nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-022-00501-8

Check for updates

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

Yuquan Li[1,2,9], Chang-Yu Hsieh[2,9], 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] and Xiaojun Yao[1,3] ✉

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 their architectures and hyperparameters, 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 takes a solid step forward to the purpose of aiding researchers to design better drugs with high efficiency.

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] have been 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 because of their ability to amplify insights available from existing drug-related datasets[5]. Using the insights to predict molecular interactions and properties[6,7] is 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 their architectures and hyperparameters, making them lose advantage in repurposing to new data generated during drug discovery. In practice, researchers tend to crystallize, that is, select and optimize architectures and hyperparameters from a huge design space to achieve the best performance on a dataset[11–13]. This heavily limits repurposing to newly generated data, which tend to be increasingly complex[14] in drug discovery. Most graph learning methods rely heavily on these architectures and hyperparameters to achieve their claimed state-of-the-art results, and if the author does not release these specific architectures and hyperparameters, their claimed state-of-the-art results cannot be reproduced[15].

Recently, a few works have been reported to address the crystallization problem. MolMapNet introduced an out-of-the-box deep learning method based on broadly learning knowledge-based representations to achieve reliable prediction performance on more datasets without human intervention[7]. A recent work also introduced a neural architecture search-based method to design a neural architecture for any dataset of molecular property prediction automatically[11].

In this Article we propose 'graph learning-based adaptive machine' (GLAM), a flexible method that can adapt to any dataset and make accurate predictions without human intervention. We compared our proposed method with previously reported methods in terms of prediction performance, on a wide range of datasets. The results show that our proposed method can adapt to all tested datasets exceptionally well and obtain far better prediction performance than the other reported methods. We also investigated the robustness and interpretability of our proposed method, and found that it was more robust than the other tested methods and can provide meaningful interpretability, making it a more reliable method.

Results

Method overview. Our method utilizes an automated pipeline to learn from datasets and build a predictor, as shown in Fig. 1. Previous graph learning methods[16–21] rely heavily on human experts to design the architecture, set the model hyperparameters, select the optimizer and select the loss function. We creatively combine these four items and build a configuration space. Starting from this configuration space, GLAM performs a series of processes to build a blended predictor, as shown in Fig. 2.

[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. [9]These authors contributed equally: Yuquan Li, Chang-Yu Hsieh. ✉e-mail: xjyao@lzu.edu.cn

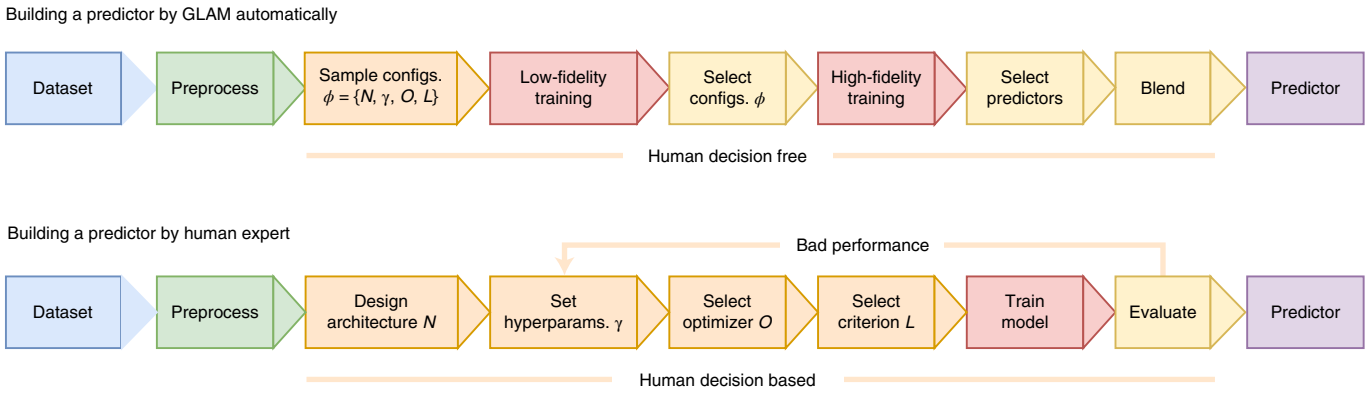


Fig. 1 | Overview of GLAM and the traditional method. GLAM performs configuration sampling, low-fidelity training based on sampled configurations, configuration selection, high-fidelity training based on selected configurations, predictor selection and predictor blending to build a predictor automatically. With traditional methods human experts perform architecture design, tune model hyperparameters, select the optimizer and select the loss function.

