MolE: a molecular foundation model for drug discovery

Anonymous Author(s)
Affiliation
Address
email

Abstract

Models that accurately predict properties based on chemical structure are valuable tools in drug discovery. However, for many properties, public and private training sets are typically small, and it is difficult for the models to generalize well outside of the training data. Recently, large language models have addressed this problem by using self-supervised pretraining on large unlabeled datasets, followed by fine-tuning on smaller, labeled datasets (1; 2; 3). In this paper, we report MolE, a molecular foundation model that adapts the DeBERTa (4) architecture to be used on molecular graphs together with a two-step pretraining strategy. The first step of pretraining is a self-supervised approach focused on learning chemical structures, and the second step is a massive multi-task approach to learn biological information. We show that fine-tuning pretrained MolE achieves state-of-the-art results on 9 of the 22 ADMET tasks included in the Therapeutic Data Commons (5).

1 Introduction

Machine learning has been successfully applied to chemical sciences for many decades (6). In particular, molecular property prediction has been critical in successfully advancing material and drug discovery projects (7). Nonetheless, a major challenge in this area is to represent a molecule in a way that is compatible with machine learning algorithms with minimum information loss. Initially, molecules were represented in terms of their physicochemical properties (e.g., partition coefficient) or information that can be obtained from the molecular formula such as molecular weight or number of heteroatoms (8). While this approach was successful for the first quantitative structure-activity relationship (QSAR) studies (9), it uses global properties of the molecule that do not provide information about the topology of the molecular graph. With time, molecules were described in more sophisticated ways using molecular fingerprints such as MACCS keys (10) and Extended Connectivity Fingerprints (ECFPs) (11). Currently, more recent deep learning architectures can directly use molecules as strings or graphs.

In this paper, we show how architectures developed for natural language processing can be used with molecular graph representations for property prediction. In particular, we present MolE, a model that learns molecular embeddings, at the atomic environment level, directly from a molecular graph. MolE is based on DeBERTa (4), a transformer architecture that uses disentangled attention in order to account for relative token positions. We also describe a pretraining strategy that combines self-supervised pretraining on chemical structures with supervised multi-task pretraining of pharmacological and other biological properties, yielding a molecular foundation model capable of being fine-tuned on small datasets to achieve top performance on typical benchmark tasks. This work is of relevance for chemical sciences where large amounts of molecular structures are available but the size of labeled datasets is usually very small.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

2 Experiments

2.1 Datasets

The self-supervised pretraining was done using the GuacaMol (12) dataset, which was previously used to train MolBERT (13) and consists of ∼1.2 million molecules for training and a validation set of ∼79K molecules, mostly extracted from ChEMBL. It is worth mentioning that only molecules with no more than 100 heavy atoms were used, and we removed from the training set all molecules included in TDC test sets to avoid information leakage. All remaining SMILES were transformed into molecular graphs using RDKit from which distance matrices and atom environments were calculated. Atom environment identifiers were aggregated into two vocabularies, one used for input and one for labels. The input vocabulary consists of 207 tokens corresponding to all atom environments of radius 0 present in the 1.2 million molecules in GuacaMol, plus the ∼880 million molecules in ZINC20 (14). Similarly the vocabulary used for labels contains ∼141K atom environments or ∼114K functional atom environments (Table 3). These were selected taking the 90K most frequent atom environments or functional environments from GuacaMol training set plus the 90K most frequent form ZINC20 and removing those that appear in less than 3 molecules.

The supervised pretraining was done using ∼456K molecules with activity data on 1,310 prediction tasks from ChEMBL, which was curated following the protocol proposed by Mayr et al. (15) and used for pretraining by Hu et al. (16). Here again we removed ∼9,900 molecules that were present in the test sets of the benchmark datasets.

2.2 Training

MolE uses the DeBERTa base configuration (12 transformer layers with 12 attention heads each) with a prediction head connected to the output of a class token composed of a two-layer MLP with a GELU and dropout in between (Figure 1a). Models were pretrained for 60,000 steps using a batch size of 512 molecules in both the supervised and self-supervised cases. In case of the self-supervised models, the learning rate was increased to 2×10^{-4} during the first 10,000 warm up steps, followed by a linear decaying learning rate schedule. For the supervised pretraining, we used a learning rate of 5×10^{-6} with the same schedule as the self-supervised training. Gradient norms were clipped at 1.0 and no weight decay was used.

For fine-tuning, only the weights of the prediction head were randomly initialized. Models were trained for 100 epochs using a batch size of 32 molecules. We ran hyperparameter optimization to find the best learning rate (1e-5, 8e-6, 5e-6, 3e-6, 1e-6, 5e-7) and dropout rate (0, 0.1, 0.15 in the prediction head) with a 5-fold cross validation using the folds provided in the TDC benchmark datasets selected via scaffold splitting. During training, the learning rate was linearly increased during the first 10% of the training steps, and then was kept constant.

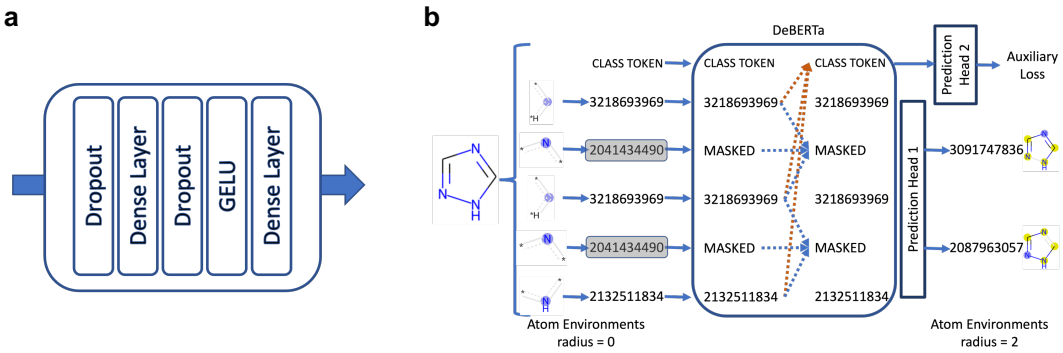

Figure 1: a) Representation of the prediction head used in this study. b) Representation of the self-supervised approach trained with an auxiliary loss. We evaluated two different auxiliary losses: logP prediction and fingerprint prediction.

