reliable method.

2 Results

Method Overview

Our method utilizes an automated pipeline to learn from datasets and build a predictor, as shown in **Figure 1.** Our method is mainly different from previous graph learning methods[17–22], which rely heavily on human experts to design architecture, tune model hyperparameters, select the optimizer, and select the loss function. We creatively combine these four items into a configuration and put potential configurations into a configuration space. Starting from this configuration space, we perform configuration sampling, low-fidelity training based on sampled configurations, configuration selection, high-fidelity training based on selected configurations, predictor selection, and predictor blending to build a predictor.

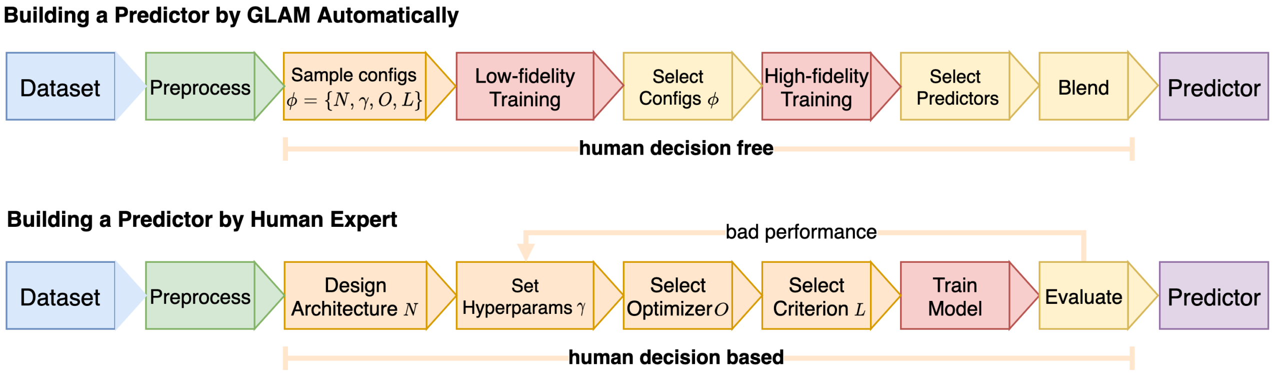


Figure 1. Overview of GLAM and traditional method.

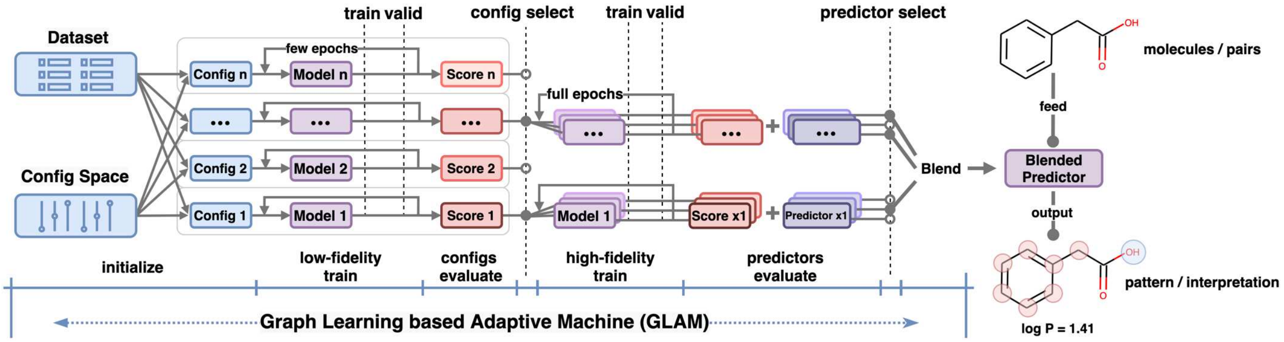


Figure 2. GLAM pipeline details.