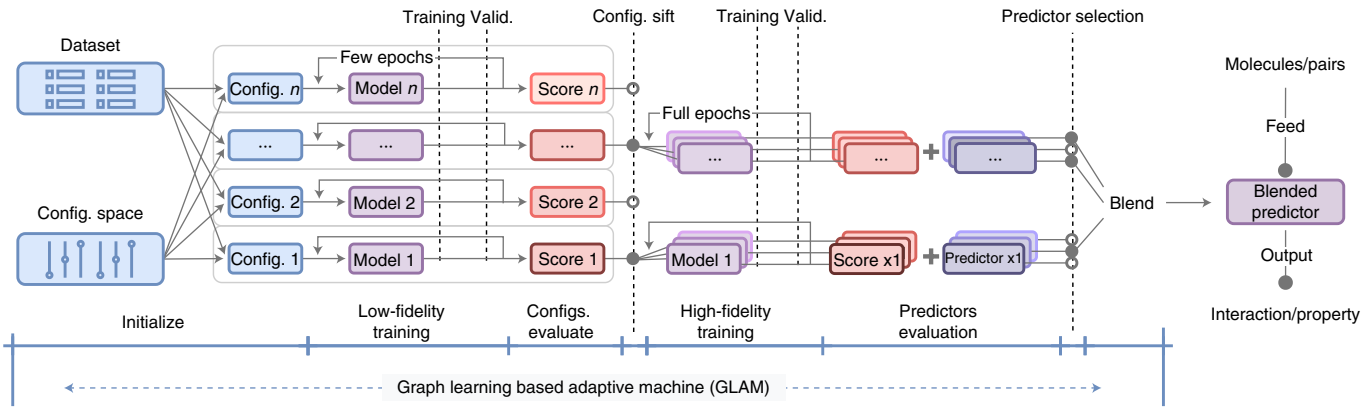


Fig. 2 | GLAM pipeline details. Initially, GLAM samples a lot of configurations from the configuration space. The dataset is then fed to these configurations for low-fidelity training, which produces evaluation scores to select the high-performance configurations. In the next high-fidelity training, each previously selected configuration is fed with the dataset to perform full epochs training with three different random seeds. High-fidelity training can produce predictors and the corresponding validation scores used for top-performance predictors selection. The number of selected predictors is predefined by the ensemble size. Finally, all selected predictors are blended into a predictor. Low-fidelity training refers to a rapid training process with few epochs to obtain validation scores for all configurations to estimate their prediction performance quickly. High-fidelity training refers to a slow training process with enough epochs to accurately estimate the prediction performance of selected configurations.

We designed two general architectures, one for molecular interaction and another for molecular property, as shown in Fig. 3. Each block in the general architecture is created with its own design space, as shown in Extended Data Fig. 1. These architectures take graphs, including molecular graphs and protein graphs, as input. A molecular graph is constructed with atoms as nodes and bonds as edges. A protein graph is constructed with amino acids as nodes and contact information calculated by RaptorX[22] as edges. The architecture takes a molecular graph and protein graph as input when performing drug–target interaction tasks, two molecular graphs as input when performing drug–drug interaction tasks, and a single molecular graph as input when performing molecular property prediction tasks.

Adapt to datasets for high performance. GLAM is designed to adapt to any given dataset to obtain a 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[7,16–20,23–25]. The types of tested dataset include drug–protein interactions, drug–drug interactions, physical–chemistry property, bioactivity, pharmacokinetics and toxicity. Given that different splits of datasets produce different performances, we let all methods share the same splits of datasets so as to obtain a fair evaluation. We also manually adjusted the architectures and hyperparameters of previously reported methods to achieve their best performance on two representative datasets. Finally, we ran benchmarks and analysed their adaptability and performance on these datasets.

Compared to all traditional methods, our proposed methods can adapt to datasets well and achieve promising prediction performance, as shown in Tables 1 and 2 and Supplementary Table 1. GLAM thus establishes a new state of the art for both molecular interactions and properties prediction. Relative to the best scores in previously reported results, the proposed method achieved an average 18.7% decrease in prediction error on 14 datasets compared to the best of the traditional methods. In addition, GLAM can consistently achieve the best scores without human intervention, whereas previously reported methods achieve uneven prediction performances. GLAM is therefore poised to be a flexible, reliable and trustworthy method that works well across a wide range of activities in drug design.

