WHITE PAPER

Benchmark study of DeepAutoQSAR, ChemProp, and DeepPurpose on the ADMET subset of the Therapeutic Data Commons

Schrödinger

Abstract

With the advent of more powerful hardware and methods, the use of machine learning (ML) methods has seen a significant upsurge in chemistry-related applications recently. Specifically in drug discovery, the prediction of ADMET (absorption, distribution, metabolism, excretion and toxicity) properties is a main target for ML applications. Herein, we present performance metrics for Schrödinger's automated ML model building engine, DeepAutoQSAR, on the ADMET subset of the Therapeutic Data Commons (TDC) — a large collection of public data for ML model building and benchmarking. We also compare the performance of DeepAutoQSAR to the performance of two open source projects, namely ChemProp and DeepPurpose.

DeepAutoQSAR is among the top performing methods in 20 of the 22 investigated cases, clearly outperforming the other methods in 9 of those. For the other 11 cases, at least one of the other tested methods performs similarly. We believe that continuous development and further improvement of DeepAutoQSAR, in accuracy, robustness to chemical data shift and label efficiency will enable faster and more cost effective means of drug discovery, ultimately leading to the introduction of novel therapeutics.

Introduction

It is widely recognized that the ADMET (absorption, distribution, metabolism, excretion and toxicity) profile of novel molecules plays a key role in the successful development of new drugs. This is reinforced by the amount of time and effort spent both in academia and the pharmaceutical industry to develop reliable models to measure and predict numerous related endpoints[1]. Due to the potentially catastrophic impact of an unfavorable ADMET profile in the later stages of drug development, a common goal is to identify potential issues as early as possible.

With the rise of ultra-large on-demand libraries and DNA encoded libraries (for example Enamine REAL Space or WuXi LabNetwork), early identification of liabilities requires methods that are computationally fast, cheap, and accurate enough to evaluate hundreds of millions of compounds without discarding potentially good candidates. This obviously precludes the use of experimental *in vivo* or even in vitro methods. Modern machine learning (ML) approaches, often coined artificial intelligence (AI), can easily process millions of molecules on short timescales and low computational costs with acceptable accuracy.

In contrast to physics-based in *silico* methods, ML/AI methods require high fidelity data to be trained to predict a given endpoint. High quality training data is often unavailable; data need to be clean and well-curated, and datasets in chemistry applications are often smaller than those used in other domains like ML on images or text. These strict data requirements can limit the application of more complex ML/AI approaches since there is often insufficient amounts of training data to fit complex and accurate models.

However, recognizing the importance of profiling ADMET properties over the past decades, large pharmaceutical companies have generated a wealth of data which is often unfortunately non-public and exclusively applied for internal programs. Public data is rarer, but there are efforts to collect and aggregate public data [2] and also to share non-public data in smart ways to improve existing models while retaining data confidentiality [3].

The successes of deep learning (DL) approaches have led to a renaissance of ML/AI in chemistry applications, with a large number of both open-source and commercial software to pick from when targeting ADMET endpoints. While open-source software oftentimes can profit from faster development cycles and thus implements new scientific insights more quickly, application is often limited to domain experts. On the other hand, commercial software has the benefits of structured quality assurance (QA), documentation and support, and comes coupled with comprehensive user interfaces which significantly lower the barrier to entry for non-experts.

In this paper, we will take a closer look at the performance of two of the more popular open-source packages, ChemProp and DeepPurpose, and Schrödinger's ML/AI package DeepAutoQSAR, demonstrating their comparative performance on a recently published set of benchmarks.

Description of Datasets

In an effort to illustrate the performance differences between ML methods on relevant ADMET problems, we show model performance on a variety of datasets that represent both idealized, data-rich conditions and more limited real-world datasets. The variance in dataset size and diversity serves two purposes: firstly, we aim to explore gaps in benchmark performance in cases where plentiful data facilitates the training of expressive ML models that can take advantage of rich datasets. For these problems, models must be flexible enough to capture complex patterns in the data, as larger datasets typically contain more diverse chemical matter than is observed in real-world ADMET datasets.

Our second type of dataset is more limited, a challenge which forces models to learn generalizable rules from smaller sets of examples. These datasets, many of which come from publicly available data-dumps released by large pharmaceutical companies, test ML methods' abilities to provide accurate predictions given small amounts of training data with significant selection biases. Unlike the larger datasets, these real world data are limited in chemical diversity, and often contain only a small number of chemical series sharing common scaffolds. Among compounds with common cores, individual data points may only differ by minor modifications, changes that reflect developing project needs over the discovery effort's trajectory [4]. These training data present a particular challenge, and offer more relevant insight into how ML models may impact decision making within a drug discovery campaign.

For both of these problem types, we report ML model performance on the Therapeutic Data Commons' (TDC) ADMET and Tox prediction challenges, a collection of datasets sourced from existing public literature and databases [2].

TDC ADMET

Datasets from the TDC's ADME and Toxicity single instance prediction problems are some of the most comprehensive and well curated ADMET datasets available. The collection of 21 ADME and 8 Tox datasets contains various endpoints of practical interest, such as compound lipophilicity, aqueous solubility, and many other experimental endpoints. Many of these datasets have been aggregated by the TDC contributors from other benchmarks [5], existing databases [6], and previously published literature.