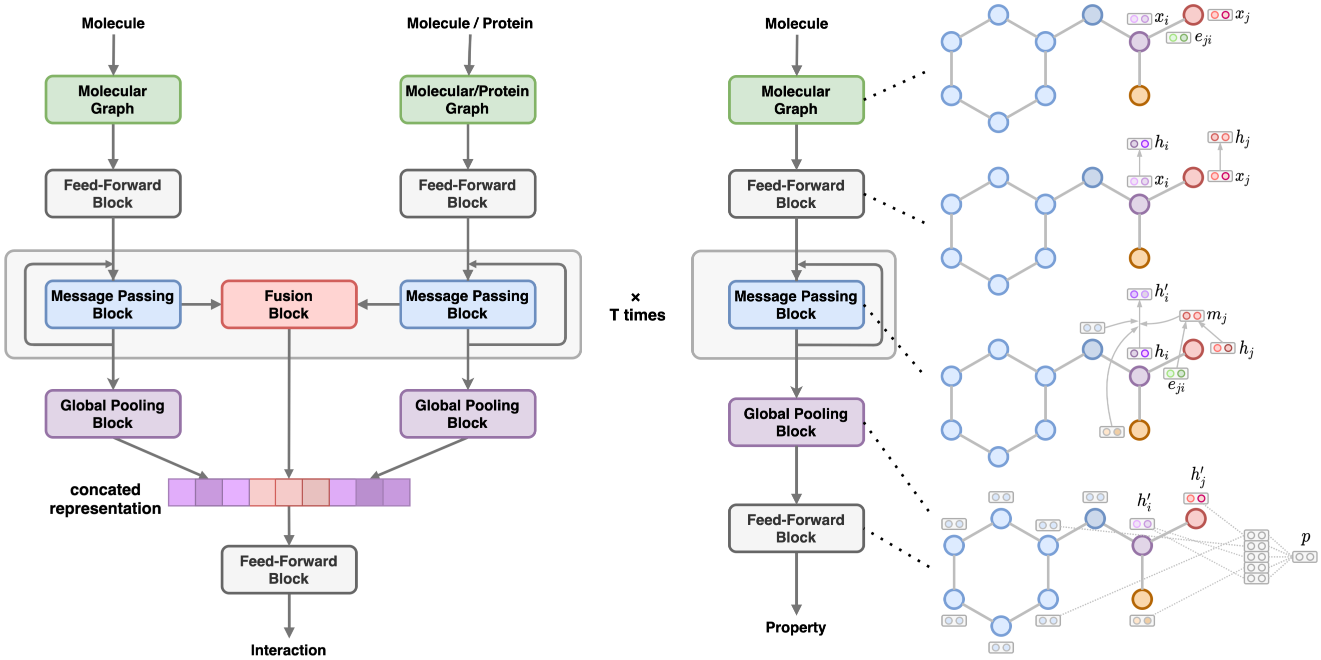


Figure 3. the general architectures for molecular interactions and properties prediction in GLAM.

In detail, the method has a well-designed pipeline that is to achieve high adaptability, prediction performance, and acceptable speed, as shown in **Figure 2.** Initially, GLAM samples a lot of configurations from the configuration space. Then the dataset is fed to these configurations for low-fidelity training, which produces evaluation scores to select the high-performance configurations. In the next high-fidelity training, each previously selected configuration is fed with the dataset to perform full epochs training with three different random seeds. Accordingly, the evaluation scores of the high-fidelity training are used to select high-performance predictors. Finally, all selected predictors are blended into a predictor.

We have designed two general architectures, one for molecular interaction and another for molecular property, as shown in **Figure 3**. In practice, the input data are mapped to output by five steps: (1) preprocess a molecule to a molecular graph, (2) linear transformation by feed-forward block, (3) iterated convolution by message-passing block for T time-steps, (4) node reduction by global pooling block, (5) linear transformation by final feed-forward block. Each block in the general architecture is created with its own design space, as shown in **Extended Data Figure 1**.

Adapt to any dataset for high performance

GLAM was created to adapt to any dataset to obtain high prediction performance. To investigate the adaptability and performance of our method, we compared its performance on 14 datasets with a range of representative traditional methods[7,17–21,23–25]. The types of tested datasets include drug-protein interactions, drug-drug interactions, physical-chemistry property, bioactivity, pharmacokinetics, and toxicity. Considering that different splits of datasets cause different performances, we let all methods share the same splits of datasets for a fair evaluation. We also try our best to adjust the parameters of all methods manually to achieve the best performance. Finally, we run benchmarks on molecular interaction and molecular property datasets.

Compared to all traditional methods, our proposed methods can adapt to datasets well and achieve promising prediction performance, as shown in **Table 1**, **Table 2** and **Extended Data Table 1**. (All tables can be found at the end of this manuscript.) Given the above, GLAM sets a new state-of-the-art on both molecular interactions and properties prediction. Relative to the best scores of reported results, the proposed method got an average 18.7% decrease of prediction error on 14 datasets than the best of traditional methods. In addition, GLAM can consistently achieve the best scores without human intervention. Therefore, GLAM is poised to be a flexible, reliable, and trustworthy method that works well across a wide range of activities in drug design.

High robustness against molecular structure perturbation

The following essential issue is about the robustness[26] of the proposed method. Graph learning methods usually are susceptible to outliers, and it can be disastrous if data is contaminated with outliers. In fact, some safe-sensitive domains (such as healthcare) require more robustness of a method as even a slight perturbation will lead to a wrong result with serious consequence35. Admittedly, the robustness of a graph learning method is also an essential issue.

To evaluate the robustness of our proposed method, we first introduce an algorithm named **p**roperty-slightly-**a**ffected **s**tructure **p**erturbation (PASP). The PASP works by iteratively mutating the molecular structure with small perturbations that do not significantly affect the properties of the molecules.

Then we perform the PASP algorithm on a dataset named PhysProp[28], consisting of 14,176 molecules structures with their corresponding properties $\log P$. To estimate the potential change of $\log P$ upon mutations, we use RDKit[29] to calculate the $\log P$ values before and after PASP as an observer. Moreover, we limit an acceptable change (value change < 0.2 before and after the PASP), which can ensure that the PASP does not significantly affect the $\log P$ of molecules. In this experiment, The molecular structures whose similarity (before and after the mutation) of fingerprint in the scale of 0.8-1.0, 0.5-0.8, and 0.3-0.5 are marked as levels 1, 2, and 3, respectively.

