Supplementary information

1 Supplementary notes

1.1 Datasets

Pre-training dataset The pre-training dataset contains about 11 million molecules, which are collected from ZINC and ChemBL database. ZINC is a free database of commercially-available compounds, containing over 750 million purchasable compounds. ChEMBL is a manually curated database of bioactive molecules with drug-like properties. Current Release—ChEMBL27 contains about 2 million compounds. Specifically, we filtered 9 million in-stock molecules with all range of molecular weight from ZINC and collected all the molecules from ChEMBL27. We removed the duplicate molecules to generate the final pre-training dataset.

Downstream datasets In this paper, we have evaluated our method on multiple tasks in chemistry or bioinformatics, which can be divided into three groups: molecular properties prediction, drug-drug interaction (DDI), and drug-target interaction (DTI). The molecular properties datasets, including MUV, HIV, BACE, BBBP, Tox21, ToxCast, SIDER, ClinTox, are derived from MoleculeNet. TWOSIDES and BIOSNAP datasets are used for DDI prediction. For CPI, we employed Human and C.elegan datasets. In the following, we would describe the dataset in detail.

- ESOL is a small dataset consisting of water solubility data for 1128 compounds.
- The Free Solvation Database (FreeSolv) provides experimental and calculated hydration free energy of small molecules in water. 16 A subset of the compounds in the dataset are also used in the SAMPL blind prediction challenge. 15 The calculated values are derived from alchemical free energy calculations using molecular dynamics simulations.
- Lipophilicity curated from ChEMBL database, provides experimental results of octanol/water distribution coefficient (logD at pH 7.4) of 4200 compounds.
- BACE is a database that consists of binding results for a set of inhibitors of human β-secretase 1.
- The Blood-brain barrier penetration (BBBP) dataset comes from a recent study on the modeling and prediction of the barrier permeability. As a membrane separating circulating blood and brain extracellular fluid, the blood-brain barrier blocks most drugs, hormones and neurotransmitters. Thus penetration of the barrier forms a long-standing issue in development of drugs targeting central nervous system. This dataset includes binary labels for over 2000 compounds on their permeability properties.
- Tox21 contains qualitative toxicity measurements for 8014 compounds on 12 different targets, including stress response pathways and nuclear receptors.

- ToxCast is another toxicity database providing toxicology data for compounds based on virtual screening. The processed collection in MoleculeNet contains qualitative results of 617 experiments on 8615 compounds.
- SIDER is a database of marketed drugs and adverse drug reactions (ADR), which grouped into 27 system organ classes.
- ClinTox dataset contains qualitative data of drugs approved by the FDA and those that have failed clinical trials for toxicity reasons.
- Pseudomonas Aeruginosa dataset consists of 2335 molecules. The test set contains 238 molecules, while the rest – 2097 molecules paired with their activity – are used for training. Molecules that inhibited growth >80% were labelled as active. The training set includes 48 active compounds.
- TWOSIDES contains side effects caused by the combination of drugs, we used a recent benchmark dataset [1] derived from TWOSIDES, which contains 548 drugs and 48,584 pairwise drug-drug interactions.
- BIOSNAP [2] that consists of 1,322 approved drugs with 41,520 labelled DDIs, obtained through drug labels and scientific publications. We adopt the dataset from Huang et al. [3]
- Human and C.elegan, created by Liu et al. [4], include highly credible negative samples of compound–protein pairs by using a systematic screening framework. Positive samples of the datasets were retrieved from DrugBank 4.1 and Matador. We used a balanced dataset with a ratio of 1:1 of positive and negative samples following Tsubaki et al. [5].

All molecules of datasets above are preprocessed into hydrogen-depleted molecular graphs with nodes features, edge features, and adjacency matrix with RDKIT [6]. The detailed information about nodes features and edge features can be referred to S1.

1.2 Training details

For both pre-training and fine-tuning, models were trained via the standard batch gradient descent method with the error back-propagation algorithm [7]. Specifically, we used the optimization algorithm Adam to update the parameters. The Xavier initialization algorithm [8] was exploited to initialize their parameters. Two regularization techniques including dropout [9] and the early stopping criterion [10] were employed to eliminate the potential over fitting problem. We employ poly-warmup strategy to prevent training instability. Our MPG framework were implemented by the PyTorch library 1.0.1 [11] and Pytorch Geometric 1.5.0. NVIDIA V100 GPUs were used in the training and testing processes. We use 16 Nvidia V100 GPUs to pre-train MolGNet. Pre-training MolGNet took 2 days.