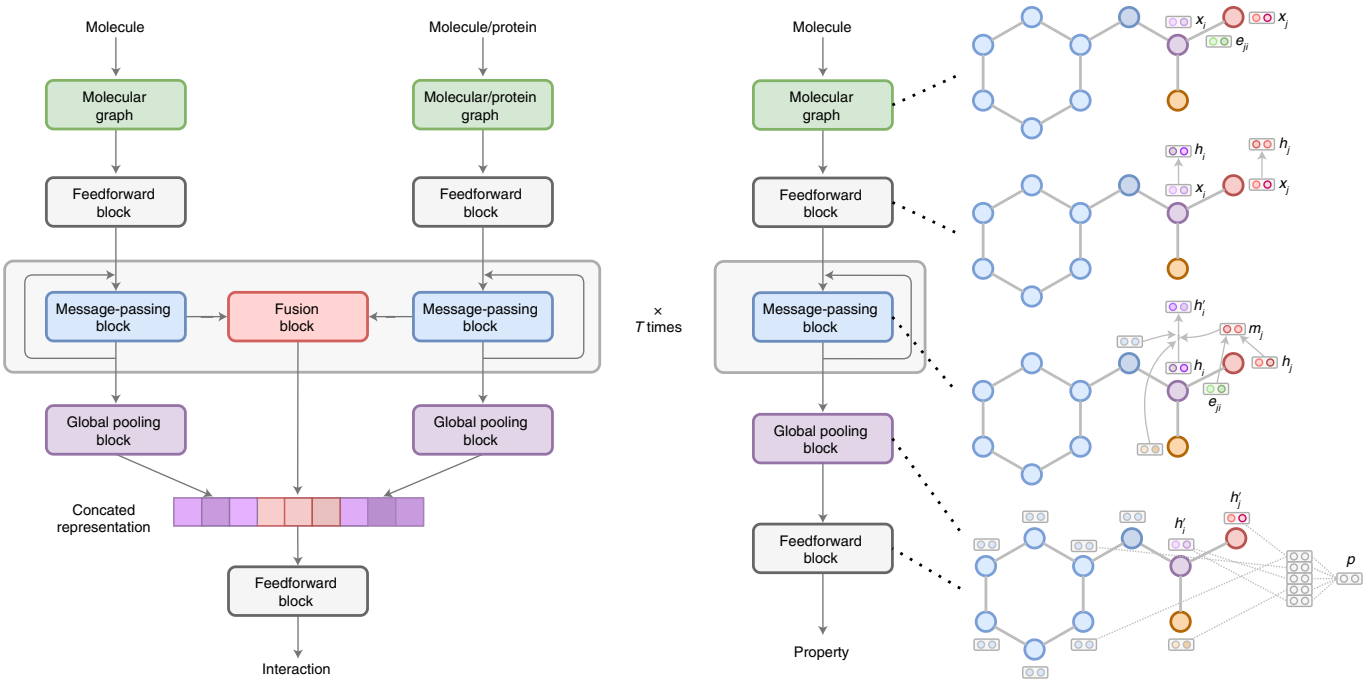


Fig. 3 | The general architectures for molecular interactions and properties prediction in GLAM. The input data are mapped to the output by five steps: (1) preprocessing a molecule to a molecular graph, (2) linear transformation by a feedforward block, (3) iterated convolution by a message-passing block for T time steps, (4) node reduction by a global pooling block and (5) linear transformation by a final feedforward block. See the Methods for an explanation of the variables.

Table 1 | Performance comparison on datasets of molecular interactions

Dataset	ALDH1	ESR1_ant	KAT2A	MAPK1	BindingDB	Parameters adjust on
Metrics	AUC (s.d.) [higher is better]					
Glide (SP)	0.607 (–)	0.590 (–)	0.474 (–)	0.592 (–)	-	-
Glide (XP)	0.582 (–)	0.540 (–)	0.441 (–)	0.579 (–)	-	-
RFScore-VS	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

(1) These datasets follow the splits of previous works. ALDH1, ESR1_ant, KAT2A and MAPK1 were released with official splits that are unbiased. BindingDB was split by the authors of TransformerCPI. (2) Glide[44] docking scores were obtained on Schrödinger version 2015 with the precision of SP and XP. (3) RFScore-VS is a novel Random Forest-based scoring function for virtual screening (VS) that predicts binding affinity. The RFScore-VS features for the datasets were calculated by the Open Drug Discovery Toolkit (ODDT) Python package, and model training was implemented by the scikit-learn Python package. (4) DGraphDTA[16] and TransformerCPI[24] were implemented from their open-source code. Their hyperparameters were adjusted to obtain the best performance. (5) All deep learning methods were run with three different split seeds, then we took the average score and the s.d. (in parentheses). (6) The bold entries are the best results. (7) In this table, the ensemble size used for GLAM is 3. (8) The benchmarks used only single-target datapoints for the training process on ALDH1, ESR1_ant, KAT2A and MAPK1. AUC, area under the curve.

High robustness against molecular structure perturbation. The next issue to consider is robustness[26], another essential indicator of good practice in method development. We assume that a robust predictor should not change its output greatly when a small perturbation of structures that has little effect on the molecular property is applied. Natural perturbations can always affect a machine learning method, and may lead to a wrong result with serious consequences in some safety-sensitive domains (such as healthcare). Admittedly, the robustness of a graph learning method is also an essential issue.

To evaluate the robustness of our proposed method, we first introduced a principle termed property-slightly-affected structure perturbation (PASP), then built a real-world perturbed dataset following this principle from the PhysProp[27] dataset. We then performed a robustness experiment based on the dataset. More details are provided in the Methods.

Table 3 shows that GLAM is less affected by molecular structure perturbations and demonstrates higher robustness than the conventional methods. We fed original molecules and perturbed molecules to the predictor and investigated the differences in the outputs to measure the effect score. Compared with conventional methods, our method is less affected by PASP than conventional methods on all three levels of PASP. The robustness of GLAM is most probably 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

Table 2 | Performance comparison on datasets of molecular properties

Task	Physical chemistry			Bioactivity	Pharmacokinetics		Toxicity		Parameters adjust on
Dataset	ESOL	Lipophilicity	FreeSolv	BACE	BBBP	SIDER	Tox21	ToxCast	
Metrics	R.m.s.e. (s.d.) [lower is better]			AUC (s.d.) [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

(1) All datasets are split by scaffold. (2) GCN[17], GAT[18], GIN[21] and MPNN[19] are implemented with PyTorch Geometric[45]. AttentiveFP[20] is implemented from its open-source code. Their hyperparameters have been adjusted to obtain the best performance. (3) All methods are run with three different split seeds, and then we take the average score and the s.d. (in parentheses). (4) The bold entries are the best results. (5) In this table, the ensemble size used for GLAM is 3. R.m.s.e., root-mean-square error.

from averaging. So, perturbing the molecular structure may affect the individual predictors but not the blended model.