Table 3 shows that GLAM is less affected by the PASP and manifests higher robustness than conventional methods. We first run GLAM on the unperturbed train and validation sets to get a predictor. Then, we feed the test set to the predictor and migrate the testing to the three perturbed test sets. Compared with conventional methods, including GCN[18], GAT[19], GIN[20,22], MPNN, the results show that all these methods suffer a significant performance drop, as shown in **Table 3**. In contrast, our method is less affected by PASP than conventional methods on all three levels of PASP. The robustness of the GLAM is most likely due to the model blending at the end of the pipeline. The main idea of blending is to train several models and draw a final prediction from averaging. So, perturbing the molecular structure may cause a molecule misclassified by a predictor but not by all predictors.

Interpretation cases

To better understand the models in the predictor, we investigate its decision-making process and interpret its learned knowledge. In the past, most machine learning models have long been considered as black boxes. Previous works adopt attention mechanisms[21,30,31] to aid interpretation of the model. Here we explain the model from the hidden states by averaging and visualizing it, which can directly utilize the information provided by the models in the predictor.

Extended Data Figure 2a shows some case studies of solubility prediction, which are consistent with the intuition of chemists. Generally, hydroxyl and amino groups are considered to be more hydrophilic and alkyl and halogen groups are considered to be more lipophilic. We select and visualize some representative molecules from the PhysProp dataset. The atoms in the hydrophilic group tend to be bluer in our visualization, which means their weights are closer to 1.

In contrast, the atoms in the lipophilic group tend to be redder in our visualization, which means their weights are closer to -1. These observations are consistent with the intuition of chemists, indicating that the models in predictor can detect essential atomic groups with clear interpretability of results of solubility.

In the same way, we visualize some cases of drug-drug interaction identification as shown in **Extended Data Figure 2b**. We consider the interactions between sidenafil/udenafil and nitrates(nicorandil/ isosorbide dinitrate) as cases. They are combined into four pairs of drug-drug interactions and fed into models in predictor to visualize the decision process. (Typically, the sildenafil/udenafil can selectively inhibit the phosphodiesterase type 5 (PDE5) targets in the human body. The N-methyl groups in the pyrazolorpyrimidone rings of sildenafil/udenafil are important for the activity and selectivity of the PDE5. For the reasons above, combining sildenafil/udenafil with nitrate drugs may lead to serious blood pressure reduction and heart diseases.) The visualization results show the models in predictor pay more attention to the nitrates of isosorbide dinitrate and nicorandil, and pay more attention to the N-methyl of sildenafil and udenafil. Thus, Our visualization results are consistent with previous findings of drug interaction, indicating that models in the predictor can provide deep insights into molecular interactions.

3 Discussion

We have shown that GLAM can adapt to all tested datasets well and make accurate predictions automatically. In the past, the adaptability for new data has largely been ignored as most researchers pay almost all attention to getting high prediction performance. Our well-designed method can serve as a reliable method to predict molecular interactions and properties with high adaptability, prediction performance, robustness and interpretability. Furthermore, the automated pipeline of our proposed method enables more researchers, even those who lack machine learning experience, to make full use of the power of machine learning. These advantages of our proposed method will greatly increase the acceptance of machine learning aided drug discovery.

4 Limitations and frontiers

Adaptive feature input can help the models in our proposed method to extract important and sufficient representations. In this work, we only provide the graph model with the basic node features, such as atomic/residual number. Adaptive feature input can be of great help to some particular prediction jobs. Adding the feature decisions to configuration space might improve our proposed method. An strategy that provides neither too little information nor too redundant information would contribute a lot to the representation extraction process.

5 Outlook

Our proposed method is expected to advance and evolve the automated drug design[1,32]. Recent advances such as chemical retrosynthesis predictions[33,34] and molecular generations[35–37] have laid the foundation for an automated drug design pipeline in the future. Automated drug design or semi-automatic drug design will become a trend. The proposed method can serve as a predictor generator contributing to the automated drug design. For further applications, our proposed method can be repurposed to more scientific discovery fields, such as agrochemicals and materials design.

Methods

Details of datasets

We use LIT-PCBA[38], BindingDB[39], and DrugBank[40] to evaluate our proposed method for molecular interaction prediction. In LIT-PCBA, we selected four datasets of representative protein based on the number of positive and negative samples. On the other hand, we used datasets in MoleculeNet[41] to evaluate the proposed method for molecular property prediction. MoleculeNet [41] is a set of benchmarking datasets for molecular machine learning to achieve a fair performance comparison.

