Supplementary Information for: Docking-based Virtual Screening with Multi-Task Learning

Zijing Liu[1], Xianbin Ye[2], Xiaomin Fang[1], Fan Wang[1], Hua Wu[3], Haifeng Wang[3]

[1] Baidu Inc., Shenzhen, China
[2] Jinan University, Guangzhou, China
[3] Baidu Inc., Beijing, China

S.1 Details of multi-task learning model based on GNN

We first introduce the MTL model for docking score prediction in detail (Figure S1).

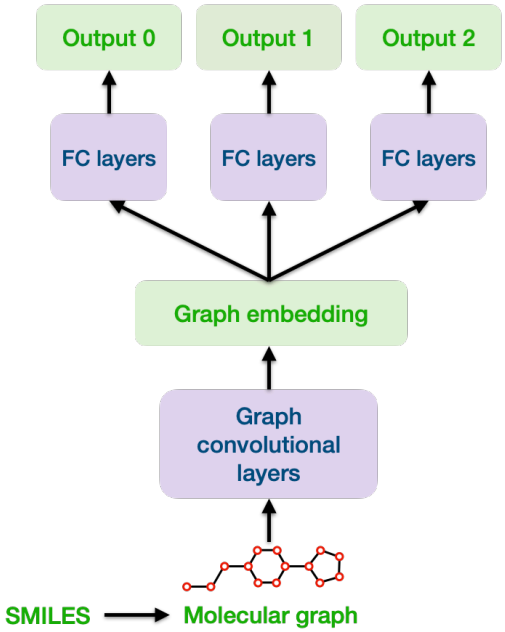

Figure S1: The MTL model for virtual screening with graph neural networks. The weights of the graph convolutional layers are shared across all the tasks. For each task, the output embedding of the graph convolutional layers goes into task-specific fully connected layers to get the final prediction.

S.1.1 Input molecular features

In a chemical library, the compounds are usually represented by SMILES (Simplified Molecular Input Line Entry System)[1]. Recently, modeling the molecules as graphs has shown a great success in many problems related to drug discovery, including molecular property prediction, drug-target interaction prediction, and molecule design[2–5]. Thus we convert the SMILES codes to molecular

graphs with the package RDKit[6], such that a compound is a graph with the nodes being the atoms and the edges being the chemical bonds. RDKit is also used to generate features for the molecular graph, including seven atom features and two bond features. All the features are discrete values and encoded as one-hot vectors with different lengths, as shown in Table S1.

Table S1: Input features for the molecular graph

Atom feature	atomic type	119 bits one hot
	formal charge	16 bits one hot
	degree	11 bits one hot
	chirality tag	4 bits one hot
	number of hydrogens	9 bits one hot
	aromaticity	2 bits one hot
	hybridization	5 bits one hot
Bond feature	bond direction,	7 bits one hot
	bond type	4 bits one hot
	is in ring	2 bits one hot

S.1.2 Parameter sharing module

With the input of molecular graph data, the parameter-sharing part of our MTL model G_ϕ is based on the Graph Isomorphism Network (GIN), which has been shown to achieve the state-of-the-art performance in drug-target affinity prediction[5,7]. As shown in Table S1, the features generated by RDKit are all one-hot vectors, which are denoted as $f_{v,i}$ $(i = 1, \ldots, 7)$ and $f_{e,j}$ $(j = 1, 2, 3)$ for node v and edge e. In order to input the features to GNNs, we first use embedding operations to map the one-hot vectors into d-dimensional real vectors:

$$h_v^0 = \sum_{i=1}^{7} \text{NodeEmbedding}_i(f_{v,i}),$$
$$h_e^k = \sum_{j=1}^{3} \text{EdgeEmbedding}_j^k(f_{e,j}) \quad k = 0, 1, \ldots, K-1,$$

where K is the total number of GNN layers. In the k-th layer, GIN updates the node representations by

$$h_v^k = \sigma\left(g^k\left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} h_u^{k-1} + \sum_{e=(v,u): u \in \mathcal{N}(v) \cup \{v\}} h_e^{k-1}\right)\right),$$

