HUGGINGMOLECULES: AN OPEN-SOURCE LIBRARY FOR TRANSFORMER-BASED MOLECULAR PROPERTY PREDICTION

Piotr Gaiński[1], Łukasz Maziarka[1 2], Tomasz Danel[1 2], Stanisław Jastrzebski[1 3]
[1]Jagiellonian University
[2]Ardigen
[3]Molecule.one
piotr.gainski@student.uj.edu.pl

ABSTRACT

Large-scale transformer-based methods are gaining popularity as a tool for predicting the properties of chemical compounds, which is of central importance to the drug discovery process. To accelerate their development and dissemination among the community, we are releasing HuggingMolecules – an open-source library, with a simple and unified API, that provides implementation of several state-of-the-art transformers for molecular property prediction. In addition, we add a comparison of these methods on several regression and classification datasets.

1 INTRODUCTION

Predicting molecule properties is a predominant task in the drug discovery pipeline. A good predictive model is therefore a key tool in this process as it can accelerate the whole pipeline as well as prevent from costly mistakes in clinical trials (Chan et al., 2019). Over the last few years, the machine learning community has started adopting more advanced artificial intelligence methods for predicting the properties of molecules, including Random Forest or Support Vector Machines over molecular fingerprints (Korotcov et al., 2017), SMILES-based neural networks (Jastrzebski et al., 2016; Segler et al., 2018) or graph neural networks (Duvenaud et al., 2015; Kearnes et al., 2016; Coley et al., 2017; Yang et al., 2019a).

Transformer-based methods (Vaswani et al., 2017), together with a large-scale semi-supervised pretraining (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019b; Clark et al., 2020) have pushed the boundaries of how neural networks understand the natural language and became the state-of-the-art methods in multiple NLP tasks, like token and sequence classification, question answering or language generation (He et al., 2020; Brown et al., 2020). Transformers caused an industry-wide shift in how neural networks are applied in NLP, where the primary choice is now a big model, pre-trained on a large unsupervised corpora and then fine-tuned on a downstream task, instead of a model trained from scratch. This shift increased the data-efficiency of language models across multiple tasks (Wang et al., 2019).

Large pre-trained models based on the transformer architecture are also gaining popularity in the molecular property prediction tasks. The machine learning community developed the models based on SMILES representation of molecules (Chithrananda et al., 2020; Fabian et al., 2020), as well as the transformers based on the molecular graph representations (Maziarka et al., 2020; Rong et al., 2020). We believe that in order to accelerate the development of transformer-based methods in chemistry and to spread their use among practitioners, it is necessary to create one package in which it will be possible to use different models in a simple, consistent way, as in the case of huggingface-transformers in NLP (Wolf et al., 2020). Moreover, the literature lacks a consistent comparison between these methods, which would help in choosing the right model for the right molecular task.

The contribution of this work is as follows:

1. We propose HuggingMolecules – an open-source python library for a simple, consistent usage of different transformer-based methods for molecular property prediction tasks (see Appendix A).
2. We make a strict comparison of transformer-based methods implemented in our library on many molecular property prediction datasets, for both regression and classification tasks.

Our HuggingMolecules package is available at `github.com/gmum/huggingmolecules`. Inside the package, one can find off-the-shelf models, with included pre-trained weights, ready to be used in a wide range of chemical prediction tasks. We believe this set of transformer models will accelerate research in the field of small-molecule cheminformatics.

2 METHODS

2.1 MODELS

HuggingMolecules includes implementation of 4 transformer-based methods, together with their pre-trained weights (see Table 1). Moreover, we add a popular non-Transformer model named D-MPNN (Yang et al., 2019a), to better embed our benchmark in the literature.

Table 1: Models used in our benchmark.

Model name	Citation	Type	No. params
MAT	Maziarka et al. (2020)	graph-based	42M
GROVER	Rong et al. (2020)	graph-based	48M/107M
ChemBERTa	Chithrananda et al. (2020)	SMILES-based	83M
MolBert	Fabian et al. (2020)	SMILES-based	85M
D-MPNN	Yang et al. (2019a)	graph-based	355k

In our benchmark we use 3 versions of MAT – pretrained on 200k, 2M and 20M of molecules from ZINC 15 database (Sterling & Irwin, 2015). We also use 2 versions of GROVER – base version, with 48M trainable parameters and large version, with 107M trainable parameters. Moreover we tested 3 different versions of D-MPNN – vanilla version (refered as D-MPNN), model with *rdkit_2d_normalized* features generator, where the information of additional 200 assorted rdkit descriptors is added (D-MPNN 2d) and model with *morgan_count* features generator (D-MPNN mc), where the information about the molecular count-based Morgan fingerprint (Rogers & Hahn, 2010), with radius 2 and 2048 bits is added to the model.