Of the available datasets, we choose a subset of 18 ADME and 4 Tox (the "ADMET Benchmark Group" as defined in the TDC publication) datasets to run our benchmarks.

The 18 ADME tasks are composed of 6 absorption endpoints, 3 distribution endpoints, 6 metabolism endpoints, and 3 excretion endpoints. The associated table 1 contains short descriptions of the relevant datasets. Additionally tables 2 (regression) and 3 (classification) show per-dataset information indicating the number of compounds, the associated performance metric as decided by the TDC benchmark authors, and the type of data split considered. Please consult [2] for a more thorough description of the datasets and their sources, and for further information on the impact of these datasets.

Table 1: The collection of ADMET datasets sourced from the TDC data repository that we consider for model benchmarking. Provided are the dataset abbreviation codes, the ADMET classification, and a high level description of the prediction task.

Dataset Abbr.	Dataset Type	Dataset Description
Caco2	Absorption	Caco-2 Effective Permeability
HIA	Absorption	Human Intestinal Absorption
Bioav	Absorption	Oral Bioavailability
Pgp	Absorption	P-glycoprotein Inhibition
Lipo	Absorption	Lipophilicity
AqSol	Absorption	Aqueous Solubility
BBB	Distribution	Blood Brain Barrier
PPBR	Distribution	Plasma Protein Binding Rate
VDss	Distribution	Volume of Distribution
CYP2C9 Inhibition	Metabolism	CYP P450 2C9 Inhibition
CYP2D6 Inhibition	Metabolism	CYP P450 2D6 Inhibition
CYP3A4 Inhibition	Metabolism	CYP P450 3A4 Inhibition
CYP2C9 Substrate	Metabolism	CYP P450 2C9 Substrate
CYP2D6 Substrate	Metabolism	CYP P450 2D6 Substrate
CYP3A4 Substrate	Metabolism	CYP P450 3A4 Substrate
Half Life	Excretion	Terminal Phase Half Life
CL-Hepa	Excretion	Hepatocyte Clearance
CL-Micro	Excretion	Microsome Clearance
LD50	Toxicity	Acute Toxicity LD50
hERG	Toxicity	hERG Inhibition
Ames	Toxicity	Ames Mutagenicity
DILI	Toxicity	Drug Induced Liver Injury

Table 2: Regression datasets from the TDC ADMET collection. Here we give information on dataset size, accuracy measurement method, and train/test split selection method.

Dataset	Num Compounds	Metric	Split
Caco2	906	MAE	Scaffold
Lipo	4200	MAE	Scaffold
AqSol	9982	MAE	Scaffold
PPBR	1797	MAE	Scaffold
VDss	1130	Spearman	Scaffold
Half Life	667	Spearman	Scaffold
CL-Hepa	1020	Spearman	Scaffold
CL-Micro	1102	Spearman	Scaffold
LD50	7385	MAE	Scaffold

Table 3: Classification datasets from the TDC ADMET collection. Here we give information on dataset size, accuracy measurement method, and train/test split selection method

Dataset	Num Compounds	Metric	Split
HIA	578	AUROC	Scaffold
Pgp	1212	AUROC	Scaffold
Bioav	640	AUROC	Scaffold
BBB	1975	AUROC	Scaffold
CYP2C9 Inhibition	12092	AUPRC	Scaffold
CYP2D6 Inhibition	13130	AUPRC	Scaffold
CYP3A4 Inhibition	12328	AUPRC	Scaffold
CYP2C9 Substrate	666	AUPRC	Scaffold
CYP2D6 Substrate	664	AUPRC	Scaffold
CYP3A4 Substrate	667	AUROC	Scaffold
hERG	648	AUROC	Scaffold
Ames	7255	AUROC	Scaffold
DILI	475	AUROC	Scaffold

Methods

Here we provide descriptions of the supervised learning models examined in our experiments. All the modeling packages we discuss implement regression and classification methods using a combination of classical machine learning methods as well as variations of deep neural networks. Owing to the intricacies of each method, we provide only a general overview of features and direct readers to links in our supplementary material for more information.

DeepAutoQSAR

DeepAutoQSAR is one of Schrödinger's tools for supervised learning on molecules. The program constructs ensembles of machine learning models to predict regression and binary classification targets given a collection of labeled examples. DeepAutoQSAR attempts to do this in an automated fashion, meaning that models are constructed without manual fine-tuning or human intervention in model selection. This type of machine learning tool is an example of AutoML, as DeepAutoQSAR requires minimal domain experience to use in exchange for increased computation time [7].

Over the course of training, DeepAutoQSAR will fit a collection of models with different architectures and hyperparameters, scoring each model configuration via cross validation on the supplied input dataset. The program then uses the cross validation scores associated with each model to rank its fitness for selection into an ensemble of well performing models. After a fixed, user-specified amount of training time, the best performing ML models are saved for later inference. At inference time, the stored models in the ensemble are inferenced independently, and their predictions are averaged to produce a final estimate. For regression, this result is an average of the continuous values contributed by each sub-model; for binary classification, each sub-model outputs a continuous value between zero and one which represents the probability of a one in a Bernoulli trial.