2.3 Benchmark

MolE models were evaluated using the ADMET benchmark group from the Therapeutic Data Commons (TDC) (5). This benchmark provides datasets that have been previously standardized and divided into training and test sets (80%/20% using scaffold splitting) to fairly evaluate molecular property prediction models. It is composed of 22 different classification and regression tasks for properties relevant to drug discovery. More information about each of these tasks is listed in Table 4.

TDC maintains a leaderboard of the performance of different models on these tasks. These models provide a baseline for performance comparison purposes since they use different architectures and encoding strategies, e.g., pre-calculated descriptors such as Morgan or RDKit 2D fingerprints (17), CNNs trained using SMILES, and different flavors of graph-based approaches such as NeuralFP (18), GCNs (19), AttentiveFP (20), and others. It also includes models pretrained with different strategies e.g., AttrMasking and ContextPred (16).

3 Results

3.1 Effect of pretraining strategies

Table 1 and Appendix D shows the benchmark results of all models pretrained with different strategies compared to the best model reported on the TDC leaderboard. We also present the results of MolE without any pretraining, which was used as a baseline. Interestingly, the model without pretraining already exhibits better results on 4 of the 22 tasks compared to models reported by Huang et al. (5). This suggests that just the use of Transformers with disentangled attention already positively impacts prediction power despite using small datasets. Nonetheless, these models did not perform better than the best models reported in the leaderboard (as of September 2022) in any of the predictive tasks. The same table also shows results after self-supervised pretraining using two approaches, namely, MolE (AtomEnvs) and MolE (FunctionalEnvs). In the former, the pretraining task is to predict the atom environment of radius 2 taking as input an atom environment of radius 0. The latter is similar, but now the pretraining task is to predict the functional atom environment of radius 2. High accuracy (> 98%) on a validation set was obtained with either pretraining strategy. However, MolE (AtomEnvs) performed slightly better on the benchmark tasks, where it outperformed previous models (5; 21) in the leaderboard on 6 tasks whereas MolE (FunctionalEnvs) did so on 3. This is not an unexpected result since we expect that pretraining using atom environments provides a better representation of the molecule compared to the less-specific description provided by functional environments.

We noticed that supervised pretraining helps to improve prediction power even when used without any self-supervised pretraining. Nonetheless the improvement is marginal compared to not using any pretraining at all. We hypothesize that it is hard to learn a transferable representation of molecules at the same time as the prediction task, especially with a small number of examples, and for that reason the gain of just using supervised pretraining is not remarkable. In a similar way, using supervised learning after the self-supervised approach helped MolE (AtomEnvs) outperform previously reported models on 9 of the 22 tasks. A similar improvement in performance was observed after adding multi-task supervision to MolE (FunctionalEnvs). However, we note that using the supervised approach had a negative effect on tasks related to toxicity where performance was decreased.

Table 1: Summary of number of tasks in which our models perform better than the models reported in the TDC leaderboard (September 2022) for 22 ADMET predictive tasks. Complete results can be found in Appendix D.

pretraining	Self-Supervised label	No Auxiliary Loss	logP as Auxiliary Loss	FP as Auxiliary Loss
None	—-	0	—-	—-
Only Supervised	—-	3	—-	—-
Only Self-Supervised	AtomEnvs	6	4	3
	FunctionalEnvs	3	5	4
Self-Supervised + Supervised	AtomEnvs	9	6	7
	FunctionalEnvs	7	7	7

3.2 Using auxiliary losses in pretraining

We also explored the idea of using auxiliary losses during self-supervised pretraining. For this we explored two different approaches: one is to learn the partition coefficient (logP) and the other is to learn a binary fingerprint (FP) of the molecule. In the first task, we take the standard approach of adding a class token to the input and predict logP from its embeddings (Figure 1b). Here, logP values are calculated for every molecule using RDKit (17). Table 1 shows the results of this task for both MolE (AtomlEnvs) and MolE (FunctionalEnvs). In general, only marginal improvements were observed for the self-supervised version of MolE (FunctionalEnvs), which is best on 5 of 22 tasks. Interestingly, using this auxiliary loss decreased the performance of MolE (AtomEnvs) with both pretrainings.

For the second approach, we framed the fingerprint learning as a multitask binary classification problem where the task is to identify the presence or absence of each atom environment in the vocabulary. Results are again shown in Table 1, where it can be observed that only fine-tuning was improved performance over the self-supervised-only version of MolE (FunctionalEnvs), outperforming the leaderboard models in 4 of 22 tasks (compared with 3 without using an auxiliary loss). Our hypothesis is that MolE (FunctionalEnvs) takes greater advantage of this approach to overcome the imprecise nature of functional environments. Interestingly, using an auxiliary loss during self-supervision did not improve the models when supervised pretraining is also used.

Table 2: Comparison between the best models reported in the TDC leaderboard (as of September 2022) and the model reported in this paper.