LIT-PCBA is a virtual screening dataset consisting of fourteen targets, 7844 confirmed active, and 407,381 confirmed inactive compounds[38]. BindingDB dataset contains 39,747 positive examples and 31,218 negative examples from a public database[39]. DrugBank includes 1,850 approved drugs with 221,523 DDI positive labels [40]. The blood-brain barrier penetration (BBBP) contains 2053 molecules on their permeability properties [41]. SIDER is a database of marketed drugs and adverse drug reactions, which grouped into system organ classes for 1427 approved drugs [41]. The BACE is a database consisting of binding results for a set of inhibitors of human β-secretase 1 with 1522 compounds [41]. Tox21 contains qualitative toxicity measurements for 8014 compounds on 12 different targets, including stress response pathways and nuclear receptors [41]. ToxCast is another toxicity database that contains qualitative results of 617 experiments on 8615 compounds [41]. The Free Solvation Database (FreeSolv) provides experimental and calculated hydration free energy of 643 small molecules in water [41]. ESOL is a small dataset consisting of water solubility data for 1128 compounds [41]. Lipophilicity is an important feature of drug molecules that affects both membrane permeability and solubility of 4200 compounds [41]. PhysProp consists of 14,176 molecules and their corresponding $Log\ P$ values[28].

Configuration space

The Configuration space of GLAM consists of two parts: architecture decisions and training decisions. The architecture decisions decide how to build an architecture of a model. The general architectures of pairs- and single-molecule contain 8 and 4 blocks with their independent design space, respectively, as shown in Figure 1. The training decisions decide how to train the model, including batch size, number of epochs, type of loss, type of optimizer, learning rate, reduce rate of learning rate, reduce patience of learning rate, early-stop patience.

Graph learning in architectures

Given a molecular graph $G$with x_i denoting node features of node i and e_{ji} denoting edge features from node j to node i, the feed-forward block can be described as

$$h_i = f_{nn}(x_i) \tag{1.}$$

, where h_i denote the hidden node features and f_{nn} denote a feed-forward neural network. And the message-passing block can be described as

$$h_i' = f_u(h_i, \sum_{j \in N_i} f_i(h_j, e_{ji})) \tag{2.}$$

where f_u denote the update function and f_i denote the interaction function. The output property p are transformed by the global pooling block and final feed-forward block can be described as

$$p = f_{nn}\left(f_{pool}(h_i')\right) \tag{3}$$

where f_{nn} denote a feed-forward neural network and f_{pool} denote the global pooling layer.

Design spaces of blocks

Each block is created with its own design space, as show in **Extended Data Figure 1**. The feed-forward block consists of a normalization layer, a dropout layer, a feed-forward layer, and an activation layer. The normalization layer, dropout layer, and activation layer can be chosen to be empty in this block. Most parts of the message-passing block are the same as the feed-forward block, but the core is changed to a message-passing layer with a choice of five possible types. The fusion block is designed to extract information on a pair of interacting molecules. Global pool block consists of one layer of graph pool layer with a choice of three types of pooling layers.

Preparing for both molecular interactions and properties prediction

We prepare two general architectures for both molecular interactions and properties prediction, as shown in the **Figure 3**. The pair-graph architecture for molecular interactions accepts a pair of molecules as input and outputs their interaction. The single-graph architecture for molecular properties accepts a molecule as input and outputs one or multiple properties of the molecule. Some essential tasks in drug discovery are molecular interactions, such as protein-ligand interactions. All molecules are processed to graphs with basic node attributes as input of the architectures. Small molecules are processed into atom-level molecular graphs, where the edge information is provided by chemical bonds. Proteins are processed into residue-level graphs, where the edge information is provided by contact maps predicted by RaptorX46.

Multi-GPU parallel

GLAM works in parallel with multiple graphics cards. The most time-consuming parts of a graph learning process are the training, validation, and test. The proposed methodhave lots of independent graph learning processes. Subsequently, we let them work in parallel to utilize the computational source fully. In detail, we build a queue and insert all these processes as jobs into it. If a graphics card is free, a job will be popped up and be assigned to the card until all the jobs are popped up.

Robustness experiments using molecular structure perturbations

Molecular structure perturbation experiment aims to investigate the robustness of a model. If the accuracy of a method is not greatly affected by some perturbations, the method is robust enough. A dataset is first divided into training set, validation set and test set. Among them, the training set is used to train a model of the method, the verification set is used to decide when to save the model weights, and the test set is used for robustness experiment.

Given a molecule set of $M = \{m_i | 1 \leq i \leq N\}$ and a method f, we can predict the property set $P = \{p_i | 1 \leq i \leq N\}$ by equation (4). Given a property observer function o, we can calculate a property observer set $Q = \{q_i | 1 \leq i \leq N\}$ by equation (5).

$$P = f(M) \tag{4}$$

$$Q = o(M) \tag{5}$$

Applying an algorithm g to M, we can obtain a perturbed test set $M' = \{m'_i | 1 \leq i \leq N'\}$, where N' is the size of molecules that have been applied g successfully. We also predict the property set $P' = \{p'_i | 1 \leq i \leq N'\}$ by equation (6). Using the observer function o above, we can calculate a property observer set $Q' = \{q'_i | 1 \leq i \leq N'\}$ of M' by equation (7).