The ML models eligible for selection to DeepAutoQSAR's ensemble are deep neural networks (DNN), gradient boosted trees (XGBoost), and Random Forest (RF) models. The deep neural networks can be further classified by the types of molecular representations considered: dense neural networks take ECFP (Morgan) bit fingerprints as inputs, whereas Graph Convolutional (GNN) or Message Passing Neural Networks (MPNN) operate on featurized graph representations of molecules. These models are trained with standard gradient-based optimization techniques, such as variants of stochastic gradient descent, or tree pruning methods for RF and XGBoost.

To account for alternative methods of molecular featurization, DeepAutoQSAR implements several schemes for representing molecules as vectors amenable to machine learning. For per-molecule vectorization, DeepAutoQSAR uses ECFPs with a range of atomic neighborhood radii and hash-lengths, yielding a large number of possible featurizations for a given molecule. These ECFPs are further modified by including features dictating RS chirality, or additional features specified by the user. The method also featurizes molecules as vectors of cheaply computed properties via the DescriptaStorus, an encoding of molecules into 200 rdkit properties with normalized values. These property vectors can be utilized by the same ML models that operate on the ECFPs (dense NNs, XGBoost, RF) [8].

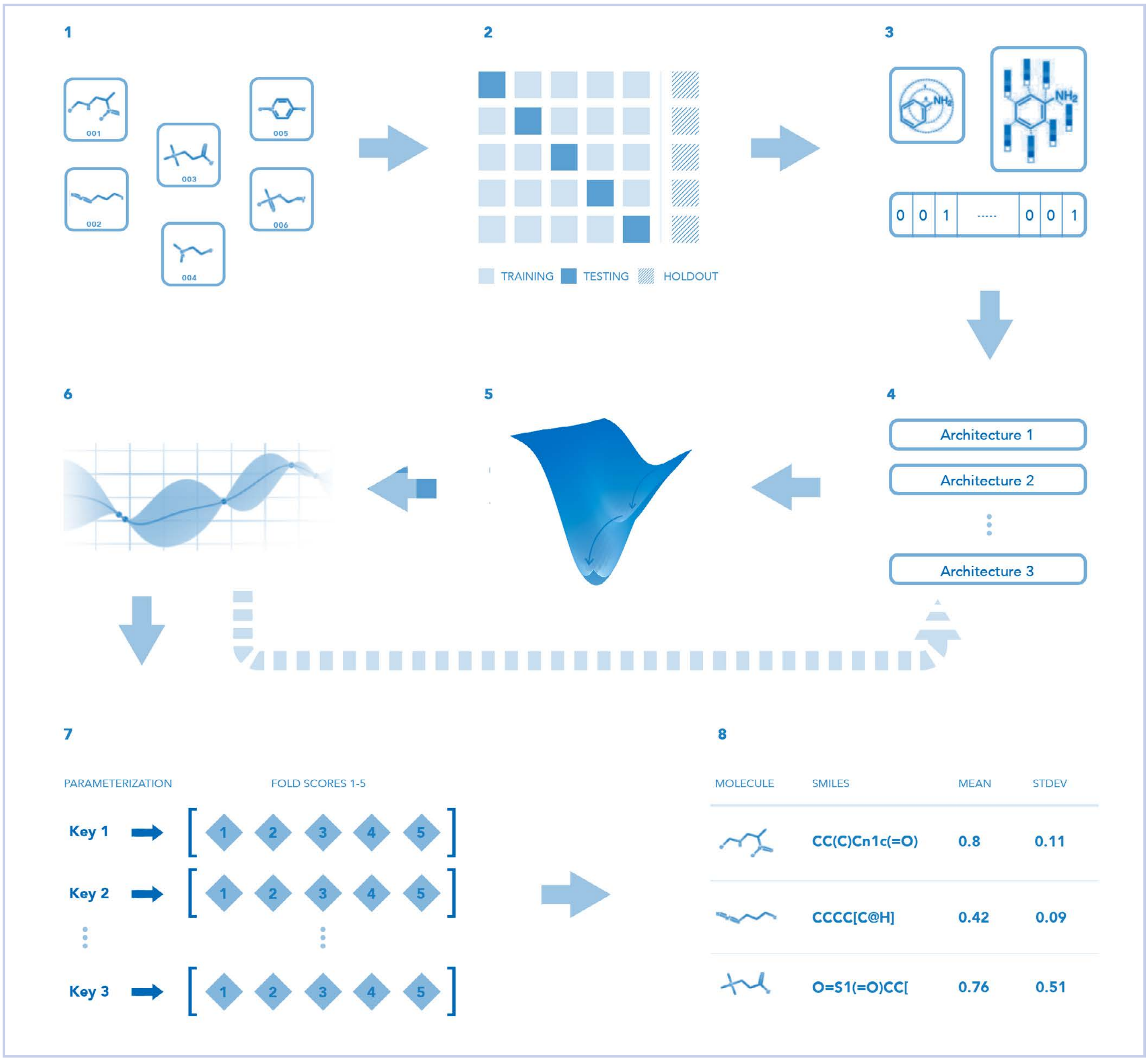
Per-atom featurizations for GNN and MPNN approaches include one-hot encodings of atomic number, implicit valence, formal charge, atomic degree, number of radical electrons, hybridization, and aromaticity. These binary feature vectors assigned to each heavy atom in the ligand are further complimented by RS chirality features and/or user specified additional features.