	Dataset	Metric	Best in TDC Leaderboard (September 2022)		MolE (AtomEnvs) + Supervised	
			Current Best Model	Result	Result	Rank
Absorption	Caco2	MAE	XGBoost	0.291 ± 0.015	0.310 ± 0.010	2
	HIA	AUROC	XGBoost	0.988 ± 0.002	0.963 ± 0.019	6
	Pgp	AUROC	SimGCN	0.929 ± 0.010	0.915 ± 0.005	5
	Bioavailability	AUROC	SimGCN	0.748 ± 0.033	0.654 ± 0.028	5
	Lipophilicity	MAE	ContextPred	0.535 ± 0.012	**0.469 ± 0.009**	1
	Solubility	MAE	XGBoost	0.734 ± 0.006	0.792 ± 0.005	3
Distribution	BBB	AUROC	XGBoost	0.907 ± 0.002	0.903 ± 0.005	2
	PPBR	MAE	XGBoost	8.252 ± 0.190	**8.073 ± 0.335**	1
	VDss	Spearman	XGBoost	0.627 ± 0.009	**0.654 ± 0.031**	1
Metabolism	CYP2D6 inhibition	AUPRC	XGBoost	0.717 ± 0.001	0.682 ± 0.008	2
	CYP3A4 inhibition	AUPRC	XGBoost	0.872 ± 0.005	0.867 ± 0.003	2
	CYP2C9 inhibition	AUPRC	XGBoost	0.769 ± 0.000	**0.801 ± 0.003**	1
	CYP2D6 substrate	AUPRC	RDKit2D + MLP (DeepPurpose)	0.677 ± 0.047	**0.699 ± 0.018**	1
	CYP3A4 substrate	AUROC	XGBoost	0.677 ± 0.007	0.670 ± 0.018	2
	CYP2C9 substrate	AUPRC	SimGCN	0.433 ± 0.017	**0.446 ± 0.062**	1
Excretion	Half life	Spearman	SimGCN	0.392 ± 0.065	**0.549 ± 0.024**	1
	Clearance microsome	Spearman	SimGCN	0.597 ± 0.025	**0.607 ± 0.027**	1
	Clearance hepatocyte	Spearman	ContextPred	0.439 ± 0.026	0.381 ± 0.038	6
Toxicity	hERG	AUROC	SimGCN	0.874 ± 0.014	0.823 ± 0.009	4
	Ames	AUROC	XGBoost	0.859 ± 0.000	0.813 ± 0.005	8
	DILI	AUROC	XGBoost	0.925 ± 0.012	0.883 ± 0.021	5
	LD50	MAE	MACCS keys + autoML	0.588 ± 0.005	**0.577 ± 0.019**	1

4 Conclusion

In this paper we report MolE, which uses a Transformer with disentangled attention (DeBERTa) to predict chemical and biological properties directly from molecular graphs. Our contributions are: (1) We showed that Transformers can use molecular graphs as input when atom environments are the tokens and relative position embeddings are used. (2) We proposed a new, powerful self-supervised approach for training molecular graphs where we predict atom environments of radius > 0 from atom environments of radius 0. (3) By using a two-step (self-supervised then supervised) pretraining approach, we produced a molecular foundation model that achieves current state-of-the-art performance on 9 of the 22 ADMET tasks included in the Therapeutic Data Commons (Table 2).

References

[1] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are Few-Shot learners. May 2020. URL: `http://arxiv.org/abs/2005.14165`, `arXiv:2005.14165`.

[2] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: Open pre-trained transformer language models. May 2022. URL: `http://arxiv.org/abs/2205.01068`, `arXiv:2205.01068`.

[3] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. April 2022. URL: `http://arxiv.org/abs/2204.02311`, `arXiv:2204.02311`.

[4] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. June 2020. URL: `http://arxiv.org/abs/2006.03654`, `arXiv:2006.03654`.

[5] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. February 2021. URL: `http://arxiv.org/abs/2102.09548`, `arXiv:2102.09548`.

[6] Yvonne Connolly Martin. Hansch analysis 50 years on. *WIREs Computational Molecular Science*, 2(3):435–442, 2012. URL: `https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1096`, `doi:10.1002/wcms.1096`.

[7] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018. URL: `http://dx.doi.org/10.1038/s41586-018-0337-2`, `doi:10.1038/s41586-018-0337-2`.

[8] Peter Willett, John M Barnard, and Geoffrey M Downs. Chemical similarity searching. *J. Chem. Inf. Comput. Sci.*, 38(6):983–996, November 1998. `doi:10.1021/ci9800211`.

[9] Corwin Hansch and Toshio Fujita. p-σ-π analysis. a method for the correlation of biological activity and chemical structure. *J. Am. Chem. Soc.*, 86(8):1616–1626, April 1964. `doi:10.1021/ja01062a035`.

[10] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.*, 42(6):1273–1280, November 2002. `doi:10.1021/ci010132r`.

[11] David Rogers and Mathew Hahn. Extended-Connectivity fingerprints. *J. Chem. Inf. Model.*, 50:742–754, 2010.

[12] Nathan Brown, Marco Fiscato, Marwin H S Segler, and Alain C Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *J. Chem. Inf. Model.*, 59:1096–1108, 2018. URL: http://arxiv.org/abs/1811.09621, doi:10.1021/acs.jcim.8b00839.

[13] Benedek Fabian, Thomas Edlich, Héléna Gaspar, Marwin Segler, Joshua Meyers, Marco Fiscato, and Mohamed Ahmed. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv*, 2020.

[14] John J Irwin, Khanh G Tang, Jennifer Young, Chinzorig Dandarchuluun, Benjamin R Wong, Munkhzul Khurelbaatar, Yurii S Moroz, John Mayfield, and Roger A Sayle. ZINC20—A free Ultralarge-Scale chemical database for ligand discovery. *J. Chem. Inf. Model.*, 60(12):6065–6073, December 2020. doi:10.1021/acs.jcim.0c00675.