$$P' = f(M') \tag{6}$$

$$Q = o(M) \tag{7}$$

By observing the perturbation Effect score of the model after perturbation, we get to know the robustness of a model. Given a loss function L, if $L(P, P') > L(Q, Q')$, the model is not robust, that is, the perturbation will have an essential impact on performance of the model. If $L(P, P') \leq L(Q, Q')$, the model is robust and the perturbation will have less impact. We define a perturbation Effect score by equation (8), where the $\Delta(f, g)$ represents the perturbation Effect score of method f on the perturbation function g.

$$\Delta(f, g) = L(P, P') - L(Q, Q') \tag{8}$$

Property-slightly-affected structural perturbation

Property-slightly-**a**ffected **s**tructural **p**erturbation (PASP) works by iteratively mutating the molecular structure with small perturbations that do not significantly affect the properties of the molecules. In each iteration, PASP mutate the structure of a molecule by modifying/adding a random atom. If the mutated molecule can meet two conditions at the same time in this number of perturbations during the iterations, we got a new perturbed molecular x_i'. The first condition is that the molecular fingerprint similarity $S(x_i, x_i') \in [\gamma_{min}, \gamma_{max})$, where $[\gamma_{min}, \gamma_{max})$ is a preset similarity area. The second condition is that the loss $L(q_i, q_i') < \epsilon_2$, where ϵ_1 is a preset maximum loss. Finally, we combine all the perturbed molecules into a set $M' = \{m_i' | 1 \leq i \leq N'\}$, where N' is the size of all molecules that be perturbed successfully.

Node-level interpretation

We extract the output of the message-passing block, which is a matrix $X = \{x_{ij} | \ 1 \leq i \leq N, 1 \leq j \leq M\}$. N is the number of atoms, and M is the dimension of the outputs. Then we obtain the weight $W = \{w_i \mid 1 \leq 1i \leq N\}$ of each atom according to $w_i = \frac{1}{M}\sum_{j=1}^{M} x_{ij}$, and visualize the molecule with the weights. In some cases, the weight w_i may be scaled to [-1,1] or [0, 1].

Acknowledgements

This work was supported by the National Natural Science Foundation of China (21775060). We acknowledge that the work in this paper was performed at the high-performance computing resources from Supercomputing Center of Lanzhou University. We thank the help from Prof. Jinbo Xu, the author of RaptorX. We thank the help from Mingjian Jiang, the author of DGraphDTA.

Author Contributions

Y.L., C.H. and X. Y. conceived the project, Y.L., C.H., R.L., X.G., X.W., and P.L. designed and conducted the experiments. C.H., S.L. , Y.T. , D.J., J.Y., Q.B., and H.L. evaluated the experiments and contributed ideas. S.Z., C.H. and X.Y. managed and supervised the project. All authors co-wrote the manuscript.

Competing Interests Statement

The authors declare no competing interests.

Data availability

All data used in this paper are publicly available and can be accessed at http://drugdesign.unistra.fr/LIT-PCBA/ for LIT-PCBA dataset. https://github.com/lifanchen-simm/transformerCPI for BindingDB dataset, https://github.com/kexinhuang12345/CASTER for the DrugBank dataset, and http://moleculenet.ai/datasets-1 for all molecules datasets.

Code availability

All code of GLAM is available at https://github.com/yvquanli/GLAM.

References

1. Schneider, G. Automating drug discovery. *Nature Reviews Drug Discovery* **17**, 97–113 (2018).
2. Schneider, P. *et al.* Rethinking drug design in the artificial intelligence era. *Nature Reviews Drug Discovery* **19**, 353–364 (2020).
3. Inglese, J. & Auld, D. S. High Throughput Screening (HTS) Techniques: Applications in Chemical Biology. in *Wiley Encyclopedia of Chemical Biology* (2008). doi:10.1002/9780470048672.wecb223.
4. Sliwoski, G., Kothiwale, S., Meiler, J. & Lowe, E. W. Computational methods in drug discovery. *Pharmacological Reviews* vol. 66 334–395 (2014).
5. Fleming, N. How artificial intelligence is changing drug discovery. *Nature* **557**, S55–S57 (2018).
6. Zheng, S., Li, Y., Chen, S., Xu, J. & Yang, Y. Predicting drug–protein interaction using quasi-visual question answering system. *Nature Machine Intelligence* **2**, 134–140 (2020).

7. Shen, W. X. *et al.* Out-of-the-box deep learning prediction of pharmaceutical properties by broadly learned knowledge-based molecular representations. *Nature Machine Intelligence* (2021) doi:10.1038/s42256-021-00301-6.

8. Kotsias, P.-C. *et al.* Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence* **2**, 254–265 (2020).

9. Méndez-Lucio, O., Baillif, B., Clevert, D. A., Rouquié, D. & Wichard, J. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature Communications* **11**, 1–10 (2020).

10. Chen, H., Engkvist, O., Wang, Y., Olivecrona, M. & Blaschke, T. The rise of deep learning in drug discovery. *Drug Discovery Today* **23**, 1241–1250 (2018).

11. Jiang, S. & Balaprakash, P. Graph Neural Network Architecture Search for Molecular Property Prediction. *arXiv* (2020) doi:10.1109/bigdata50022.2020.9378060.