where $\mathcal{N}(v)$ is the set of nodes adjacent to node v, $\sigma(\cdot)$ is the ReLU activation function, and $g^k(\cdot)$ is a two-layer perceptron with $2d$ hidden neurons followed by batch normalization[8]. Since the docking score prediction is a graph-level regression task, the graph representation is obtained by averaging the node embeddings of the last GIN layer:

$$z = \text{MEAN}_v(h_v^K).$$

S.1.3 Task-specific module

The output of the GIN layers gives a vector representation of the compounds, which contains shared knowledge across all the tasks. These vector representations are then fed into the task-specific neural networks to make predictions for each task. Here, we apply a two-layer fully connected (FC) neural network on top of the graph representation z to obtain the final prediction for each task.

S.2 Experiment details

S.2.1 Datasets

The first docking dataset was obtained from the work of Deep docking[9], which contained 12 targets (PDB id: 1ERR, 1T7R, 2ZV2, 4AG8, 4F8H, 4R06, 4YAY, 5EK0, 5L2S, 5MZJ, 6D6T, 6IIU), and each target was docked with 3 million compounds randomly sampled from ZINC15 by the program FRED[10]. The second dataset was obtained from the work of Lyu et al.[11], in which 99 million lead-like compounds were docked against the target AmpC (AmpC β-lactamase) using DOCK3.7.2[12]. These two datasets allow us to test the performance of MTL for virtual screening in a large scale and across different docking softwares. To further test the ability of MTL in drug discovery, we also collect an experimental drug-target affinity dataset (the half-maximal inhibitory concentrations of the ligands on their protein targets, i.e., IC_{50}) with eight targets from the ChEMBL database[13]. ChEMBL uses the pChEMBL value as the measurement of the ligand-target affinity, which is defined as:

$$\text{pChEMBL} = -\log(\text{IC}_{50}). \tag{S1}$$

For example, an IC_{50} measurement of 10 uM would have a pChEMBL value of 5. We use this pChEMBL value as the ground truth in our experiment. When a ligand-target pair has multiple pChEMBL values, we take the average of all the pChEMBL values. The details of the drug-target affinity dataset are summarized in TABLE S2.

Table S2: Details of the drug-target affinity dataset collected from ChEMBL

ChEMBL ID	Target name	Data points
203	Epidermal growth factor receptor erbB1	6414
279	Fetal liver kinase 1	7777
267	Tyrosine-protein kinase SRC	2775
325	Histone deacetylase 1	4473
333	Matrix metalloproteinase-2	2673
2842	Serine/threonine-protein kinase mTOR	3662
2971	Tyrosine-protein kinase JAK2	4855
4005	PI3-kinase p110-alpha subunit	4459

S.2.2 Evaluation metrics

The performance of the machine learning approaches for docking score prediction is evaluated by the virtual hit ratio. We define the top-k compounds as the virtual hits. For the first dataset of 3 million compounds, k is set to be 3,000 or 30,000. For the larger dataset of 99 million compounds,

k is 10,000 or 100,000. The predicted virtual hits are taken as the top 2%, 3% or 5% compounds of the predictions. The hit ratio is calculated as the recall:

$$\text{recall} = \frac{TP}{TP + FN},$$

where TP (true positives) are the correctly predicted virtual hits, and FN (false negatives) are the virtual hits discarded by the prediction incorrectly.

The performance of the drug-target affinity prediction is measured by the mean-square error (MSE) and concordance index (CI). The CI is defined as

$$\text{CI} = \frac{\sum_{k,l} \mathbf{1}_{\tilde{y}_k > \tilde{y}_l} \cdot \mathbf{1}_{y_k > y_l}}{\sum_{k,l} \mathbf{1}_{y_k > y_l}},$$

where $\tilde{y}_k$ and y_k are the predicted and true value of the k-th sample respectively, and $\mathbf{1}_{y_k > y_l} = 1$ if $y_k > y_l$, else 0.