[15] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.*, 9(24):5441–5451, 2018. URL: http://dx.doi.org/10.1039/c8sc00148k, doi:10.1039/c8sc00148k.

[16] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. pages 1–22, May 2019. URL: http://arxiv.org/abs/1905.12265, arXiv:1905.12265.

[17] Gregory A Landrum. RDKit: Open-source cheminformatics. http://www.rdkit.org. URL: http://www.rdkit.org.

[18] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett, editor, *Advances in Neural Information Processing Systems*, September 2015. URL: https://proceedings.neurips.cc/paper/2015/file/f9be311e65d81a9ad8150a60844bb94c-Paper.pdf.

[19] Thomas N Kipf and Max Welling. Semi-Supervised classification with graph convolutional networks. September 2016. URL: http://arxiv.org/abs/1609.02907, arXiv:1609.02907, doi:10.48550/arXiv.1609.02907.

[20] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J. Med. Chem.*, 63(16):8749–8760, August 2020. doi:10.1021/acs.jmedchem.9b00959.

[21] Hao Tian, Rajas Ketkar, and Peng Tao. Accurate ADMET prediction with XGBoost, May 2022. URL: http://arxiv.org/abs/2204.07532, doi:10.48550/arXiv.2204.07532.

[22] H L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. Chem. Doc.*, 5(2):107–113, May 1965. URL: https://pubs.acs.org/doi/abs/10.1021/c160017a018, doi:10.1021/c160017a018.

[23] John MacCuish, Christos Nicolaou, and Norah E MacCuish. Ties in proximity and clustering compounds. *J. Chem. Inf. Comput. Sci.*, 41(1):134–146, January 2001. doi:10.1021/ci000069q.

[24] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. DeepTox: Toxicity prediction using deep learning. *Front. Environ. Sci. Eng. China*, 3(80), 2016. URL: https://www.frontiersin.org/articles/10.3389/fenvs.2015.00080, doi:10.3389/fenvs.2015.00080.

[25] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.*, (8):3370–3388, April 2019. URL: http://arxiv.org/abs/1904.01561, arXiv:1904.01561, doi:10.1021/acs.jcim.9b00237.

[26] Robin Winter, Floriane Montanari, Frank Noé, and Djork Arné Clevert. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.*, 10(6):1692–1701, 2019. URL: http://dx.doi.org/10.1039/c8sc04175j, doi:10.1039/c8sc04175j.

[27] Seyone Chithrananda and Bharath Ramsundar. ChemBERTa: Utilizing Transformer-Based attention for understanding chemistry.

[28] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. ChemBERTa-2: Towards chemical foundation models. September 2022. URL: http://arxiv.org/abs/2209.01712, arXiv:2209.01712.

[29] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. SMILES-BERT: Large scale unsupervised Pre-Training for molecular property prediction. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '19, pages 429–436, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3307339.3342186.

[30] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chem. Sci.*, 9(2):513–530, 2018. URL: http://dx.doi.org/10.1039/c7sc02664a, doi:10.1039/c7sc02664a.

[31] Bharath Ramsundar, Peter Eastman, Patrick Walters, and Vijay Pande. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*. "O'Reilly Media, Inc.", April 2019. URL: https://play.google.com/store/books/details?id=5uiRDwAAQBAJ.

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. October 2018. URL: http://arxiv.org/abs/1810.04805, arXiv:1810.04805, doi:10.48550/arXiv.1810.04805.

[33] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022. URL: http://arxiv.org/abs/2102.10056, doi:10.1038/s42256-022-00447-x.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, December 2017. URL: http://arxiv.org/abs/1706.03762, doi:10.48550/arXiv.1706.03762.

[35] David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, February 1988. URL: https://pubs.acs.org/doi/abs/10.1021/ci00057a005, doi:10.1021/ci00057a005.

[36] David Weininger, Arthur Weininger, and Joseph L Weininger. SMILES. 2. algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.*, 29(2):97–101, May 1989. URL: https://pubs.acs.org/doi/abs/10.1021/ci00062a008, doi:10.1021/ci00062a008.

[37] Mario Krenn, Florian Häse, Akshat Kumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4), 2020. URL: http://arxiv.org/abs/1905.13741, doi:10.1088/2632-2153/aba947.

A Related work

A.1 Molecular fingerprints

In the last decades, the most common way to encode molecules for machine learning models is using molecular fingerprints. Among the most popular are MACCs keys (10) and ECFPs (11). MACCs keys encode a molecule into a binary vector usually of 166 bits though they can be longer if needed. Each bit encodes the presence or absence of a particular pre-defined chemical group or substructure in the molecule. In a similar way, ECFPs encode molecular substructures into a binary vector of fixed length using the Morgan algorithm (22). An alternative to ECFPs are functional-class fingerprints (FCFPs) which use pharmacophoric information for atom identifiers, namely, hydrogen-bond acceptor/donor, negatively/positively ionizable, aromatic, and halogen. This is a more abstract form of fingerprint which represents equivalent chemical groups (e.g., halogens) in the same way irrespective of the atom type resulting in functional atom environments.

Despite the fact that ECFPs and FCFPs represent the molecular structure in an appropriate way for similarity search and have been successfully used for machine learning, they are some issues. The use of a hash function to project them into a defined length bit vector causes 'collisions', meaning that two atom environments can be mapped to the same bit. The number of collisions and hence information loss increases as the length of the bit vector decreases (23).