DeepAutoQSAR selects the independent ensemble members by performing Bayesian Optimization (BO) trials over each model architecture, thereby optimizing the hyper-parameters of each eligible model [9]. The BO trials begin with Sobol quasi-random sampling of the predefined space of hyper-parameters; the initial random samples provide training data to fit Gaussian Process Regressors that model the mean cross validation performance as a function of the selected hyper-parameters. These GPs serve as surrogate models to BO acquisition functions which define pointwise estimates of the cross validation performance as a function of selected hyper-parameters. The optima of these acquisition functions become the next suggested model parameterizations to evaluate in cross validation.

Along with model training, selection, and validation, DeepAutoQSAR performs data normalization and transformation. Data are standardized to zero mean and unit variance when applicable, and log-transformations are considered further hyper-parameters if all target values are greater than zero.

In addition to the standard supervised learning features, DeepAutoQSAR contains functionality for transfer learning via model stacking, model checkpointing and compression, and a large collection of visualizations and performance reporting [10]. Many of these functions are not relevant to the present benchmark but are vital for DeepAutoQSAR's successful deployments to internal Schrödinger projects.

Figure 1: DeepAutoQSAR Graphical Abstract: The program workflow for DeepAutoQSAR begins with data preprocessing and cross validation splitting, as well as set-up processes for model ensemble selection. Independent ML models are then trained to optimize performance on held-out molecules, yielding a final model ensemble with robust performance at inference time.



MOLECULE	SMILES	MEAN	STDEV
	CC(C)Cn1c(=O)	0.8	0.11
	CCCC[C@H]	0.42	0.09
	O=S1(=O)CC[	0.76	0.51

ChemProp

ChemProp is an open-source tool which uses message passing neural networks for molecular property prediction [11]. Like DeepAutoQSAR, ChemProp trains an ensemble of graph convolutional deep neural networks, optimizes hyper-parameters with Bayesian Optimization, and includes rdkit descriptors as features via the DescriptaStorus.

For it's core ML model, ChemProp leverages a custom network (D-MPNN) which uses both atom and bond features to construct molecular representations. ChemProp's D-MPNN aggregates messages via directed edges along the molecular graph, producing an update step which passes messages from bond-to-bond rather than from atom-to-atom.

DeepPurpose

DeepPurpose is a "deep learning based molecular modeling and prediction toolkit on drug-target interaction prediction, compound property prediction, protein-protein interaction prediction, and protein function prediction" that is freely available on Github [12]. In this study we employed it in three different flavors: Two of them use multi-layer perceptrons (MLP) on different fingerprints (Morgan and RDKit2D), and are dubbed DeepPurpose (Morgan + MLP) and DeepPurpose (RDKit2D + MLP), respectively. The third approach uses the DeepPurpose implementation of a convolutional neural network (CNN) that featurizes input SMILES strings, and will be called DeepPurpose (CNN) in this study. Details on the different methods can be found in the DeepPurpose publication [12]. For our results, we used performance scores from the TDC leaderboard rather than re-run DeepPurpose in efforts to maintain consistency with existing results [13]. Though not included in the published performance figures for the TDC benchmark, DeepPurpose has, in the interim, added more sophisticated features such as Bayesian Optimization for hyper-parameter selection. These additional features are described in the documentation and Github pages.

Results

TDC Results

Tables 4-8 summarize the results for the TDC datasets and the different ML approaches investigated herein. The datasets are grouped by endpoint type, as suggested in the original publication: Adsorption, Distribution, Metabolism, Excretion and Toxicity. We applied the same performance metrics as the TDC challenge to enable comparison to the original data and future submissions to the TDC leaderboard: Mean Absolute Error (MAE, lower values are better), Spearman's Rank Correlation Coefficient (Spearman, higher values are better), Area Under the Receiver Operator Curve (AUROC, higher values are better), and Area Under the Precision Recall Curve (AUPRC, higher values are better). Arrows next to the metrics indicate whether higher (↑) or lower (↓) values indicate improved performance. For information on the abbreviations used for the specific datasets please refer to Table 1 or the respective paragraphs below.

For stability, all methods are trained and evaluated 5 times with different random seeds. These separate instantiations are scored independently and we report the mean and standard deviation scores as *mean score ± std score*. Note that these reported uncertainties are unrelated to the models' estimations of epistemic uncertainty derived from internal mechanisms such as ensembling. We leave such an analysis as a future project.

To enable quick comparison and easy identification of the top performing method for each dataset, the best result for each is always **bolded** in the respective column. Here, we consider a result to be significantly superior when there are no other scores within the error range of the top performing method.

Table 4: Performance results on absorption datasets for DeepAutoQSAR, ChemProp, and DeepPurpose models. Each entry details mean performance and standard deviation across 5 trials.