12. Cai, S. *et al.* Rethinking Graph Neural Architecture Search from Message-passing. (2021).

13. Zhang, Z., Wang, X. & Zhu, W. Automated Machine Learning on Graphs: A Survey. (2021).

14. He, X., Zhao, K. & Chu, X. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems* **212**, (2021).

15. Ekins, S. *et al.* Exploiting machine learning for end-to-end drug discovery and development. *Nature Materials* **18**, 435–441 (2019).

16. Sculley, D. *et al.* Hidden technical debt in machine learning systems. in *Advances in Neural Information Processing Systems* vols. 2015-Janua 2503–2511 (2015).

17. Jiang, M. *et al.* Drug-target affinity prediction using graph neural network and contact maps. *RSC Advances* **10**, 20701–20712 (2020).

18. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. in *International Conference on Learning Representations* (2017).

19. Veličković, P. *et al.* Graph attention networks. *International Conference on Learning Representations* 1–12 (2018).

20. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. in *International Conference on Machine Learning* vol. 3 2053–2070 (2017).

21. Xiong, Z. *et al.* Pushing the boundaries of molecular representation for drug discovery with graph attention mechanism. *Journal of Medicinal Chemistry* acs.jmedchem.9b00959 (2019) doi:10.1021/acs.jmedchem.9b00959.

22. Xu, K., Jegelka, S., Hu, W. & Leskovec, J. How powerful are graph neural networks? in *7th International Conference on Learning Representations, ICLR 2019* (2019).

23. Li, H., Leung, K. S., Wong, M. H. & Ballester, P. J. Improving autodock vina using random forest: The growing accuracy of binding affinity prediction by the effective exploitation of larger data sets. *Molecular Informatics* **34**, 115–126 (2015).

24. Chen, L. *et al.* TransformerCPI: Improving compound–protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. *Bioinformatics* (2020) doi:10.1093/bioinformatics/btaa524.

25. Huang, K., Xiao, C., Hoang, T., Glass, L. & Sun, J. CASTER: Predicting Drug Interactions with Chemical Substructure Representation. *Proceedings of the AAAI Conference on Artificial Intelligence* **34**, 702–709 (2020).

26. Yang, Y.-Y., Rashtchian, C., Zhang, H., Salakhutdinov, R. & Chaudhuri, K. A Closer Look at Accuracy vs. Robustness. 1–14 (2020).

27. Qayyum, A., Qadir, J., Bilal, M. & Al-Fuqaha, A. Secure and Robust Machine Learning for Healthcare: A Survey. *IEEE Reviews in Biomedical Engineering* vol. 14 156–180 (2021).

28. Tetko, I. v., Tanchuk, V. Y. & Villa, A. E. P. Prediction of n-Octanol/Water Partition Coefficients from PHYSPROP Database Using Artificial Neural Networks and E-State Indices. *Journal of Chemical Information and Computer Sciences* **41**, 1407–1421 (2001).

29. Landrum, G. RDKit : A software suite for cheminformatics , computational chemistry , and predictive modeling. https://www.rdkit.org/ (2011).

30. Zeng, Y., Chen, X., Luo, Y., Li, X. & Peng, D. Deep drug-target binding affinity prediction with multiple attention blocks. *Briefings in Bioinformatics* **00**, 1–10 (2021).

31. Withnall, M., Lindelöf, E., Engkvist, O. & Chen, H. Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction. *Journal of Cheminformatics* **12**, 1–18 (2020).

32. Xiong, J., Xiong, Z., Chen, K., Jiang, H. & Zheng, M. Graph neural networks for automated de novo drug design. *Drug Discovery Today* **26**, 1382–1393 (2021).

33. Dai, H. *et al.* Retrosynthesis Prediction with Conditional Graph Logic Network. in *NeurIPS* 1–11 (2020).

34. Wang, X. *et al.* RetroPrime: A Diverse, plausible and Transformer-based method for Single-Step retrosynthesis predictions. *Chemical Engineering Journal* **420**, 129845 (2021).

35. Kuznetsov, M. & Polykovskiy, D. MolGrow: A Graph Normalizing Flow for Hierarchical Molecular Generation. *AAAI 2021* (2021).

36. Luo, Y., Yan, K. & Ji, S. GraphDF: A Discrete Flow Model for Molecular Graph Generation. *ICML 2021* (2021).

37. Liu, M., Yan, K., Oztekin, B. & Ji, S. GraphEBM: Molecular Graph Generation with Energy-Based Models. *ILCR workshop on energy based models* 1–16 (2021).

38. Tran-Nguyen, V. K., Jacquemard, C. & Rognan, D. LIT-PCBA: An unbiased data set for machine learning and virtual screening. *Journal of Chemical Information and Modeling* **60**, 4263–4273 (2020).