2.2 DATASETS

HuggingMolecules incorporates Therapeutics Data Commons (TDC) framework (Huang et al., 2021) to systematically access the entire range of datasets used both in chemistry and the drug discovery process. Additionally, our package allows to conveniently use any custom dataset in order to circumvent the limitations of TDC framework. For the purpose of our benchmark we used datasets described in Table 2. For all the datasets we used 80:10:10 split ratio (the trainining set contains 80% of the data, the validation set - 10%, and the test set - 10%).

2.3 EXPERIMENTAL SETTINGS

We performed our benchmark in the following way: on the given dataset we fine-tune the given model on 10 learning rates and 6 data splits (60 fine-tunings in total). Then we choose the learning rate that optimizes an averaged (on 6 data splits) validation metric (metric computed on the validation dataset, e.g. RMSE or ROC AUC). The final result of the benchmark is the averaged value of the metric computed on the test set for the chosen learning rate. This benchmark measures the ability of a model to generalize well on a dataset with a small search budget. We believe it roughly renders the usual way the HuggingMolecules package is going to be used - by researchers with small or medium computing resources.

Table 2: Datasets used in our benchmark.

Dataset	Category	Task type	Compounds	Label scaling	Metric	Split method	from TDC
FreeSolv	ADME	regression	642	-	RMSE	random	yes
Caco-2	ADME	regression	910	-	RMSE	random	yes
Clearance	ADME	regression	731	normalization	RMSE	random	yes
QM7	ADME	regression	6830	-	MAE	random	no
HIA	ADME	classification	578	-	ROC AUC	random	yes
Bioavailability	ADME	classification	640	-	ROC AUC	random	yes
PPBR	ADME	classification	765	-	ROC AUC	random	yes
BBBP	ADME	classification	2039	-	ROC AUC	scaffold	no
Tox21 (NR-AR)	ADME	classification	7256	-	ROC AUC	random	yes

3 RESULTS

Results are presented in Tables 3 and 4. We also include rank plots in Figure 1 (separate rank plots for regression and classification tasks can be found in the Appendix B). From the given scores we can see that graph-based transformers (MAT and GROVER) surpass these operating on SMILES (ChemBERTa, MolBERT). Moreover, we can see that both graph-based transformer methods outperforms the non-transformer state-of-the-art D-MPNN model.

Furthermore, we observe that bigger models are usually better than the smaller ones (GROVER Large vs Base) and that models pre-trained on bigger datasets outrun these trained on the smaller ones (MAT 20M vs 2M vs 200k).

The results also show that there is no one model that works best in all the setups and for all different datasets. Oftentimes, there is a trade-off between performance and the size of a model. However, sometimes smaller models are optimal for the problem at hand. For example, MAT on average outperforms both SMILES models despite having only half the number of their parameters. Its results are excellent for the classification tasks, while for the regression tasks, e.g. QM7, GROVER yields smaller prediction errors. These observations confirm that access to a wide range of models, which can be rapidly run and tested, should be beneficial for the projects involving chemical prediction.

Table 3: Benchmark results for the regression tasks. As the metric we used MAE for QM7 and RMSE for the rest of datasets.

	FreeSolv	Caco-2	Clearance	QM7	Mean rank
MAT 200k	.913 ± .196	**.405 ± .030**	.649 ± .341	87.578 ± 15.37	5.25
MAT 2M	.898 ± .165	.471 ± .070	.655 ± .327	81.557 ± 5.08	6.75
MAT 20M	**.854 ± .197**	.432 ± .034	.640 ± .335	81.797 ± 4.17	5.0
GROVER Base	.917 ± .195	.419 ± .029	.629 ± .335	**62.27 ± 3.58**	3.25
GROVER Large	.950 ± .202	.414 ± .041	**.627 ± .340**	64.94 ± 3.62	**2.5**
ChemBERTa	1.218 ± .245	.430 ± .013	.647 ± .314	177.242 ± 1.81	8.0
MolBERT	1.027 ± .244	.483 ± .056	.633 ± .332	177.117 ± 1.79	8.0
D-MPNN	1.061 ± .168	.446 ± .064	.628 ± .339	74.83 ± 4.79	5.5
D-MPNN 2d	1.038 ± .235	.454 ± .049	.628 ± .336	77.91 ± 1.21	6.0
D-MPNN mc	.995 ± .136	.438 ± .053	**.627 ± .337**	75.58 ± 4.68	4.25

4 CONCLUSIONS

In this work we proposed HuggingMolecules – an open-source python library for a simple and consistent usage of different transformer-based methods for molecular property prediction. Moreover, we made a large comparison of different models available in HuggingMolecules. This is the first comparison of various molecular transformer-based methods known to us.

In the HuggingMolecules package we used the pre-trained weights provided by the authors of the compared models. We leave the implementation of different pre-training methods, as well as adding them to the benchmark as our future work.