Interpretation cases. To better understand the predictors generated by GLAM, we investigated its decision-making process and interpreted its learned knowledge. In the past, most machine learning models have been considered as black boxes. Previous works have adopted attention mechanisms[20,28,29] to aid interpretation of models. Here we explain the model from the hidden states by averaging and visualizing it, thus directly utilizing the information provided by the models in the predictor, as shown in Extended Data Fig. 2.

Extended Data Fig. 2a presents 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 selected and visualized 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. Meanwhile, 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 the predictor can detect essential atomic groups with clear interpretability of their solubility.

In the same way, we also visualized some drug–drug interaction identification cases, as shown in Extended Data Fig. 2b. We considered the interactions between sidenafil/udenafil and nitrates (nicorandil/isosorbide dinitrate) as cases. These were combined into four pairs of drug–drug interactions and fed into models in the predictor to visualize the decision process. (Typically, sildenafil/udenafil can selectively inhibit phosphodiesterase type 5 (PDE5) targets in the human body. The *N*-methyl groups in the pyrazolopyrimidone rings of sildenafil/udenafil are important for the activity and selectivity of PDE5. For these reasons, combining sildenafil/udenafil with nitrate drugs may lead to serious drug-drug interactions. As a result, the two kind of drugs may have interactions when they are combined.) The visualization results show that the models in the predictor pay more attention to the nitrates of isosorbide dinitrate and nicorandil, and more attention to the *N*-methyl groups of sildenafil and udenafil. Accordingly, our visualization results are consistent with previous findings for these drug interactions, indicating that the models in the predictor can provide deep insights into molecular interactions.

Ablation studies. *Time consumption and resource cost.* We analysed the time consumption and resources costs of GLAM and other methods, as shown in Supplementary Table 2. We analysed multiple aspects, including computing time, computing device, training details and dataset size. From the comparison, GLAM costs ~10 times more in terms of time consumption and four times more resources. We believe that these computing resources and time are worth it for the high performance achieved.

Preferences analysis. The preferences of GLAM may lead to ideas for the design of new methods, so we analysed the preferences of GLAM in multiple configuration items, as shown in Supplementary Table 3. These analyses were performed on three representative datasets: ESOL, BBBP and Tox21. GLAM prefers the global pool method and Adam optimizer on all tested datasets. In addition, GLAM may prefer message-passing networks (MPNs) on small datasets and prefers complex cores on bigger datasets for the choice of message-passing cores. On the choice of message-passing steps, GLAM may prefer bigger steps on bigger datasets.

Ensemble size. It is well known that ensembling models can create better predictors. We designed and conducted two experiments to investigate the effect of ensemble size on performance and robustness, as shown in Table 3 and Supplementary Table 4. From Supplementary Table 4, we can see that the performance improved as the ensemble size increased from 1 to 3, and remained stable from 3 to 7. On the other hand, we can see that a bigger ensemble size, from 1 to 7, always brings less effect from PASP, as shown in Table 3. We also tested many ensembled competing graph models, such as 3*MPNN and GCN+GIN+MPNN. Their performance and robustness improved, but were still not as good as those of GLAM.

Comparison with Auto-Sklearn[30] and Auto-Gluon[31]. It is now common to feed an automated machine learning method[30,31] with structured data to obtain an excellent predictor. Auto-Sklearn and Auto-Gluon are well-known examples. However, we cannot make a fair comparison with them because they are very different in terms

Table 3 | Effect score of the molecular structure perturbation test

Method	Effect score (s.d.) [lower is better]		
	Level 1	Level 2	Level 3
GCN	0.385 (0.161)	0.712 (0.169)	0.997 (0.183)
3*GCN	0.288	0.543	0.740
GAT	0.388 (0.055)	0.615 (0.087)	0.943 (0.145)
3*GAT	0.245	0.394	0.634
GIN	0.312 (0.017)	0.526 (0.039)	0.764 (0.015)
3*GIN	0.249	0.435	0.673
MPNN	0.315 (0.014)	0.518 (0.054)	0.750 (0.048)
3*MPNN	0.272	0.457	0.679
GCN + GAT + GIN	0.290	0.524	0.674
GCN + GAT + MPNN	0.290	0.507	0.652
GCN + GIN + MPNN	0.280	0.518	0.640
GAT + GIN + MPNN	0.256	0.428	0.610
GLAM (best, $n=1$)	0.290 (0.010)	0.493 (0.074)	0.656 (0.118)
GLAM ($n=2$)	0.276 (0.025)	0.468 (0.080)	0.617 (0.127)
GLAM ($n=3$)	0.259 (0.004)	0.418 (0.069)	0.581 (0.097)
GLAM ($n=5$)	0.220 (0.047)	0.368 (0.020)	0.513 (0.060)
GLAM ($n=7$)	0.200 (0.058)	0.353 (0.013)	0.502 (0.018)

(1) All baselines above are implemented with PyTorch Geometric[45], and their hyperparameters have been manually adjusted for several rounds. (2) The losses between the ground-truth labels P and the predicted labels P′ of level 1, 2 and 3 perturbated test sets are 0.0624, 0.0593 and 0.0578, respectively.

of their accepted data and objective optimization. Auto-Sklearn and Auto-Gluon take structured data (images and so on) as input and optimize independent machine learning models with hyperparameters. GLAM takes multiple unstructured data (double/single molecular graphs) as input and optimizes configurations consisting of architectures, hyperparameters, optimizers and losses. Despite this, we processed the molecules into structured data (molecular fingerprints[32]) and fed them to AutoMLs so that we could make a comparison with GLAM, as shown in Supplementary Table 5. Although their performance is not as good as that of GLAM, they have a considerable advantage in their computational speed and cost.