1.3 Hyperparameters

The hyperparameters used for pre-training MolGNet are listed in Table S2. For fine-tuning on downstream datasets, a grid search procedure was applied to obtain the optimal hyperparameters. The hyperparameters tuning process involved the learning rate λ, learning rate decay α, the dropout rate d, the graph pooling method P. We applied a coarse grid search approach over $\lambda \in \{0.001, 0.0001, 0.00015\}$ with $\alpha \in \{0.99, 0.995, 0.9995\}$, $d \in \{0, 0.2, 0.5\}$, and graph pooling

methods including mean pooling, set2set pooling, attention pooling and collection node on the validation dataset to select the best settings. The collection node pooling methods means that the embedding of collection node is regarded as the global graph representation. Finally, we reported the test performance based on the selected hyperparameters (shown in Table S4).

1.4 SMILES-BERT implementation

We pre-trained a Transformer model on SMILES strings of 11 million molecules mentioned above using masking strategy. Specifically, we regard a SMILES as a sentence, and randomly mask 15% letters of a SMILES to let the model predict the masked letters. The hyperparameters used for pre-training SMILES-BERT are listed in Table S3

1.5 DTI framework using MolGNet as encoder

In this study, we adapt Tsubaki et al.'s DTI framework to accomplish the DTI prediction task by replacing their GNN model with our MolGNet. As shown in Figure S3, the DTI framework consist of a molecular encoder—GNN model (MolGNet) and a target sequence encoder–CNN model. It uses attention mechanism to derive the protein sequence representation, and then concatenates the protein representation with the molecular representation to give the final prediction. Specifically, given a molecular vector y_m and a set of hidden vectors of sub-sequences in a protein $C = (c_1, c_2, \cdots, c_n)$, they weight for c_i considering y_m. The weight value α_i is calculated by:

$$h_m = f(\boldsymbol{W} h_m + b) \tag{1}$$
$$h_i = f(\boldsymbol{W} c_i + b) \tag{2}$$
$$\alpha_i = \sigma(h_m^T h_i) \tag{3}$$
$$\tag{4}$$

where $\boldsymbol{W}$ is the learned weight matrix and b is the bias vector. Using the attention weights, the protein representation is obtained by the weighted sum of h_i as follows:

$$y_p = \sum_i^n (\alpha_i h_i) \tag{5}$$

More details about the Tsubaki et al.'s DTI framework are referred to their paper [5] .

1.6 Analysis about what the network pre-trained by PHD can learn

Similar to the experimental setting in Figure 4 (c) and (d) of the main manuscript, we selected ten common scaffolds and randomly sampled 1000 molecules for each scaffolds to prepare the datasets. Based on the molecular representation learned from MolGNet pre-trained by PHD, we used a linear classification to identify which scaffold the molecule belongs to. We conducted ten-fold cross validation to evaluate the performance. We also performed the same analysis on the MolGNet model that was not pre-trained for comparison. The results shows that the molecular representation extracted by the pre-trained model with PHD achieved 96.1% in terms of 10-fold classification accuracy, which significantly outperform the representation extracted from no-pretrained model (91.6%).

2 Supplementary figures and tables

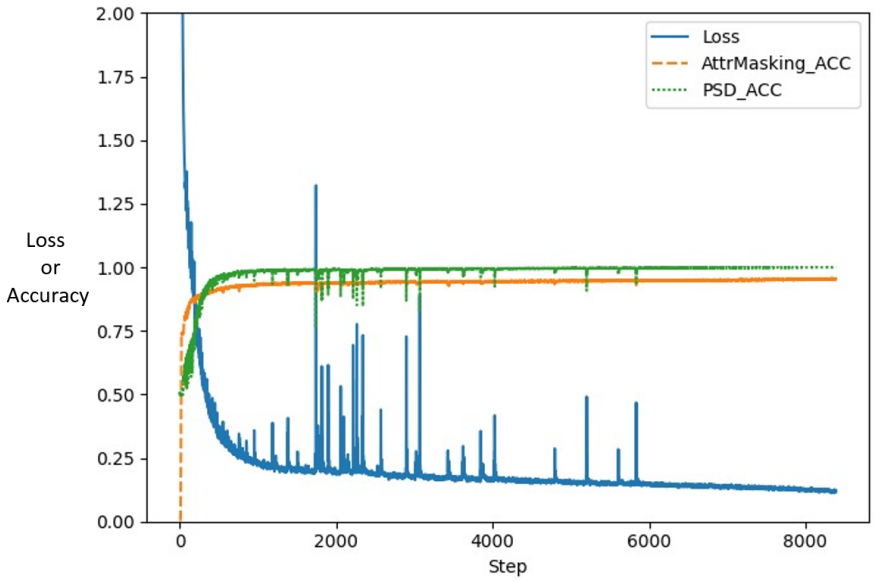

Figure S1: Learning curves of pre-training MolGNet on 11 million of molecules.