Table 4: Benchmark results for the classification tasks. We used ROC AUC as the metric.

	HIA	Bioavailability	PPBR	Tox21 (NR-AR)	BBBP	Mean rank
MAT 200k	**.943 ± .015**	.660 ± .052	.896 ± .027	.775 ± .035	.709 ± .022	5.8
MAT 2M	.941 ± .013	.712 ± .076	**.905 ± .019**	**.779 ± .056**	.713 ± .022	4.2
MAT 20M	.935 ± .017	.732 ± .082	.891 ± .019	**.779 ± .056**	**.735 ± .006**	**3.4**
GROVER Base	.931 ± .021	.750 ± .037	.901 ± .036	.750 ± .085	**.735 ± .006**	4.0
GROVER Large	.932 ± .023	**.747 ± .062**	.901 ± .033	.757 ± .057	.728 ± .005	4.2
ChemBERTa	.923 ± .032	.666 ± .041	.869 ± .032	**.779 ± .044**	.717 ± .009	7.0
MolBERT	.942 ± .011	.737 ± .085	.889 ± .039	.761 ± .058	.742 ± .020	4.6
D-MPNN	.924 ± .069	.724 ± .0644	.847 ± .052	.766 ± .040	.726 ± .008	7.0
D-MPNN 2d	.900 ± .094	.712 ± .067	.874 ± .030	.775 ± .041	.724 ± .006	6.8
D-MPNN mc	.924 ± .082	.740 ± .060	.869 ± .033	.772 ± .041	.722 ± .008	6.2

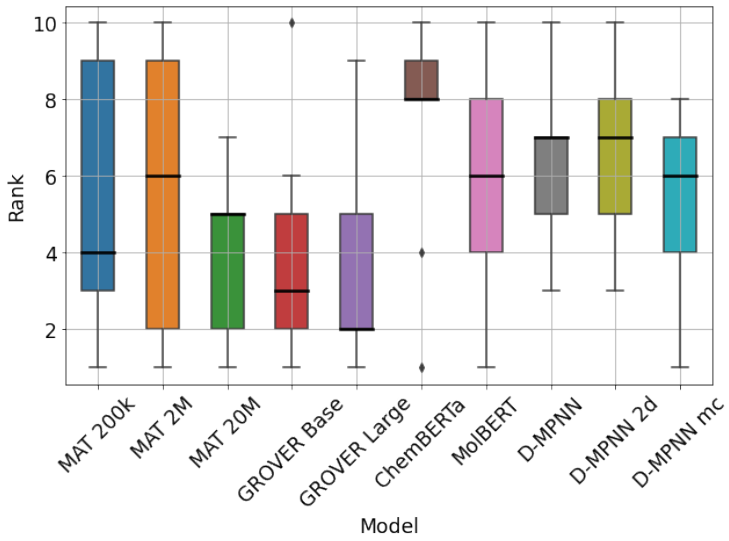

Figure 1: Rank plot for the datasets from our benchmark. We can see that the graph-base transformers outperforms these based on SMILES, moreover they beat D-MPNN, which is the non-transformer state-of-the-art in molecular property prediction tasks.

ACKNOWLEDGMENTS

This research was funded by the Priority Research Area Digiworld under the program Excellence Initiative – Research University at the Jagiellonian University in Kraków. The work of Ł. Maziarka was supported by the National Science Centre (Poland) grant no. 2019/35/N/ST6/02125. We would like to thank NVIDIA for supporting us with the computational resources required to complete this work.

REFERENCES

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

HC Stephen Chan, Hanbin Shan, Thamani Dahoun, Horst Vogel, and Shuguang Yuan. Advancing drug discovery via artificial intelligence. *Trends in pharmacological sciences*, 40(8):592–604, 2019.

Seyone Chithrananda, Gabe Grand, and Bharath Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, 57(8):1757–1772, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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. *arXiv preprint arXiv:1509.09292*, 2015.

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 preprint arXiv:2011.13230*, 2020.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.

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 therapeutics, 2021.

Stanisław Jastrzebski, Damian Leśniak, and Wojciech Marian Czarnecki. Learning to smile (s). *arXiv preprint arXiv:1602.06289*, 2016.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8): 595–608, 2016.

Alexandru Korotcov, Valery Tkachenko, Daniel P Russo, and Sean Ekins. Comparison of deep learning with multiple machine learning methods and metrics using diverse drug discovery data sets. *Molecular pharmaceutics*, 14(12):4462–4475, 2017.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Łukasz Maziarka, Tomasz Danel, Sławomir Mucha, Krzysztof Rataj, Jacek Tabor, and Stanisław Jastrzebski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33, 2020.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1): 120–131, 2018.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019a.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019b.