39. Gilson, M. K. *et al.* BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research* **44**, D1045–D1053 (2016).

40. Wishart, D. S. *et al.* DrugBank: A knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research* **36**, (2008).

41. Wu, Z. *et al.* MoleculeNet: A benchmark for molecular machine learning. *Chemical Science* **9**, 513–530 (2018).

42. Template-based protein structure modeling using the RaptorX web server. (2012) doi:10.1016/j.physbeh.2017.03.040.

43. Halgren, T. A. *et al.* Glide: A New Approach for Rapid, Accurate Docking and Scoring. 2. Enrichment Factors in Database Screening. *Journal of Medicinal Chemistry* **47**, 1750–1759 (2004).

44. Fey, Matthias and Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. in *International Conference on Learning Representations* (2019).

Table 1. Performance comparison on datasets of molecular interactions

Dataset	ALDH1	ESR1_ant	KAT2A	MAPK1	BindingDB	Params adjust on
Metrics	AUC(higher is better)					
Glide (SP)	0.607(-)	0.590(-)	0.474(-)	0.592(-)	-	-
Glide (XP)	0.582(-)	0.540(-)	0.441(-)	0.579(-)	-	-
RFScore V3	0.556(-)	0.562(-)	0.511(-)	0.640(-)	-	-
DGraphDTA	0.679(0.007)	0.603(0.022)	0.633(0.017)	0.654(0.020)	0.914(0.027)	ALDH1
	0.673(0.013)	0.610(0.011)	0.599(0.032)	0.665(0.031)	0.921(0.023)	BindingDB
TransformerCPI	0.694(0.008)	0.590(0.010)	0.633(0.022)	0.683(0.008)	0.926(0.017)	ALDH1
	0.665(0.008)	0.616(0.032)	0.650(0.042)	0.662(0.012)	0.937(0.016)	BindingDB
GLAM	**0.761**(0.006)	**0.666**(0.012)	**0.709**(0.033)	**0.730**(0.016)	**0.954**(0.008)	None

Note: (a) All datasets are split randomly. (b) Glide[43] docking scores are obtained on Schrödinger version 2015 with the precision of SP and XP. (c) DGraphDTA[17], TransformerCPI[24] are implemented from their open-source code. Their hyper-parameters have been adjusted to obtain the best performance. (d) All deep learning methods are run with three different split seeds, and then we take the average score and the standard deviation (in parentheses). (e) the highlight text with bold-black style means the best.

Table 2. Performance comparison on datasets of molecular properties

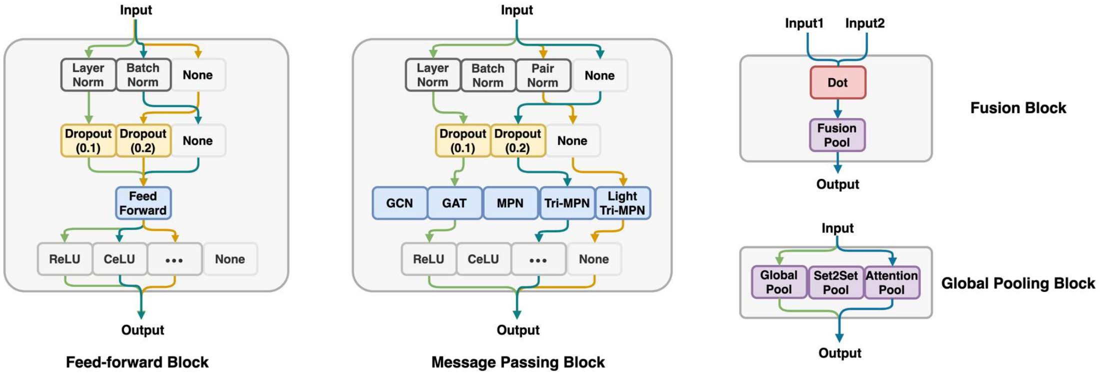
Task	Physical chemistry			Bioactivity	Pharmacokinetics		Toxicity		Params adjust on
Dataset	ESOL	Lipophilicity	FreeSolv	BACE	BBBP	SIDER	Tox21	ToxCast	
Metrics	RMSE (lower is better)			AUC (higher is better)					
GCN	1.017(0.064)	0.807(0.044)	2.307(0.147)	0.772(0.050)	0.830(0.057)	0.619(0.028)	0.762(0.026)	0.678(0.007)	ESOL
	1.056(0.096)	0.799(0.062)	2.858(0.524)	0.797(0.018)	0.792(0.083)	0.612(0.030)	0.752(0.030)	0.663(0.024)	BACE
GAT	1.079(0.080)	0.925(0.031)	2.491(0.465)	0.716(0.033)	0.815(0.067)	0.606(0.053)	0.769(0.033)	0.675(0.013)	ESOL
	1.188(0.058)	0.834(0.037)	2.343(0.272)	0.759(0.019)	0.842(0.042)	0.613(0.023)	0.713(0.022)	0.687(0.017)	BACE
GIN	0.704(0.078)	0.948(0.058)	2.662(0.289)	0.812(0.032)	0.858(0.035)	0.595(0.031)	0.793(0.026)	0.671(0.019)	ESOL
	0.742(0.058)	0.864(0.044)	2.098(0.272)	0.831(0.040)	0.875(0.019)	0.605(0.040)	0.732(0.019)	0.672(0.016)	BACE
MPNN	0.755(0.077)	0.769(0.031)	1.897(0.092)	0.820(0.047)	0.831(0.036)	0.626(0.038)	0.752(0.011)	0.697(0.023)	ESOL
	0.884(0.061)	0.825(0.047)	2.038(0.421)	0.816(0.042)	0.816(0.057)	0.641(0.014)	0.803(0.031)	0.686(0.017)	BACE
AttentiveFP	0.726(0.032)	0.724(0.030)	1.775(0.392)	0.815(0.072)	0.856(0.023)	0.654(0.027)	0.763(0.022)	0.726(0.020)	ESOL
	0.738(0.059)	0.783(0.036)	1.371(0.446)	0.850(0.017)	0.872(0.024)	0.621(0.033)	0.740(0.039)	0.680(0.024)	BACE
MolMapNet	0.752(0.040)	0.731(0.012)	1.398(0.312)	0.868(0.094)	0.911(0.013)	0.634(0.015)	0.813(0.021)	0.703(0.025)	None
GLAM	**0.592**(0.036)	**0.596**(0.025)	**1.319**(0.346)	**0.888**(0.033)	**0.932**(0.015)	**0.659**(0.017)	**0.841**(0.010)	**0.744**(0.008)	None