A.2 Molecular property prediction

Different machine learning methods have been used for molecular property prediction. For example, models like multilayer perceptrons and XGBoost, in combination with molecular fingerprints, have been used to successfully predict bioactivity and toxicity of molecules (24; 15; 21). Nonetheless, more suitable architectures such as message passing neural networks (MPNNs), a form of graph neural network (GNN), have shown improvements in different tasks such as the prediction of biological activity, toxicity, and quantum and physicochemical properties (25). Also, several approaches have taken advantage of NLP architectures (e.g., RNNs and Transformers) since molecules can be represented by SMILES (26; 13; 27; 28; 29).

It is important to mention that many of these models have been trained and tested on the same datasets, however it is difficult to compare them in a robust and fair manner that represents their performance in real-world applications (5). Initially, frameworks like MoleculeNet (30) and DeepChem (31) compiled different datasets and tasks for use as molecular benchmarks. More recently, Huang et al. developed the Therapeutic Data Commons (TDC) (5) a framework for systematic evaluation of machine learning models across a variety of pharmacological tasks. All models reported in this paper were benchmarked against the ADMET datasets in TDC.

A.3 Self-supervised pretraining for chemistry models

Though pretraining a model on large unlabeled datasets is a common and successful practice for language modeling, it is less common for molecular models. Some approaches (13; 27; 28; 29) use a masked language model similar to BERT (32) for predicting the identity of masked tokens of a SMILES string based on the rest of the SMILES context. After pretraining, the model is fine-tuned for different downstream tasks. A different approach based on SMILES was presented by Winter et al. (26) where they trained an autoencoder on 72 million molecules and used the latent space embeddings as inputs for a multilayer perceptron.

On the other hand, pretraining strategies for molecular graphs are not as straightforward. For example, Hu et al. pretrained GNNs using context prediction or attribute masking (16). Context prediction uses a binary classifier to indicate whether a particular atom environment corresponds to a particular context graph (i.e., nodes beyond the atom environment). In attribute masking, random nodes are masked and the task is to predict their attributes such as atom type or chirality. Wang et al. proposed an alternative pretraining strategy based on contrastive learning (33). They applied different augmentation strategies (e.g., atom masking, bond deletion, subgraph removal) and generated two correlated molecular graphs for each molecule in the training set. They proceeded to encode each augmented graph with a GNN using a contrastive loss to maximize the agreement between the two latent vectors corresponding to the same parent molecule. This approach seems suitable for learning meaningful embeddings for the whole molecule and not just the atomic level.

B Method

B.1 Model

Transformer architectures are powerful models for performing NLP tasks (34). However, they need to be adapted for tasks that are invariant to input order. A solution to this problem is to remove the typical positional encoding used and instead use relative position embeddings. For this reason, we chose to use a variation of the DeBERTa model (4). DeBERTa is a Transformer architecture in which attention weights include information about a token's content and its relative position to other tokens. In fact, standard self-attention consists of queries, keys and values $Q, K, V \in R^{N \times d}$, and is calculated as:

$$\begin{aligned} A &= \frac{QK^T}{\sqrt{d}} \\ H_0 &= softmax(A)V \end{aligned} \tag{1}$$

where $H_0 \in R^{N \times d}$ is the output hidden vectors after self-attention, and d the hidden dimension. In DeBERTa, the disentangled self attention incorporates relative position information for each K and Q pair:

$$\begin{aligned} a_{ij} &= Q_i^c K_j^{cT} + Q_i^c K_{i,j}^{p\ T} + K_j^c Q_{j,i}^{p\ T} \\ A &= \frac{a}{\sqrt{3d}} \\ H_0 &= softmax(A)V^c \end{aligned} \tag{2}$$

where $Q^c, K^c, V^c \in R^{N \times d}$ are context queries, keys and values that contains information of the token (equivalent to the ones in the standard self-attention) but also includes positional queries and keys $Q_{i,j}^p, K_{i,j}^p \in R^{N \times d}$ to encode the relative positions of tokens i and j.

B.2 Pretraining strategy

As mentioned before, self-supervised pretraining is a good alternative to transfer information from large unlabeled datasets to smaller datasets with labels. Here we present a two-step pretraining strategy as shown in Figure 2. The first step is a self-supervised approach to learn to represent chemical structures. For this we use a BERT-like approach in which each atom is randomly masked with a probability of 15%, from which 80% of the selected tokens are replaced by a mask token, 10% replaced by a random token from the vocabulary, and 10% are not changed. Different from BERT, the prediction task is not to predict the identity of the masked token, but to predict the corresponding atom environment (or functional atom environment) of radius 2, that is, all atoms and their associated bonds that are separated from the masked atom by two or less bonds. The motivation of this step is to incentivize the model to aggregate information from neighboring atoms when learning embeddings of local molecular features. The second step uses a graph-level pretraining in a supervised way with a large labeled dataset. As proposed by Hu et al. (16), combining node- and graph-level pretraining helps to learn local and global features that improve the final prediction performance.

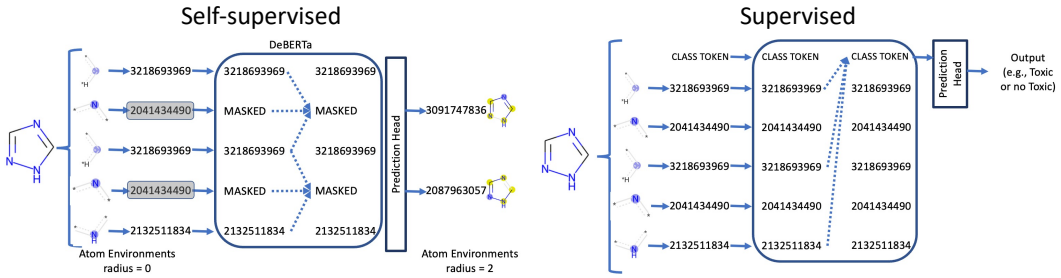

Figure 2: Pretraining approaches used in this study. a) A self-supervised approach in which an input atom is masked and the task is to predict its radius-2 atom environment. Note that in this particular example the two masked tokens are the same (2041434490), but the atom environment associated with each is different. b) A supervised approach in which a class token is added to the input and predictions are made using its final embeddings.