A Sample Code Snippet

In this section we include a sample code snippet that shows how to quickly fine-tune the MAT model on the FreeSolv dataset, using our HuggingMolecules package.

```
from huggingmolecules import MatModel, MatFeaturizer

from experiments.src import TrainingModule, get_data_loaders

from torch.nn import MSELoss
from torch.optim import Adam

from pytorch_lightning import Trainer
from pytorch_lightning.metrics import MeanSquaredError

# Build and load the pre-trained model and the appropriate featurizer:
model = MatModel.from_pretrained('mat_masking_20M')
featurizer = MatFeaturizer.from_pretrained('mat_masking_20M')

# Build the pytorch lightning training module:
pl_module = TrainingModule(model,
                           loss_fn=MSELoss(),
                           metric_cls=MeanSquaredError,
                           optimizer=Adam(model.parameters()))

# Build the data loader for the FreeSolv dataset:
train_dataloader, _, _ = get_data_loaders(featurizer,
                          batch_size=32,
                          task_name='ADME',
                          dataset_name='hydrationfreeenergy_freesolv')

# Build the pytorch lightning trainer and
# fine-tune the module on the train dataset:
trainer = Trainer(max_epochs=100)
trainer.fit(pl_module, train_dataloader=train_dataloader)

# Make the prediction for the batch of SMILES strings:
batch = featurizer(['C/C=C/C', '[C]=O'])
output = pl_module.model(batch)
```

B ADDITIONAL RESULTS

In this section we include separate rank plots for both the regression and the classification dataset (see Figures 2 and 3). The interesting conclusions can be drawn from these results. First, we can see that GROVER Base outperforms its larger version in the classification tasks. Second, MAT pre-trained on the smallest dataset (200k) is better than any other MAT versions (2M and 20M) in the regression tasks. Third, despite the slightly worse results for the regression tasks, MAT pre-trained on the medium dataset (2M) performs better than other MAT versions (200k and 20M) on the classification tasks.

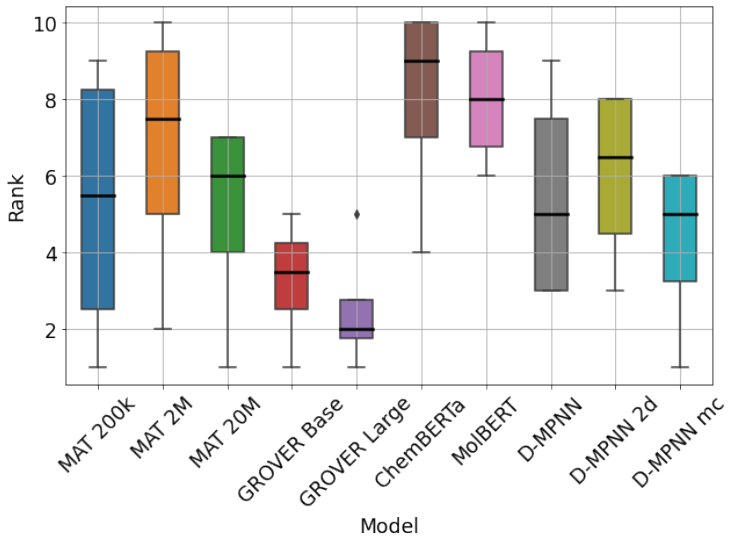

Figure 2: Rank plot for the regression tasks from our benchmark.

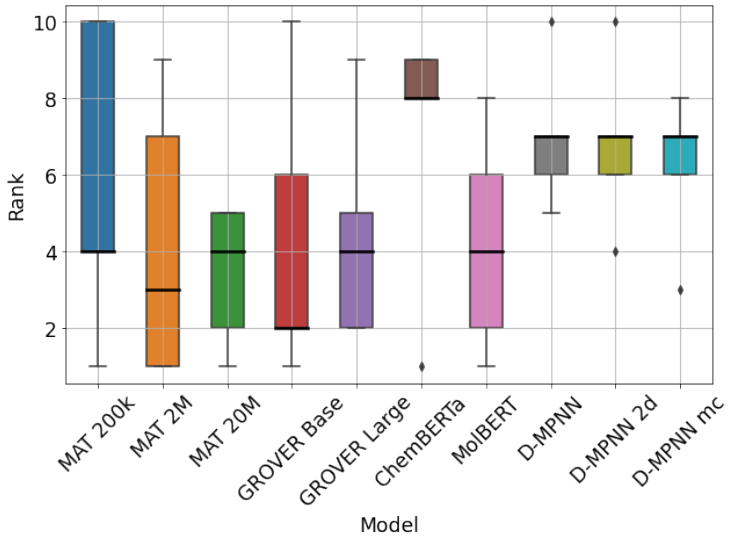

Figure 3: Rank plot for the classification tasks from our benchmark.