Discussion

We have shown that GLAM can adapt well to all tested datasets and make accurate predictions automatically. In the past, adaptability to new data has largely been ignored, as researchers have addressed almost all their attention to achieving a 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.

Limitations and frontiers

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

A more intelligent hyperparameter optimization algorithm built on multi-graph cards may increase the configuration search efficiency. The current version of GLAM uses a basic optimizer based on random search. If a more intelligent optimizer is embedded into GLAM, it will help GLAM find ideal configurations in less time.

Outlook

Our method is expected to advance and evolve automated drug design[1,33]. Recent advances, such as chemical retrosynthesis predictions[34,35] and molecular generations[36–38], 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 that will contribute to 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 used LIT-PCBA[39], BindingDB[40] and DrugBank[41] to evaluate our proposed method for molecular interaction prediction. From LIT-PCBA we selected four datasets of representative proteins based on the number of positive and negative samples. We used datasets in MoleculeNet[42] to evaluate the proposed method for molecular property prediction. MoleculeNet[42] is a set of benchmarking datasets for molecular machine learning that can be used to achieve a fair performance comparison.

LIT-PCBA is a virtual screening dataset consisting of 14 targets, 7,844 confirmed active and 407,381 confirmed inactive compounds[39]. The BindingDB dataset contains 39,747 positive examples and 31,218 negative examples from a public database[40]. DrugBank includes 1,850 approved drugs with 221,523 drug-drug interaction (DDI) positive labels[41]. The Blood–Brain Barrier Penetration (BBBP) dataset contains 2,053 molecules and their permeability properties[42]. SIDER is a database of marketed drugs and adverse drug reactions, grouped into system organ classes for 1,427 approved drugs[42]. BACE is a database consisting of binding results for a set of inhibitors of human β-secretase 1 with 1,522 compounds[42]. Tox21 contains qualitative toxicity measurements for 8,014 compounds on 12 different targets, including stress response pathways and nuclear receptors[42]. ToxCast is another toxicity database that contains qualitative results from 617 experiments on 8,615 compounds[42]. The Free Solvation Database (FreeSolv) provides experimental and calculated hydration free energies of 643 small molecules in water[42]. ESOL is a small dataset consisting of water solubility data for 1,128 compounds[42]. Lipophilicity is a dataset that contains experimental results of octanol/water distribution coefficient (logD at pH 7.4) of 4200 compounds[42]. PhysProp consists of 14,176 molecules and their corresponding log P values[27].

Configuration space. The configuration space of GLAM has two parts: architecture decisions and training decisions. The architecture decisions decide how to build the architecture of a model. The general architectures of pairs of molecules and single molecules contain eight and four blocks with their own independent design spaces, respectively, as shown in Fig. 1. The training decisions decide how to train the model, including batch size, number of epochs, type of loss, type of optimizer, learning rate, reduction of learning rate, reduction of the patience of learning rate and 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 feedforward block can be described as

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

where h_i denotes the hidden node features and f_{nn} a feedforward neural network. The message-passing block can be described as

$$h_i' = f_{\mathrm{u}}\left(h_i, \sum_{j \in N_i} f_{\mathrm{i}}(h_j, e_{ji})\right) \tag{2}$$

where f_{u} denotes the update function and f_{i} the interaction function. The output properties p are transformed by the global pooling block and the final feedforward block can be described as

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

where f_{nn} denotes a feedforward neural network and f_{pool} the global pooling layer.

Design spaces of blocks. Each block is created with its own design space, as shown in Extended Data Fig. 1. The feedforward block consists of a normalization layer, a dropout layer, a feedforward 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 in the feedforward 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. The global pool block consists of one layer of graph pool layer with a choice of three types of pooling layer.

Preparing for both molecular interactions and properties. We prepare two general architectures for both molecular interactions and properties prediction, as shown in Fig. 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 relate to 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, with the edge information provided by contact maps predicted by RaptorX[22].

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

Robustness experiments. This experiment aims to investigate the robustness of a predictor based on the perturbed dataset. If the prediction of a method is not greatly affected by a slight perturbation that has little effect on the molecular property, the method may be a robust method.

Given a molecule set M with ground-truth properties Q and a trained predictor f, we predict the property set P by equation (4), and the training and validation sets are used to train and save weights to obtain our predictor f:

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

Given the perturbed test set M' with properties Q', we predict the property set P' by equation (5):

$$P' = f(M') \tag{5}$$

We estimate the robustness of the predictor by calculating the error between the predicted value P of the original molecule set M and the predicted value P' of the perturbed molecule set M'. Given a distance function L, $L(P, P')$ represents the distance between P and P'. If $L(P, P') > L(Q, Q')$, the predictor is not robust, that is, the perturbation will have a big impact on the performance of the predictor. If $L(P, P') \leq L(Q, Q')$, the predictor is robust and the perturbation will have less impact. We define a perturbation effect score with equation (6), where the Δ represents the perturbation effect score of method f:

$$\Delta = L(P,\ P') - L(Q,\ Q') \tag{6}$$

In this work, we use the following settings. We picked out the original set M (N=2,362) and the perturbed set M' (N=2,362, 2,362, 2,362 for level 1, 2, 3) from all 14,176 molecules, and we use the r.m.s.e. as our loss function L.