Model (Metric)	**Bioav (AUROC, ↑)**	**HIA (AUROC, ↑)**	**Pgp (AUROC, ↑)**
DeepAutoQSAR	**0.682 ± .011**	**0.982 ± .003**	**0.917 ± .001**
RDKit2D + MLP	**0.672 ± .021**	0.972 ± .008	0.918 ± .007
Morgan + MLP	0.581 ± .086	0.807 ± .072	0.880 ± .006
CNN	0.613 ± .013	0.869 ± .026	0.908 ± .012
ChemProp	0.623 ± .025	0.979 ± .003	0.902 ± .017

Model (Metric)	**Caco2 (MAE, ↓)**	**Lipo (MAE, ↓)**	**AqSol (MAE, ↓)**
DeepAutoQSAR	**0.306 ± .012**	0.476 ± .015	**0.784 ± .023**
RDKit2D + MLP	0.393 ± .024	0.574 ± .017	0.827 ± .047
Morgan + MLP	0.908 ± .060	0.701 ± .009	1.203 ± .019
CNN	0.446 ± .036	0.743 ± .020	1.023 ± .023
ChemProp	0.390 ± .045	**0.437 ± .010**	0.820 ± 0.012

Absorption endpoints

Table 4 summarizes the results of the endpoints associated with Absorption, comprising the Caco2, HIA, Pgp, Bioav, Lipo and AqSol datasets.

Bioav

DeepAutoQSAR and DeepPurpose (RDKit2D + MLP) are tied on the oral bioavailability (Bioav) data, with areas under the receiver operating characteristic curve (AUROC) of 0.682 ± 0.011 and 0.672 ± 0.021, respectively. The worst performer on the set is DeepPurpose (Morgan + MLP) with an AUROC of 0.581 ± 0.086. The gap between the best and worst methods is a rather low ~ 0.1 on a total scale of 0 to 1. Dataset size is on the lower end with 640 data points.

HIA

For the human intestinal absorption data (HIA), DeepAutoQSAR shows the best performance with an area under the receiver operating characteristic curve (AUROC) of 0.982 ± 0.003. DeepPurpose (Morgan + MLP) performs worst with an AUROC of 0.807 ± 0.072. The spread of the methods is 0.18 on a total scale of 0 to 1. The dataset is one of the smallest investigated here with only 578 drugs.

Pgp

For the p-glycoprotein inhibition dataset, DeepAutoQSAR and DeepPurpose (RDKit2D + MLP) are tied for first place with areas under the receiver operating characteristic curve (AUROC) of 0.917 ± 0.001 and 0.918 ± 0.007. DeepPurpose (Morgan + MLP) is the worst performing method with an AUROC of 0.880 ± 0.006. Notably, the gap between best and worst performers is small (0.038) on the total scale of 0 to 1, and all methods display high AUROC values. The amount of data is low with 1212 data points.

Caco2

In the case of the human colon epithelial cancer cell line permeability data (Caco2), the clear winner is DeepAutoQSAR with a low MAE of 0.306 ± 0.012 log units (the data is log[cm/s]), followed by ChemProp with 0.390 ± 0.045 log units. The worst performing approach by a margin of ~0.6 log units is the DeepPurpose (Morgan + MLP) approach, compared to the absolute data spread of -7.76 to -3.51 log units for the endpoint. The amount of data in the set is rather low with only 906 data points.

Note that the description in the TDC publication is in error here (units are described as cm/s instead of log[cm/s]). Refer to the distributed data and the original data source[14] for further information.

Lipo

For the Lipophilicity (Lipo) dataset, ChemProp emerges as the winner with an MAE of 0.437 ± 0.010 log units, followed by DeepAutoQSAR with an MAE of 0.476 ± 0.015 log units (the data is the log of the distribution ratio). The worst performer in this case is the DeepPurpose (CNN) approach. The overall performance spread of the methods is below 0.3 log units, which can be considered low compared to the overall data spread (-1.5 to 4.5 log units). With 4200 data points in the set, it is medium-sized w.r.t. the other investigated endpoints.

AqSol

DeepAutoQSAR is the best performing method on the aqueous solubility (AqSol) dataset, with an MAE of 0.784 ± 0.023 log units (the endpoint is given in log[mol/L]). It is followed by ChemProp and DeepPurpose (RDKit2D + MLP) which perform similarly for the dataset, with ChemProp being slightly superior. The worst performing method on the dataset is DeepPurpose (Morgan + MLP) with an MAE of 1.203 ± 0.019 log units. This brings the overall performance spread to ~0.4 log units, compared to the overall data spread of (-13.17 to 2.14). The AqSol dataset is among the larger ones, with 9982 data points.

Distribution Endpoints

Table 5 summarizes the results of the endpoints associated with distribution, comprising the BBB, PPBR and VDss datasets.

BBB

In the case of the blood-brain barrier penetration data (BBB), ChemProp, DeepAutoQSAR and DeepPurpose (RDKit2D + MLP) share first place with areas under the receiver operating characteristic curve (AUROC) of 0.882 ± .012, 0.876 ± 0.011 and 0.889 ± 0.016. The worst performing method here is DeepPurpose (CNN) with an AUROC of 0.781 ± 0.030. All methods perform at a high level with a gap of ~ 0.1 between the best and worst performing methods (on a scale of 0 to 1). The dataset contains a rather small amount of 1975 drugs.