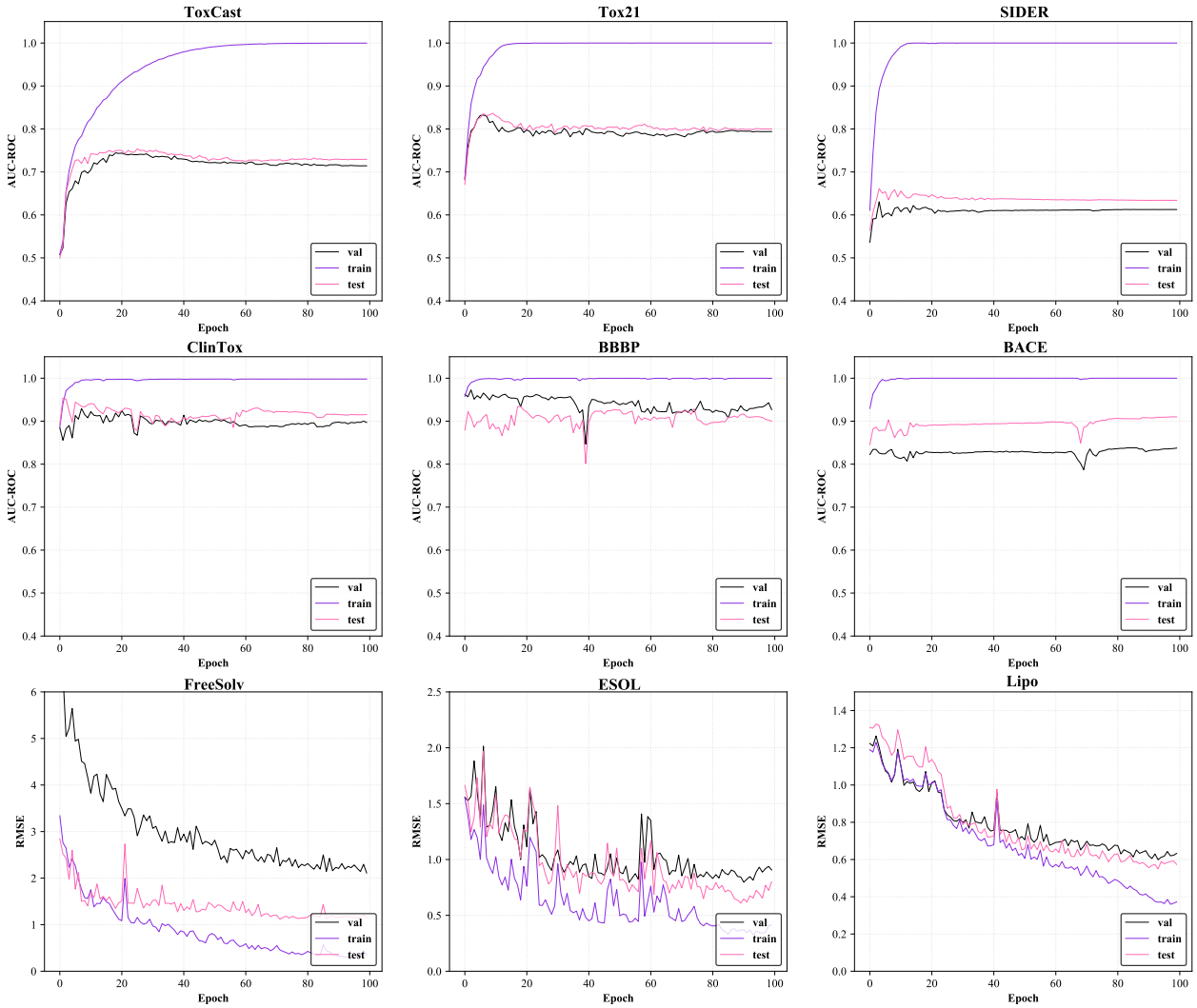

Figure S2: Learning curves of pre-trained MolGNet on molecular properties predictions.