PASP. This principle is used to determine an ideal perturbed molecule set with small perturbations that do not significantly affect the properties. We need to ensure two conditions are met to let PASP work. The first condition is that the change in property should be within an acceptable range, and the second condition is that the molecular structures do not change much.

Given a molecule pair $\{x_i,\ x_i'\}$, their properties are $\{q_i,\ q_i'\}$. Assume their molecular fingerprint similarity is $S(x_i,\ x_i') \in [\gamma_{\min},\ \gamma_{\max})$, where $[\gamma_{\min},\ \gamma_{\max})$ is a predefined similarity range, and the difference of their properties is $L(q_i,\ q_i') < \epsilon_2$, where ε_1 is a predefined acceptable value, then the molecular pair is an ideal perturbed molecule pair that follows the principle.

Building the perturbed dataset. In this experiment we build a perturbed dataset based on real-world datapoints by searching and selecting from the PhysProp[27] dataset for robustness estimation. The PhysProp dataset consists of 14,176 molecules structures and their corresponding properties ($\log P$). We compare all potential molecule pairs, calculate the fingerprint similarity of all molecules and their difference in $\log P$, and pick out molecule pairs that meet the following two conditions. The first condition is that the difference in the $\log P$ of the molecule pairs should be less than 0.2, and the second condition is that the molecular fingerprint similarity should be in the range of 0.3–1.0. We then divide these molecule pairs into three levels (range 0.8–1.0, 0.5–0.8 and 0.3–0.5, marked as levels 1, 2 and 3), pick out those molecules that exist in all three levels, and build the test set (N=2,362) with them. The corresponding molecules in the pairs of three levels are separated to build the perturbed test sets (N=2,362, 2,362 and 2,362) of the three levels. All remaining molecules are used to build the training set (N=7,684) and test set (N=2,561) to train the models and save the model weights.

Node-level interpretation. We extract the output of the message-passing block of the best predictor generated by GLAM, which is a matrix $X = \{x_{ij} | 1 \leq i \leq N,\ 1 \leq j \leq M\}$, where N is the number of atoms and M the dimension of the outputs. We then 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 $[0, 1]$.

Data availability

All data used in this paper are publicly available and can be accessed as follows: LIT-PCBA[39] (ALDH1, ESR1_ant, KAT2A, MAPK1), BindingDB[40], DrugBank[41], MoleculeNet[42] (ESOL, Lipophilicity, FreeSolv, BACE, BBBP, SIDER, Tox21, ToxCast) and Perturbed PhysProp[43].

Code availability

All code of GLAM is freely available at https://github.com/yvquanli/GLAM with an MIT licence. The version used for this publication is available at https://doi.org/10.5281/zenodo.6371164[43].

Received: 15 December 2021; Accepted: 16 May 2022;
Published online: 23 June 2022

References

1. Schneider, G. Automating drug discovery. *Nat. Rev. Drug Discov.* **17**, 97–113 (2018).
2. Schneider, P. et al. Rethinking drug design in the artificial intelligence era. *Nat. Rev. Drug Discov.* **19**, 353–364 (2020).
3. Inglese, J. & Auld, D. S. in *Wiley Encyclopedia of Chemical Biology* (ed. Begley, T. P.) (Wiley, 2008); https://doi.org/10.1002/9780470048672.wecb223
4. Sliwoski, G., Kothiwale, S., Meiler, J. & Lowe, E. W. Computational methods in drug discovery. *Pharmacol. Rev.* **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. *Nat. Mach. Intell.* **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. *Nat. Mach. Intell.* **3**, 334–343 (2021).
8. Kotsias, P.-C. et al. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nat. Mach. Intell.* **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. *Nat. Commun.* **11**, 10 (2020).
10. Chen, H., Engkvist, O., Wang, Y., Olivecrona, M. & Blaschke, T. The rise of deep learning in drug discovery. *Drug Discov. Today* **23**, 1241–1250 (2018).
11. Jiang, S. & Balaprakash, P. Graph neural network architecture search for molecular property prediction. In *Proc. IEEE International Conference on Big Data* 1346–1353 (IEEE, 2020).
12. Cai, S., Li, L., Deng, J., Zhang, B., Zha, Z. J., Su, L., & Huang, Q. Rethinking Graph Neural Architecture Search from Message-passing. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6653–6662. https://doi.org/10.1109/CVPR46437.2021.00659 (2021).
13. Zhang, Z., Wang, X., & Zhu, W. Automated Machine Learning on Graphs: A Survey. *IJCAI International Joint Conference on Artificial Intelligence*, 4704–4712. https://doi.org/10.24963/ijcai.2021/637 (2021)
14. Ekins, S. et al. Exploiting machine learning for end-to-end drug discovery and development. *Nat. Mater.* **18**, 435–441 (2019).
15. Sculley, D. et al. Hidden technical debt in machine learning systems. In *Proc. Advances in Neural Information Processing Systems*Vol. 2015-January, 2503–2511 (NIPS, 2015).
16. Jiang, M. et al. Drug–target affinity prediction using graph neural network and contact maps. *RSC Adv.* **10**, 20701–20712 (2020).
17. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *Proc. 2017 International Conference on Learning Representations* (ICLR, 2017).
18. Veličković, P. et al. Graph attention networks. In *Proc. 2018 International Conference on Learning Representations* 1–12 (ICLR, 2018).
19. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *Proc. International Conference on Machine Learning* Vol. 3, 2053–2070 (ACM, 2017).