PPBR

In the case of the human plasma protein binding rate (PPBR) dataset, ChemProp and DeepAutoQSAR are tied for the best performing method with MAEs of 7.993 ± 0.234 and 8.043 ± 0.107 percentage points. DeepPurpose (Morgan + MLP) comes in last with an MAE of 12.848 ± 0.362 percentage points. The overall performance spread is ~ 4 percentage points. The amount of data in the set is on the lower end, with 1797 data points.

VDss

The volume distribution at steady state dataset (VDss) is dominated by DeepAutoQSAR, with a Spearman correlation coefficient of 0.673 ± 0.009. Second best method is DeepPurpose (RDKir2D + MLP) with a coefficient of 0.561 ± 0.025. DeepPurpose (CNN) is ranked last with a coefficient of 0.226 ± 0.114. The spread in the results is a significant 0.45 on a scale of 0 to 1. The data set contains 1130 drugs which makes it one of the smaller sets.

Metabolism Endpoints

Table 6 summarizes the results of the endpoints associated with metabolism, comprising the CYP2C9 Inhibition/Substrate, CYP2D6 Inhibition/Substrate and CYP3A4 Inhibition/Substrate datasets.

CYP2C9 Inhibition

For the cytochrome P450 enzyme CYP2C9 inhibition data, DeepAutoQSAR performed best with an area under the precision-recall curve (AUPRC) of 0.792 ± 0.007. The second best method here is ChemProp with an AUPRC of 0.770 ± .005. DeepPurpose (CNN) performs worst with an AUPRC of 0.713 ± 0.006. The spread between the results is a low 0.08 on a total scale of 0 to 1. The dataset is one of the largest with 12092 data points.

CYP2D6 Inhibition

The best performing method on the cytochrome P450 enzyme CYP2D6 inhibition data is DeepAutoQSAR, with an area under the precision-recall curve (AUPRC) of 0.702 ± 0.006. ChemProp takes second place with an AUPRC of 0.664 ± 0.012. Worst performing method for the dataset is DeepPurpose (CNN) with an AUPRC of 0.544 ± 0.053. The spread between the methods is ~ 0.15, on a total scale of 0 to 1. With 13130 data points, this is the largest dataset investigated.

Table 5: Performance results on distribution datasets for DeepAutoQSAR, ChemProp, and DeepPurpose models. Each entry details mean performance and standard deviation across 5 trials.

Model (Metric)	BBB (AUROC, ↑)	PPBR (MAE, ↓)	VDss (Spearman, ↑)
DeepAutoQSAR	**0.876 ± .011**	**8.043 ± .107**	**0.673 ± .009**
RDKit2D + MLP	**0.889 ± .016**	9.994 ± .319	0.561 ± .025
Morgan + MLP	0.823 ± .015	12.848 ± .362	0.493 ± .011
CNN	0.781 ± .030	11.106 ± .358	0.226 ± .114
ChemProp	**0.882 ± .012**	**7.993 ± .234**	0.519 ± .033

Table 6: Performance results on metabolism datasets for DeepAutoQSAR, ChemProp, and DeepPurpose models. Each entry details mean performance and standard deviation across 5 trials.

Model (Metric)	CYP2C9 Inh. (AUPRC, ↑)	CYP2D6 Inh. (AUPRC, ↑)	CYP3A4 Inh. (AUPRC, ↑)
DeepAutoQSAR	**0.792 ± .007**	**0.702 ± .006**	**0.883 ± .003**
RDKit2D + MLP	0.742 ± .006	0.616 ± .007	0.829 ± .007
Morgan + MLP	0.715 ± .004	0.587 ± .011	0.827 ± .009
CNN	0.713 ± .006	0.544 ± .053	0.821 ± .003
ChemProp	0.770 ± .005	0.664 ± .012	0.870 ± .003

Model (Metric)	CYP2C9 Sub. (AUPRC, ↑)	CYP2D6 Sub. (AUPRC, ↑)	CYP3A4 Sub. (AUROC, ↑)
DeepAutoQSAR	**0.395 ± .034**	**0.703 ± .008**	**0.642 ± .019**
RDKit2D + MLP	0.360 ± .040	0.677 ± .047	**0.639 ± .012**
Morgan + MLP	**0.380 ± .015**	0.671 ± .066	**0.633 ± .013**
CNN	**0.367 ± .059**	0.485 ± .037	**0.662 ± .031**
ChemProp	**0.391 ± .033**	0.688 ± .024	0.610 ± .029

CYP3A4 Inhibition

With an area under the precision-recall curve (AUPRC) of 0.883 ± 0.003, DeepAutoQSAR performs best on the cytochrome P450 enzyme CYP3A4 inhibition data set. ChemProp is a close second with an AUPRC of 0.870 ± 0.003. DeepPurpose (CNN) is the worst performing method, however, all methods perform within a very small window of 0.06 (on a scale of 0 to 1). The dataset is one of the largest investigated with 12328 data points.

CYP2C9 Substrate

For the cytochrome P450 enzyme CYP2C9 substrate data, most methods perform similarly with AUPRCs around 0.380. Only DeepPurpose (RDKit2D + MLP) performs slightly worse with an AUPRC of 0.360 ± 0.040. Notably, none of the methods perform very well here (on a scale of 0 to 1), and lie very close together. The dataset size is small with only 666 data points.