Note: (a) All datasets are split by scaffold. (b) GCN[18], GAT[19], GIN[22], MPNN[20] are implemented with PyTorch Geometric[44]. AttentiveFP[21] is implemented from its open-source code. Their hyper-parameters have been adjusted to obtain the best performance. (c) All methods are run with three different split seeds, and then we take the average score and the standard deviation (in parentheses). (d) the highlight text with bold-black style means the best.

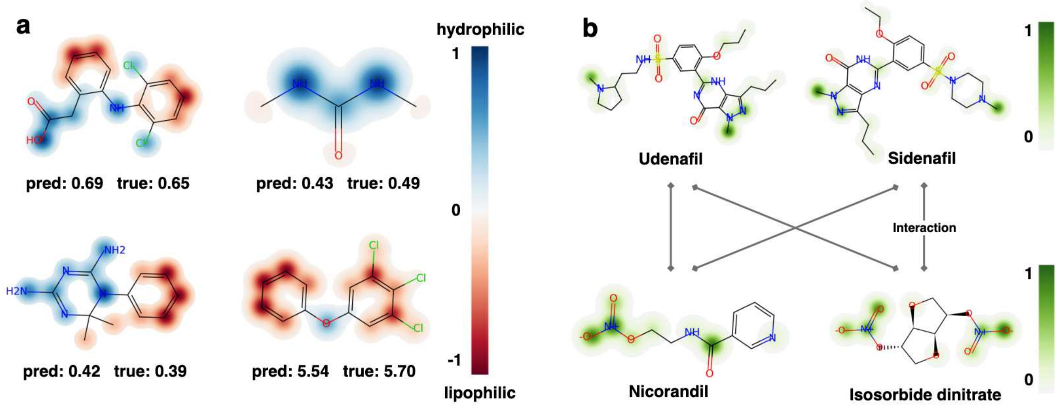
Table 3. Effect score of molecular structure perturbation test

Perturbation Level	**Level 1**	**Level 2**	**Level 3**
Fingerprint Similarity	**0.8-1.0**	**0.5-0.8**	**0.3-0.5**
Metrics	**Effect score (lower is better)**		
GCN	0.728(±0.019)	0.929(±0.015)	1.092(±0.009)
GAT	0.750(±0.022)	0.985(±0.118)	1.149(±0.124)
GIN	0.695(±0.064)	0.953(±0.100)	1.087(±0.074)
MPNN	0.598(±0.018)	0.820(±0.069)	0.998(±0.083)
GLAM	**0.360(±0.014)**	**0.469(±0.019)**	**0.632(±0.013)**

Note: (a)All baselines above are all implemented with PyTorch Geometric[44], and their hyper-parameters have been manually adjusted for several rounds. (b) The losses between P and P' of level 1,2,3 perturbated test sets are 0.0624, 0.0593, 0.0578, respectively.



Extended Data Figure 1. design space for blocks of the architecutures.



Extended Data Figure 2. Node-level interpretation. **a**, case studies of solubility prediction. **b** case studies of drug-drug interactions.

Extended Data Table 1. Performance comparison on drug-drug interaction datasets

Dataset	DrugBank	
Metrics	**ROC-AUC**	**PR-AUC**
LR	0.774(±0.003)	0.745(±0.005)
Nat.Prot	0.786(±0.003)	0.753(±0.003)
Mol2Vec	0.849(±0.004)	0.828(±0.006)
MOLVAE	0.852(±0.006)	0.828(±0.009)
DeepDDI	0.844(±0.003)	0.828(±0.002)
CASTER	0.861(±0.005)	0.829(±0.003)
GLAM	**0.911(±0.017)**	**0.899(±0.019)**

Note (a) In the drug-drug interaction task, scores of all baselines are taken from CASTER[25]. (b) the highlight score with bold-black style means this method got the highest score. (c) all scores of GLAM are run under the same default configuration setting.