B.3 Molecular representation

The way molecules are represented is of great importance for property prediction. The fact that we use a language model does not mean we are required to use a string-based representation of molecules such as SMILES (35; 36) or SELFIES (37). In fact, the way in which DeBERTa calculates disentangled attention allows MolE to directly work with graphs by providing node feature information at the token level and graph connectivity via the relative position information. In this study, we used the Morgan algorithm as implemented in RDKit (17; 22) to calculate atom environments. Each atom in the molecule is initially represented by the identifier of its atom environment of radius 0. This contains information regarding the atom and all bonds attached to it, without including information of neighboring atoms. We used the same strategy to generate atom environments and functional atom environments of radius 2 for use as labels.

In addition to tokens, DeBERTa also takes relative position information as input which is an important inductive bias since it gives information about connectivity. In the case of MolE, the graph connectivity is given as a distance matrix where the i, jth entry corresponds to the length of the shortest path of bonds between atoms i and j.

C Benchmark datasets

Table 3: Summary of tasks and input/output vocabularies used for self-supervised and supervised pretraining.

	Self-supervised pretraining		Supervised pretraining	
Label type	Input (Radius 0)	Output (Radius 2)	Molecules	Tasks
Atom Envs	207	141K	456K	1,310
Functional Envs	207	114K	456K	1,310

Table 4: Description of the TDC ADMET datasets and tasks.

	Dataset	Metric	Size
Absorption	Caco2	MAE	906
	HIA	AUROC	578
	Pgp	AUROC	1,212
	Bioavailability	AUROC	640
	Lipophilicity	MAE	4,200
	Solubility	MAE	9,982
Distribution	BBB	AUROC	1,975
	PPBR	MAE	1,797
	VDss	Spearman	1,130
Metabolism	CYP2D6 inhibition	AUPRC	13,130
	CYP3A4 inhibition	AUPRC	12,328
	CYP2C9 inhibition	AUPRC	12,092
	CYP2D6 substrate	AUPRC	664
	CYP3A4 substrate	AUROC	667
	CYP2C9 substrate	AUPRC	666
Excretion	Half life	Spearman	667
	Clearance microsome	Spearman	1,102
	Clearance hepatocyte	Spearman	1,020
Toxicity	hERG	AUROC	648
	Ames	AUROC	7,255
	DILI	AUROC	475
	LD50	MAE	7,385

D Extended results

Table 5: Results of models without any pretraining or with only supervised pretraining on the TDC ADMET tasks. Underlined values show cases where the model outperforms results on the TDC leaderboard (as of September 2022).

	Dataset	Metric	Best model in TDC Leaderboard	MolE No pretraining	MolE only Supervised pretraining
Absorption	Caco2	MAE	0.291 ± 0.015	0.400 ± 0.016	0.392 ± 0.025
	HIA	AUROC	0.988 ± 0.002	0.969 ± 0.004	**0.988 ± 0.002**
	Pgp	AUROC	0.929 ± 0.010	0.848 ± 0.021	0.896 ± 0.010
	Bioavailability	AUROC	0.748 ± 0.033	0.656 ± 0.026	0.566 ± 0.066
	Lipophilicity	MAE	0.535 ± 0.012	0.624 ± 0.025	0.586 ± 0.020
	Solubility	MAE	0.734 ± 0.006	0.808 ± 0.031	0.843 ± 0.013
Distribution	BBB	AUROC	0.907 ± 0.002	0.854 ± 0.010	0.872 ± 0.021
	PPBR	MAE	8.252 ± 0.190	8.790 ± 0.525	8.949 ± 0.396
	VDss	Spearman	0.627 ± 0.009	0.607 ± 0.039	**0.642 ± 0.013**
Metabolism	CYP2D6 inhibition	AUPRC	0.717 ± 0.001	0.601 ± 0.017	0.620 ± 0.003
	CYP3A4 inhibition	AUPRC	0.872 ± 0.005	0.802 ± 0.028	0.818 ± 0.005
	CYP2C9 inhibition	AUPRC	0.769 ± 0.000	0.710 ± 0.009	0.736 ± 0.007
	CYP2D6 substrate	AUPRC	0.677 ± 0.047	0.576 ± 0.061	0.665 ± 0.024
	CYP3A4 substrate	AUROC	0.677 ± 0.007	0.556 ± 0.017	0.606 ± 0.018
	CYP2C9 substrate	AUPRC	0.433 ± 0.017	0.374 ± 0.047	0.351 ± 0.013
Excretion	Half life	Spearman	0.392 ± 0.065	0.386 ± 0.043	**0.477 ± 0.027**
	Clearance microsome	Spearman	0.597 ± 0.025	0.443 ± 0.019	0.508 ± 0.022
	Clearance hepatocyte	Spearman	0.439 ± 0.026	0.336 ± 0.014	0.413 ± 0.015
Toxicity	hERG	AUROC	0.874 ± 0.014	0.704 ± 0.026	0.816 ± 0.012
	Ames	AUROC	0.859 ± 0.000	0.782 ± 0.009	0.792 ± 0.005
	DILI	AUROC	0.925 ± 0.012	0.897 ± 0.006	0.900 ± 0.008
	LD50	MAE	0.588 ± 0.005	0.691 ± 0.024	0.649 ± 0.017

Table 6: Results of pretrained models on the ADMET group of the TDC benchmark. Underlined values show cases where the model outperforms results on the TDC leaderboard (as of September 2022).