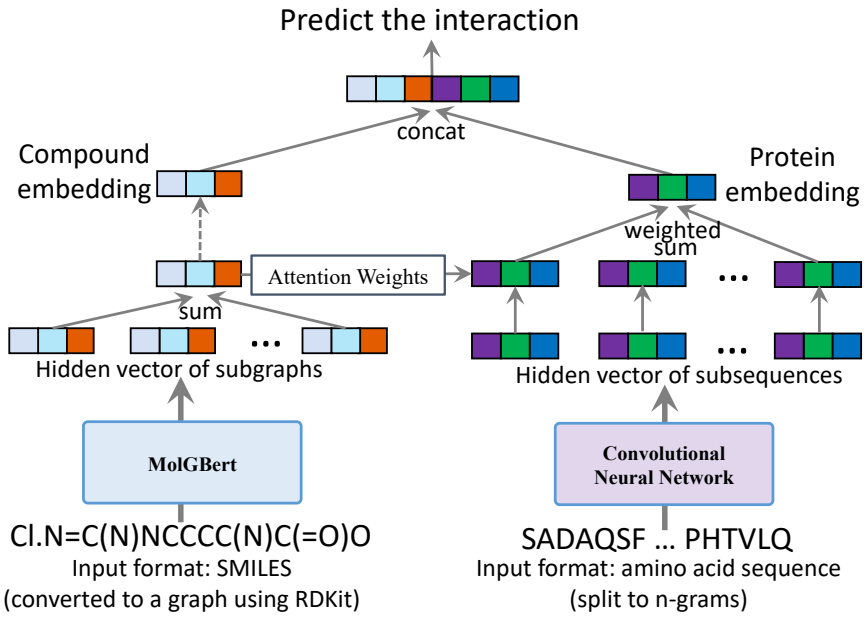

Figure S3: DTI framework to accomplish the DTI prediction task by replacing Tsubaki et al.'s GNN model with our MolGNet.

Table S1: The features used in molecular graph. These features are obtained by RDKIT.

Type	Name	Description
Node feature	Atom type	Atomic number (0-122)
	Formal charge	[unk,-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
	Chirality type	[unk, unspecifie, tetrahedral-CW, tetrahedralL-CCW, other]
	Hybridization	[unk, sp, sp2, sp3, sp3d, sp3d2, unspecified]
	NumH	Number of connected hydrogens[unk,0, 1, 2, 3, 4, 5, 6, 7, 8]
	Implicit valence	[unk, 0, 1, 2, 3, 4, 5, 6]
	Degree	Number of covalent bonds [unk, 0, 1, 2, 3, 4, 5,6,7,8,9,10]
	Aromatic	Whether the atom is part of an aromatic system [0,1]
Edge feature	Bond direction	[None,endupright, enddownright]
	Bond type	[Single, double, triple, aromatic]
	Conjugation	Whether the bond is conjugated [0,1]
	Ring	Whether the bond is in Ring [0,1]
	Stereo	[StereoNone, StereoAny, StereoZ, StereoE]

Table S2: The hyperparams of MolGNet for pre-training

Hidden size	heads	message passing steps	layers	batch size	training steps	lr	warmup proportion
768	12	3	5	32	8393	0.006	0.28

Table S3: The hyperparams of SMILES-BERT for pre-training

Hidden size	heads	layers	batch size	training steps	lr	warmup proportion
768	12	12	32	7038	0.006	0.28

Table S4: The hyperparams of MolGNet on all downstream datasets

	BBBP	**Tox21**	**ToxCast**	**SIDER**	**ClinTox**	**BACE**	**ESOL**	**FreeSol**	**LIPO**	**BIOSNAP**	**TWOSIDES**	**C.elegans**	**Human**
Epoch	100	100	100	100	100	100	100	100	100	100	100	100	100
Batch Size	16	16	16	16	16	16	32	32	32	16	16	1	1
Lr	0.00015	0.0001	0.0001	0.00015	0.0001	0.0001	0.001	0.0001	0.0001	0.0001	0.0001	0.001	0.001
Lr Decay	0.995	0.98	0.9995	0.995	0.99	0.99	0.995	0.99	0.99	0.995	0.995	0.98	0.98
Dropout	0.2	0.2	0.2	0	0.2	0	0.5	0	0	0.5	0.5	0.5	0.5

References

[1] Zheng, Y. *et al.* Ddi-pulearn: a positive-unlabeled learning method for large-scale prediction of drug-drug interactions. *BMC bioinformatics* **20**, 1–12 (2019).

[2] Zitnik, M., Rok Sosic, S. & Leskovec, J. Biosnap datasets: Stanford biomedical network dataset collection. *Note: http://snap. stanford. edu/biodata Cited by* **5** (2018).

[3] Huang, K., Xiao, C., Hoang, T., Glass, L. & Sun, J. Caster: Predicting drug interactions with chemical substructure representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 702–709 (2020).

[4] Liu, H., Sun, J., Guan, J., Zheng, J. & Zhou, S. Improving compound-protein interaction prediction by building up highly credible negative samples. *Bioinformatics* **31**, i221–i229 (2015).

[5] Tsubaki, M., Tomii, K. & Sese, J. Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics* **35**, 309–318 (2019).

[6] Landrum, G. Rdkit: Open-source cheminformatics (2006). URL http://www.rdkit.org.

[7] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *nature* **323**, 533 (1986).

[8] Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. & Titterington, M. (eds.) *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9 of *Proceedings of Machine Learning Research*, 249–256 (PMLR, Chia Laguna Resort, Sardinia, Italy, 2010). URL http://proceedings.mlr.press/v9/glorot10a.html.

[9] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15**, 1929–1958 (2014).

[10] Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, 437–478 (Springer, 2012).

[11] Paszke, A. *et al.* Automatic differentiation in pytorch. *Neural Information Processing Systems* (2017).