CYP2D6 Substrate

In the case of the cytochrome P450 enzyme CYP2D6 substrate data, DeepAutoQSAR takes first place with an area under the precision-recall curve (AUPRC) of 0.703 ± 0.008. ChemProp, DeepPurpose (Morgan + LP) and DeepPurpose (RDKit2D + MLP) are tied for a close second place with AUPRCs of 0.688 ± 0.024, 0.671 ± 0.066 and 0.677 ± 0.047, respectively. DeepPurpose (CNN) is performing significantly worse compared to the other methods with an AUPRC of only 0.485 ± 0.037. The spread of the results is ~ 0.22 on a scale of 0 to 1. Notably the top 4 out of 5 methods are only separated by 0.034. The dataset is rather small with only 664 data points.

CYP3A4 Substrate

None of the tested methods clearly outperforms the others on the cytochrome P450 enzyme CYP2D6 substrate data set; DeepAutoQSAR and all DeepPurpose methods are within each others' error estimates around an areas under the receiver operating characteristic curve (AUROC) of ~0.65. The exception to this is ChemProp, which performs slightly worse with an AUROC of 0.610 ± 0.029. The spread between the methods is ~0.05 on a scale of 0 to 1. With only 667 data points, the dataset is small.

Table 7: Performance results on excretion datasets for DeepAutoQSAR, ChemProp, and DeepPurpose models. Each entry details mean performance and standard deviation across 5 trials.

Model (Metric)	Half Life (Spearman, ↑)	CL-Hepa (Spearman, ↑)	CL-Micro (Spearman, ↑)
DeepAutoQSAR	**0.551 ± .039**	**0.432 ± .025**	**0.594 ± .027**
RDKit2D + MLP	0.184 ± .111	0.184 ± .111	**0.586 ± .014**
Morgan + MLP	0.329 ± .083	0.272 ± .068	0.492 ± .020
CNN	0.038 ± .138	0.235 ± .021	0.252 ± .116
ChemProp	0.293 ± .020	**0.423 ± .040**	**0.579 ± .019**

Table 8: Performance results on toxicity datasets for DeepAutoQSAR, ChemProp, and DeepPurpose models. Each entry details mean performance and standard deviation across 5 trials.

Model (Metric)	LD50 (MAE, ↓)	hERG (AUROC, ↑)	Ames (AUROC, ↑)	DILI (AUROC, ↑)
DeepAutoQSAR	0.59 ± .026	**0.845 ± .016**	**0.864 ± .002**	**0.933 ± .006**
RDKit2D + MLP	0.678 ± .003	0.841 ± .020	0.823 ± .011	0.875 ± .019
Morgan + MLP	0.649 ± .019	0.736 ± .023	0.794 ± .008	0.832 ± .021
CNN	0.675 ± .011	0.754 ± .037	0.776 ± .015	0.792 ± .016
ChemProp	**0.548 ± .008**	0.750 ± .019	**0.864 ± .005**	0.918 ± .008

Excretion Endpoints

Table 7 summarizes the results of the endpoints associated with excretion, comprising the Half Life, CL-Hepa and CL-Micro datasets.

Half Life

DeepAutoQSAR is the best performing method for the half life dataset, with a Spearman coefficient of 0.551 ± 0.039. DeepPurpose (Morgan + MLP) ranks second with a coefficient of 0.329 ± 0.083. DeepPurpose (CNN) ranks last, with a coefficient of 0.038 ± 0.138. Notably, the performance spread between the methods is half of the possible values (0.038 to 0.551 on a scale of 0 to 1). The amount of data available in this set is low at 667 data points.

CL-Hepa

For hepatocyte clearance data (CL-Hepa), DeepAutoQSAR and ChemProp are tied for the best performance, with Spearman correlation coefficients of 0.435 ± 0.003 and 0.423 ± 0.040. The worst performing approach here is DeepPurpose (RDKit2D + MLP) with a correlation coefficient of 0.184 ± 0.111. The performance spread between methods is a rather large ~0.25 on a total scale of 0 to 1. Dataset size is on the lower end with 1020 data points.

CL-Micro

In the case of the microsomal clearance dataset (CL-Micro), three methods are tied for the first place: ChemProp, DeepAutoQSAR and DeepPurpose (RDKit2D + MLP) with Spearman correlation coefficients of 0.579 ± 0.019, 0.594 ± 0.027 and 0.586 ± 0.014, respectively. The worst performing method is DeepPurpose (CNN) with a coefficient of 0.252 ± 0.116. Notably, this method is significantly less performant than the four others in this case, with a gap of ~0.35 to the best performing methods, and ~ 0.25 to the next best performing method (on a total scale of 0 to 1). With 1102 data points, the data set is one of the smaller sets.

Toxicity Endpoints

Table 8 summarizes the results of the endpoints associated with toxicity, comprising the LD50, hERG, Ames and DILI datasets.