S.2.3 Model training details

The embedding dimension for the features d is 256. The number of GIN layers is eight. The number of hidden neurons in the task-specific FC layers is also 256. The model is trained with the ADAM optimizer with a learning rate 0.001 and batch size 128[14]. During training, we apply dropout after the ReLU for all GNN layers and the task-specific FC layers[15]. The dropout rate is set to be 0.2. 20% of the training data are used for validation to avoid over-fitting. The model is trained for at least 100 epochs until the validation loss is larger than the training loss for more than 50 consecutive epochs. The epoch with the smallest validation loss is chosen as the final model. For active learning, the final prediction is the average of the five models in the ensemble.

S.3 Additional results

S.3.1 MTL for virtual screening

As a summary of the prediction performance, TABLE S3 gives the Pearson correlation between the real and predicted docking scores for the 12 targets in the first dataset.

S.3.2 MTL trained model contains common reusable knowledge

We have shown that MTL helps learn a good representation of the chemical compounds, which indicates that the model is able to capture common knowledge across all the tasks. We examine if the knowledge learned by MTL can be reused for a new task by transferring the MTL trained model to a new target and checking if it can improve the prediction performance. We test this idea using the first docking dataset with 12 targets. Similarly, one target is taken as the new target. An MTL model is first trained with the data of the other 11 targets. The weights of the MTL trained GIN layers are used as the initialization to train a single-task model of the new target. We first train the FC layers for 20 epochs and then train the whole neural network. From Figure S2, it is clear to see that transfer learning based on the MTL model achieves better recall values than single-task machine learning and active learning. Figure S2 also reports that when the data size of the other tasks increases from 100k to 2m, the performances of MTL and MTL initialized transfer learning become better.

Table S3: Pearson correlation of single-task machine learning (ML), active learning (AL) and MTL with different new target sizes. MTL is jointly trained with 2 million data from other targets.

	Target size=1k			Target size=2k			Target size=5k			Target size=10k		
	ML	AL	MTL	ML	AL	MTL	ML	AL	MTL	ML	AL	MTL
1ERR	0.562	0.566	**0.630**	0.605	0.610	**0.662**	0.617	0.635	**0.710**	0.627	0.645	**0.732**
1T7R	0.800	0.790	**0.844**	0.823	0.826	**0.859**	0.851	0.852	**0.882**	0.859	0.854	**0.879**
2ZV2	0.531	0.545	**0.618**	0.581	0.582	**0.646**	0.588	0.606	**0.680**	0.608	0.610	**0.696**
4AG8	0.671	0.685	**0.751**	0.718	0.737	**0.771**	0.758	0.769	**0.790**	0.770	0.777	**0.812**
4F8H	0.566	0.596	**0.658**	0.636	0.628	**0.689**	0.650	0.668	**0.721**	0.667	0.694	**0.747**
4R06	0.493	0.543	**0.597**	0.548	0.585	**0.639**	0.610	0.602	**0.680**	0.603	0.596	**0.697**
4YAY	0.528	0.553	**0.550**	0.568	0.580	**0.589**	0.585	0.590	**0.639**	0.595	0.601	**0.669**
5EK0	0.492	0.496	**0.564**	0.525	0.532	**0.587**	0.564	0.560	**0.650**	0.570	0.580	**0.680**
5L2S	0.474	0.496	**0.579**	0.519	0.516	**0.620**	0.562	0.563	**0.655**	0.586	0.527	**0.679**
5MZJ	0.663	0.677	**0.743**	0.704	0.704	**0.762**	0.725	0.744	**0.778**	0.733	0.744	**0.798**
6D6T	0.498	0.534	**0.616**	0.587	0.590	**0.655**	0.611	0.620	**0.694**	0.610	0.623	**0.715**
6IIU	0.585	0.609	**0.670**	0.639	0.648	**0.708**	0.669	0.679	**0.747**	0.694	0.667	**0.764**