	Dataset	Metric	MolE (AtomEnvs)	MolE (FunctionalEnvs)	MolE (AtomEnvs) + Supervised	MolE (FunctionalEnvs) + Supervised
Absorption	Caco2	MAE	0.471 ± 0.049	0.355 ± 0.025	0.310 ± 0.010	0.366 ± 0.063
	HIA	AUROC	0.949 ± 0.004	0.951 ± 0.016	0.963 ± 0.019	0.974 ± 0.011
	Pgp	AUROC	0.871 ± 0.021	0.873 ± 0.023	0.915 ± 0.005	0.902 ± 0.014
	Bioavailability	AUROC	0.683 ± 0.011	0.638 ± 0.027	0.654 ± 0.028	0.705 ± 0.029
	Lipophilicity	MAE	**0.464 ± 0.008**	**0.460 ± 0.007**	**0.469 ± 0.009**	**0.467 ± 0.006**
	Solubility	MAE	0.810 ± 0.020	0.799 ± 0.007	0.792 ± 0.005	0.771 ± 0.009
Distribution	BBB	AUROC	0.895 ± 0.006	0.901 ± 0.008	0.903 ± 0.005	0.903 ± 0.007
	PPBR	MAE	**8.191 ± 0.189**	8.570 ± 0.237	**8.073 ± 0.335**	8.379 ± 0.236
	VDss	Spearman	0.596 ± 0.020	0.622 ± 0.021	**0.654 ± 0.031**	**0.648 ± 0.024**
Metabolism	CYP2D6 inhibition	AUPRC	0.665 ± 0.025	0.678 ± 0.010	0.682 ± 0.008	0.693 ± 0.007
	CYP3A4 inhibition	AUPRC	0.865 ± 0.005	0.857 ± 0.006	0.867 ± 0.003	0.862 ± 0.003
	CYP2C9 inhibition	AUPRC	**0.773 ± 0.006**	0.759 ± 0.015	**0.801 ± 0.003**	**0.797 ± 0.004**
	CYP2D6 substrate	AUPRC	**0.706 ± 0.023**	**0.715 ± 0.011**	**0.699 ± 0.018**	**0.728 ± 0.025**
	CYP3A4 substrate	AUROC	0.633 ± 0.012	0.612 ± 0.014	0.670 ± 0.018	0.669 ± 0.012
	CYP2C9 substrate	AUPRC	0.429 ± 0.092	0.411 ± 0.027	**0.446 ± 0.062**	**0.452 ± 0.024**
Excretion	Half life	Spearman	**0.518 ± 0.045**	**0.579 ± 0.035**	**0.549 ± 0.024**	**0.537 ± 0.032**
	Clearance microsome	Spearman	0.531 ± 0.024	0.567 ± 0.016	**0.607 ± 0.027**	**0.598 ± 0.027**
	Clearance hepatocyte	Spearman	0.367 ± 0.039	0.373 ± 0.027	0.381 ± 0.038	0.381 ± 0.016
Toxicity	hERG	AUROC	0.844 ± 0.021	0.871 ± 0.007	0.823 ± 0.009	0.829 ± 0.004
	Ames	AUROC	0.832 ± 0.019	0.831 ± 0.017	0.813 ± 0.005	0.808 ± 0.009
	DILI	AUROC	0.883 ± 0.021	0.890 ± 0.016	0.883 ± 0.021	0.906 ± 0.016
	LD50	MAE	**0.582 ± 0.010**	0.597 ± 0.027	**0.577 ± 0.019**	0.651 ± 0.032

Table 7: Results of pretrained models using logP prediction as auxiliary loss during self-supervised pretraining. Underlined values show cases where the model outperforms results on the TDC leaderboard (as of September 2022).