20. Xiong, Z. et al. Pushing the boundaries of molecular representation for drug discovery with graph attention mechanism. *J. Med. Chem.* https://doi.org/10.1021/acs.jmedchem.9b00959 (2019).
21. Xu, K., Jegelka, S., Hu, W. & Leskovec, J. How powerful are graph neural networks? In *Proc. 7th International Conference on Learning Representations*, *ICLR 2019* (ICLR, 2019).
22. Källberg, M. et al. Template-based protein structure modeling using the RaptorX web server. *Nat. Protoc.* **7**, 1511–1522 (2012).
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. *Mol. 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* **36**, 4406–4414 (2020).
25. Huang, K., Xiao, C., Hoang, T., Glass, L. & Sun, J. CASTER: predicting drug interactions with chemical substructure representation. *Proc. AAAI Conf. Artif. Intell.* **34**, 702–709 (2020).
26. Yang, Y.-Y., Rashtchian, C., Zhang, H., Salakhutdinov, R. & Chaudhuri, K. A closer look at accuracy vs. robustness. In *Proc. 34th International Conference on Neural Information Processing Systems* Vol. 720, 8588–8601 (NIPS, 2020).
27. 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. *J. Chem. Inf. Comput. Sci.* **41**, 1407–1421 (2001).
28. Zeng, Y., Chen, X., Luo, Y., Li, X. & Peng, D. Deep drug–target binding affinity prediction with multiple attention blocks. *Briefings Bioinform.* **22**, bbab117 (2021).
29. Withnall, M., Lindelöf, E., Engkvist, O. & Chen, H. Building attention and edge message passing neural networks for bioactivity and physical-chemical property prediction. *J. Cheminform.* **12**, 1–18 (2020).
30. Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M. & Hutter, F. Auto-Sklearn 2.0: the next generation (2020); https://www.researchgate.net/publication/342801746_Auto-Sklearn_20_The_Next_Generation
31. Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. *ICML Workshop on Automated Machine Learning* (2020).
32. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
33. Xiong, J., Xiong, Z., Chen, K., Jiang, H. & Zheng, M. Graph neural networks for automated de novo drug design. *Drug Discov. Today* **26**, 1382–1393 (2021).
34. Dai, H. et al. Retrosynthesis prediction with conditional graph logic network. In *Proc. 33rd International Conference on Neural Information Processing Systems* Vol. 796, 8872–8882 (NIPS, 2020).
35. Wang, X. et al. RetroPrime: a diverse, plausible and transformer-based method for single-step retrosynthesis predictions. *Chem. Eng. J.* **420**, 129845 (2021).
36. Kuznetsov, M. & Polykovskiy, D. MolGrow: a graph normalizing flow for hierarchical molecular generation. In *Proc. AAAI Conference on Artificial Intelligence* Vol. 35, 8226–8234 (AAAI, 2021).
37. Luo, Y., Yan, K. & Ji, S. GraphDF: a discrete flow model for molecular graph generation. In *Proc. 38th International Conference on Machine Learning*, PMLR Vol. 139, 7192–7203 (PMLR, 2021).
38. Liu, M., Yan, K., Oztekin, B. & Ji, S. GraphEBM: molecular graph generation with energy-based models. *Proc. ILCR Workshop on Energy Based Models* 1–16 (2021).
39. Tran-Nguyen, V. K., Jacquemard, C. & Rognan, D. LIT-PCBA: an unbiased data set for machine learning and virtual screening. *J. Chem. Inf. Model.* **60**, 4263–4273 (2020).
40. Gilson, M. K. et al. BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res.* **44**, D1045–D1053 (2016).
41. Wishart, D. S. et al. DrugBank: a knowledge base for drugs, drug actions and drug targets. *Nucleic Acids Res.* **36**, D901–D906 (2008).
42. Wu, Z. et al. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
43. Li, Y. Code for 'An adaptive graph learning method for automated molecular interactions and properties predictions' (Zenodo, 2022); https://doi.org/10.5281/zenodo.6371164
44. Halgren, T. A. et al. Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. *J. Med. Chem.* **47**, 1750–1759 (2004).
45. Fey, M. & Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *Proc. ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds* (ICLR, 2019); https://arxiv.org/abs/1903.02428

Acknowledgements

This work was supported by the National Natural Science Foundation of China (22173038 and 21775060). We thank the Supercomputing Center of Lanzhou University for providing high-performance computing resources. We acknowledge help from J. Xu, the author of RaptorX[22], as well as help from M. Jiang, the author of DGraphDTA[16].

Author contributions

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

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-022-00501-8.