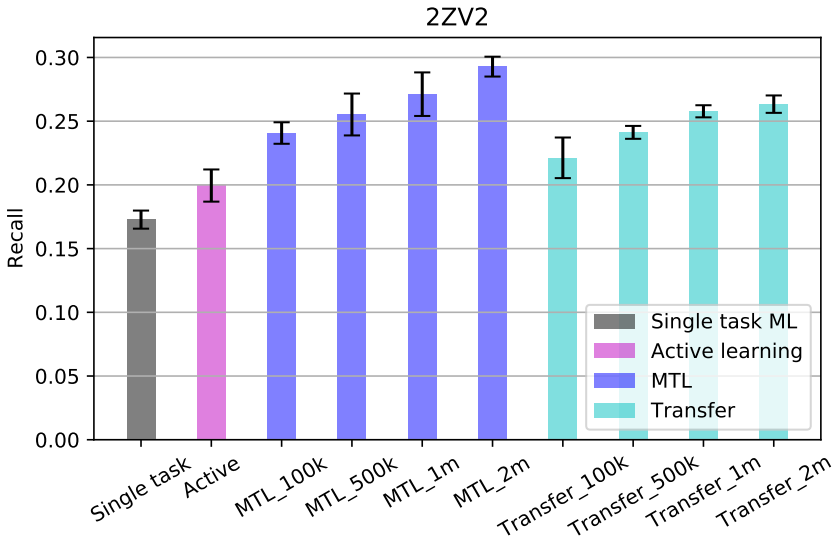

Figure S2: Bar plots of the recall of top 3k virtual hit for the target 2ZVS at top 2% predicted values of different ML approaches. The training size for the new target is 10k. MTL and transfer learning include the results of 4 different training sizes for the other targets. The error bars are obtained from three independent runs.

References

[S1] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[S2] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference*

on Machine Learning, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1263–1272. [Online]. Available: http://proceedings.mlr.press/v70/gilmer17a.html

[S3] K. Liu, X. Sun, L. Jia, J. Ma, H. Xing, J. Wu, H. Gao, Y. Sun, F. Boulnois, and J. Fan, "Chemi-Net: a molecular graph convolutional network for accurate drug property prediction," *International journal of molecular sciences*, vol. 20, no. 14, p. 3389, 2019.

[S4] M. Sun, S. Zhao, C. Gilvary, O. Elemento, J. Zhou, and F. Wang, "Graph convolutional networks for computational drug development and discovery," *Briefings in bioinformatics*, vol. 21, no. 3, pp. 919–935, 2020.

[S5] T. Nguyen, H. Le, T. P. Quinn, T. Nguyen, T. D. Le, and S. Venkatesh, "GraphDTA: Predicting drug–target binding affinity with graph neural networks," *Bioinformatics*, vol. 37, no. 8, pp. 1140–1147, 2021.

[S6] G. Landrum, "RDKit: Open-source cheminformatics," 2006.

[S7] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=ryGs6iA5Km

[S8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[S9] F. Gentile, V. Agrawal, M. Hsing, A.-T. Ton, F. Ban, U. Norinder, M. E. Gleave, and A. Cherkasov, "Deep docking: a deep learning platform for augmentation of structure based drug discovery," *ACS central science*, vol. 6, no. 6, pp. 939–949, 2020.

[S10] M. McGann, "FRED and HYBRID docking performance on standardized datasets," *Journal of computer-aided molecular design*, vol. 26, no. 8, pp. 897–906, 2012.

[S11] J. Lyu, S. Wang, T. E. Balius, I. Singh, A. Levit, Y. S. Moroz, M. J. O'Meara, T. Che, E. Algaa, K. Tolmachova *et al.*, "Ultra-large library docking for discovering new chemotypes," *Nature*, vol. 566, no. 7743, pp. 224–229, 2019.

[S12] R. G. Coleman, M. Carchia, T. Sterling, J. J. Irwin, and B. K. Shoichet, "Ligand pose and orientational sampling in molecular docking," *PloS one*, vol. 8, no. 10, p. e75992, 2013.

[S13] D. Mendez, A. Gaulton, A. P. Bento, J. Chambers, M. De Veij, E. Félix, M. P. Magariños, J. F. Mosquera, P. Mutowo, M. Nowotka *et al.*, "ChEMBL: towards direct deposition of bioassay data," *Nucleic acids research*, vol. 47, no. D1, pp. D930–D940, 2019.

[S14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[S15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.