	Dataset	Metric	MolE (AtomEnvs + logP)	MolE (FunctionalEnvs + logP)	MolE (AtomEnvs + logP)+ Supervised	MolE (FunctionalEnvs + logP) + Supervised
Absorption	Caco2	MAE	0.344 ± 0.013	0.389 ± 0.049	0.355 ± 0.049	0.325 ± 0.027
	HIA	AUROC	0.959 ± 0.005	0.969 ± 0.013	0.959 ± 0.006	0.983 ± 0.007
	Pgp	AUROC	0.880 ± 0.018	0.867 ± 0.018	0.919 ± 0.014	0.921 ± 0.013
	Bioavailability	AUROC	0.633 ± 0.014	0.622 ± 0.038	0.660 ± 0.034	0.698 ± 0.046
	Lipophilicity	MAE	**0.454 ± 0.006**	**0.448 ± 0.006**	**0.451 ± 0.014**	**0.448 ± 0.009**
	Solubility	MAE	0.775 ± 0.010	0.776 ± 0.018	0.758 ± 0.005	0.770 ± 0.013
Distribution	BBB	AUROC	0.874 ± 0.017	0.880 ± 0.013	0.889 ± 0.011	**0.914 ± 0.012**
	PPBR	MAE	**7.993 ± 0.124**	**7.926 ± 0.255**	8.605 ± 0.233	**8.128 ± 0.209**
	VDss	Spearman	**0.643 ± 0.020**	0.612 ± 0.024	**0.658 ± 0.014**	**0.641 ± 0.019**
Metabolism	CYP2D6 inhibition	AUPRC	0.666 ± 0.011	0.655 ± 0.029	0.700 ± 0.003	0.697 ± 0.002
	CYP3A4 inhibition	AUPRC	0.853 ± 0.007	0.847 ± 0.004	**0.876 ± 0.001**	0.867 ± 0.010
	CYP2C9 inhibition	AUPRC	0.758 ± 0.019	0.765 ± 0.022	**0.790 ± 0.004**	**0.791 ± 0.004**
	CYP2D6 substrate	AUPRC	0.645 ± 0.017	**0.680 ± 0.019**	**0.729 ± 0.015**	0.648 ± 0.024
	CYP3A4 substrate	AUROC	0.613 ± 0.019	0.639 ± 0.016	0.628 ± 0.026	0.643 ± 0.013
	CYP2C9 substrate	AUPRC	0.374 ± 0.069	**0.442 ± 0.023**	0.424 ± 0.067	0.420 ± 0.013
Excretion	Half life	Spearman	**0.492 ± 0.050**	**0.543 ± 0.033**	**0.533 ± 0.042**	**0.529 ± 0.046**
	Clearance microsome	Spearman	0.571 ± 0.024	0.574 ± 0.034	0.594 ± 0.012	**0.615 ± 0.017**
	Clearance hepatocyte	Spearman	0.376 ± 0.041	0.374 ± 0.043	0.417 ± 0.027	0.407 ± 0.020
Toxicity	hERG	AUROC	0.843 ± 0.022	0.817 ± 0.012	0.842 ± 0.010	0.818 ± 0.024
	Ames	AUROC	0.827 ± 0.007	0.833 ± 0.008	0.790 ± 0.017	0.821 ± 0.008
	DILI	AUROC	0.898 ± 0.007	0.874 ± 0.088	0.868 ± 0.013	0.880 ± 0.018
	LD50	MAE	0.597 ± 0.018	0.605 ± 0.035	0.603 ± 0.013	0.608 ± 0.005

Table 8: Results of pretrained models using fingerprint prediction as auxiliary loss during self-supervised pretraining. Underlined values show cases where the model outperforms results in the TDC leaderboard (as of September 2022).

	Dataset	Metric	MolE (AtomEnvs + FP)	MolE (FunctionalEnvs + FP)	MolE (AtomEnvs + FP)+ Supervised	MolE (FunctionalEnvs + FP) + Supervised
Absorption	Caco2	MAE	0.371 ± 0.032	0.341 ± 0.021	0.331 ± 0.039	0.372 ± 0.014
	HIA	AUROC	0.939 ± 0.009	0.956 ± 0.013	0.972 ± 0.007	0.945 ± 0.007
	Pgp	AUROC	0.891 ± 0.016	0.905 ± 0.011	0.902 ± 0.004	0.906 ± 0.009
	Bioavailability	AUROC	0.674 ± 0.04	0.662 ± 0.026	0.646 ± 0.059	0.678 ± 0.046
	Lipophilicity	MAE	**0.473 ± 0.007**	**0.457 ± 0.009**	**0.48 ± 0.011**	**0.483 ± 0.017**
	Solubility	MAE	0.825 ± 0.017	0.79 ± 0.012	0.803 ± 0.012	0.789 ± 0.01
Distribution	BBB	AUROC	0.891 ± 0.032	0.904 ± 0.021	0.903 ± 0.008	0.895 ± 0.038
	PPBR	MAE	8.507 ± 0.289	8.392 ± 0.181	**8.224 ± 0.183**	**8.105 ± 0.175**
	VDss	Spearman	0.605 ± 0.043	0.611 ± 0.024	**0.645 ± 0.033**	**0.662 ± 0.009**
Metabolism	CYP2D6 inhibition	AUPRC	0.677 ± 0.014	0.676 ± 0.018	0.668 ± 0.013	0.687 ± 0.007
	CYP3A4 inhibition	AUPRC	0.861 ± 0.005	0.87 ± 0.006	**0.874 ± 0.004**	0.867 ± 0.006
	CYP2C9 inhibition	AUPRC	0.753 ± 0.01	**0.781 ± 0.003**	**0.784 ± 0.003**	**0.8 ± 0.003**
	CYP2D6 substrate	AUPRC	**0.712 ± 0.018**	**0.698 ± 0.031**	0.669 ± 0.026	**0.713 ± 0.042**
	CYP3A4 substrate	AUROC	0.639 ± 0.021	0.648 ± 0.01	0.652 ± 0.013	0.664 ± 0.007
	CYP2C9 substrate	AUPRC	0.41 ± 0.027	0.415 ± 0.053	**0.459 ± 0.015**	**0.443 ± 0.016**
Excretion	Half life	Spearman	**0.523 ± 0.052**	**0.517 ± 0.049**	**0.515 ± 0.065**	**0.58 ± 0.03**
	Clearance microsome	Spearman	0.537 ± 0.024	0.532 ± 0.016	0.55 ± 0.023	0.572 ± 0.016
	Clearance hepatocyte	Spearman	0.403 ± 0.038	0.4 ± 0.018	0.384 ± 0.04	0.399 ± 0.042
Toxicity	hERG	AUROC	0.824 ± 0.007	0.846 ± 0.022	0.827 ± 0.023	0.824 ± 0.011
	Ames	AUROC	0.831 ± 0.005	0.831 ± 0.009	0.786 ± 0.012	0.808 ± 0.004
	DILI	AUROC	0.843 ± 0.093	0.844 ± 0.039	0.878 ± 0.012	0.889 ± 0.013
	LD50	MAE	0.621 ± 0.035	0.593 ± 0.009	0.599 ± 0.025	0.619 ± 0.007