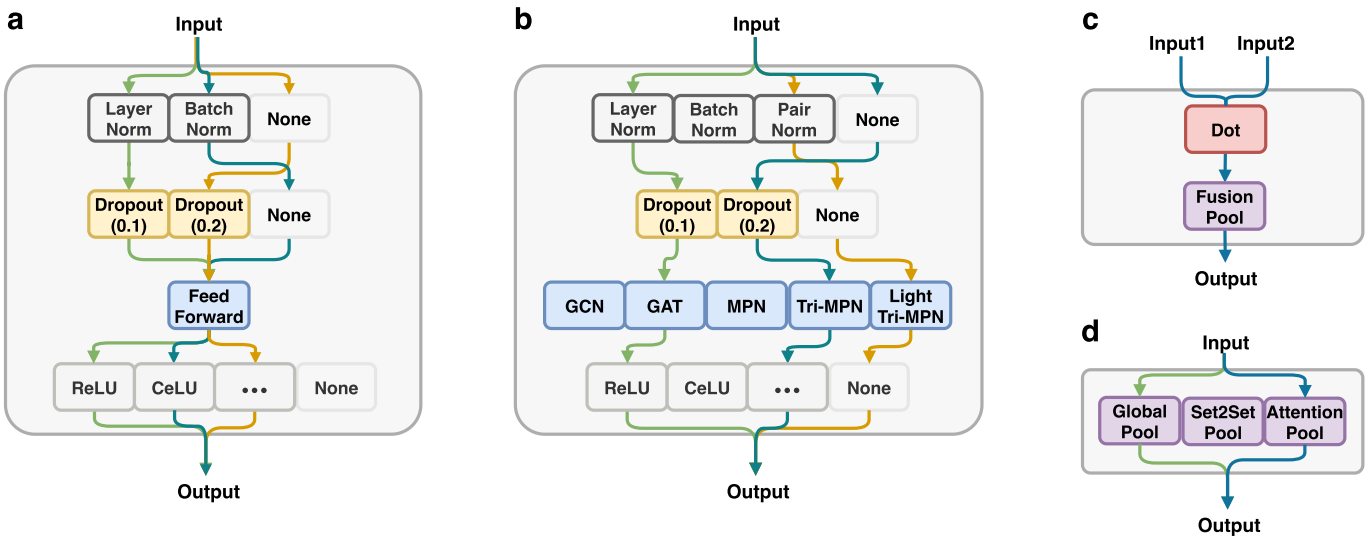
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-022-00501-8.

Correspondence and requests for materials should be addressed to Xiaojun Yao.

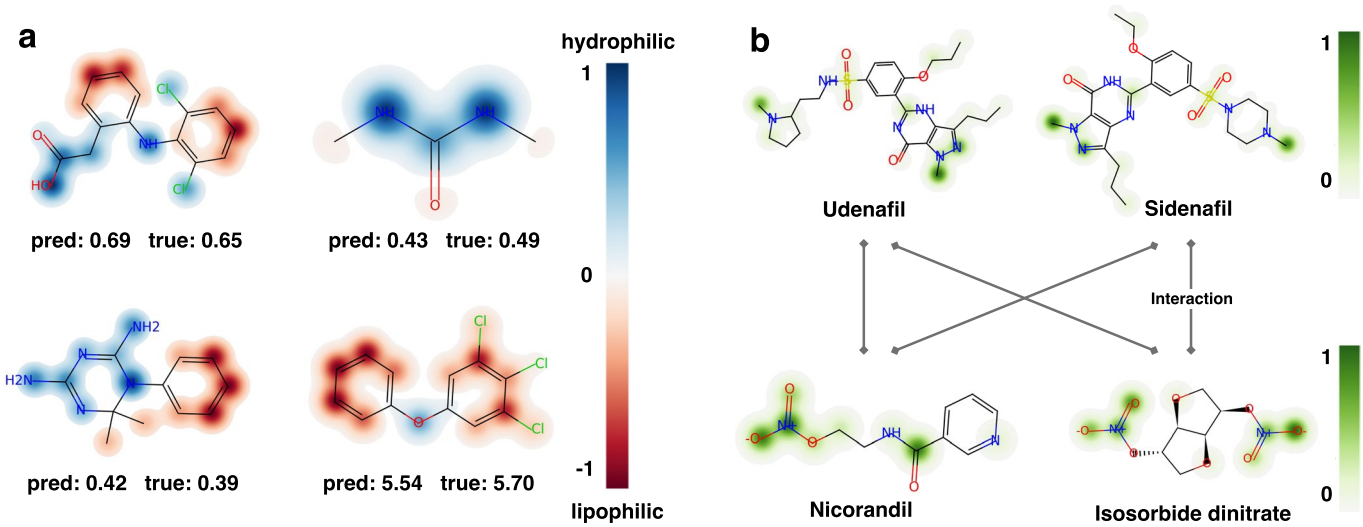
Peer review information *Nature Machine Intelligence* thanks William McCorkindale and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Extended Data Fig. 1 | Design space for blocks of the architectures. a, Feed-forward Block. It takes a tensor as input and outputs a tensor. Abbreviations and their full name correspond as follows: Norm(Normalization), ReLU(Rectified linear units), CeLU(Continuously differentiable exponential linear units). **b**, Message Passing Block. It takes a graph as input and outputs a graph. Abbreviations and their full name correspond as follows: GCN(Graph convolutional networks), GAT(Graph attention networks), MPN(Message-passing neural networks), Tri-MPN(Triplet message-passing neural networks), Light Tri-MPN(Light triplet message-passing neural networks). **c**, Fusion Block. It takes a graph as input and outputs a tensor. Dot means the dot multiplication operation. **d**, Global Pooling Block. It takes a graph as input and outputs a tensor.



Extended Data Fig. 2 | Cases of node-level interpretation. a, Case studies of solubility prediction. 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. **b**, Case studies of drug-drug interactions. 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.