LD50

ChemProp is the best performing method for the median lethal dose (LD50) dataset, with an MAE of 0.548 ± 0.008 log units (the endpoint is given in log[1/(mol/kg)]). The second best performing method is DeepAutoQSAR with an MAE of 0.59 ± 0.026 log units, while DeepPurpose (RDKit2D + MLP) exhibits the worst performance of all tested methods. Notably, all DeepPurpose based methods are rather close in this case, differing by only ~0.03 log units. Overall, the tested approaches are rather close for this dataset, with a spread of only ~0.1 log unit, compared to an overall spread of the dataset of (-0.34 to 10.20). With 7385 data points, the set can be considered medium sized w.r.t. the others.

hERG

DeepAutoQSAR and DeepPurpose (RDKit2D + MLP) are tied in performance on the human Ether-à-go-go-Related gene ion channel (hERG) inhibition data, with areas under the receiver operating characteristic curve (AUROC) of 0.845 ± 0.016 and 0.841 ± 0.020. The worst performing method in this case is DeepPurpose (Morgan + MLP) with an AUROC of 0.736 ± 0.023. The spread in the performance across methods is ~ 0.1, on a total scale of 0 to 1. Dataset size is on the lower end with 648 drugs.

Ames

With areas under the receiver operating characteristic curve (AUROC) of 0.864 ± 0.002 and 0.864 ± .005, DeepAutoQSAR and ChemProp are the best performing methods on the Ames mutation assay data. DeepPurpose (CNN) performs worst with an AUROC of 0.794 ± 0.008. Nevertheless, the performance window over all the methods is only 0.07, and all methods perform on a high level (the total scale is 0 to 1). With 7255 data points, the dataset is medium sized when compared to the other datasets in this study.

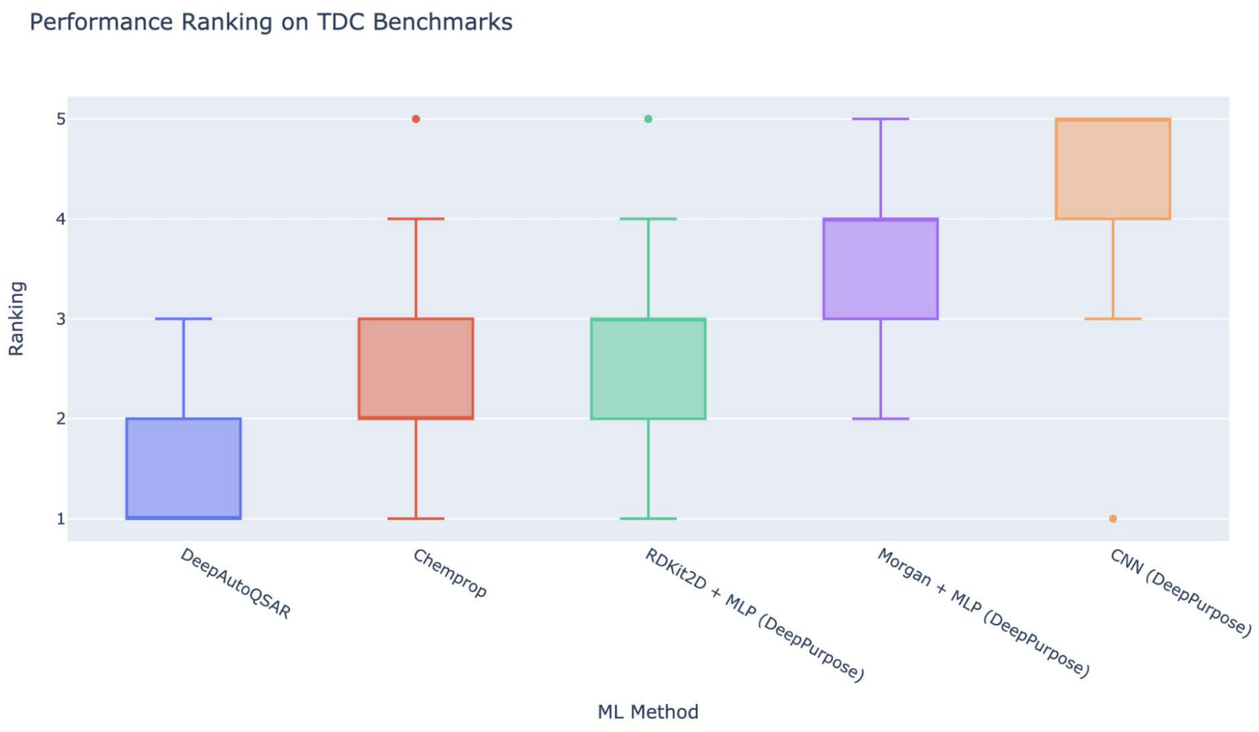
DILI

The best performing method on the drug induced liver injury dataset is DeepAutoQSAR with an area under the receiver operating characteristic curve (AUROC) of 0.933 ± 0.006. ChemProp performs second with an excellent AUROC of 0.918 ± 0.008. The worst performing approach on the DILI data is DeepPurpose (CNN) with an AUROC of 0.792 ± 0.016. The overall spread in the performance between methods is ~ 0.2, on a total scale of 0 to 1. The dataset is the smallest among the investigated sets with only 475 drugs.

Aggregate Results

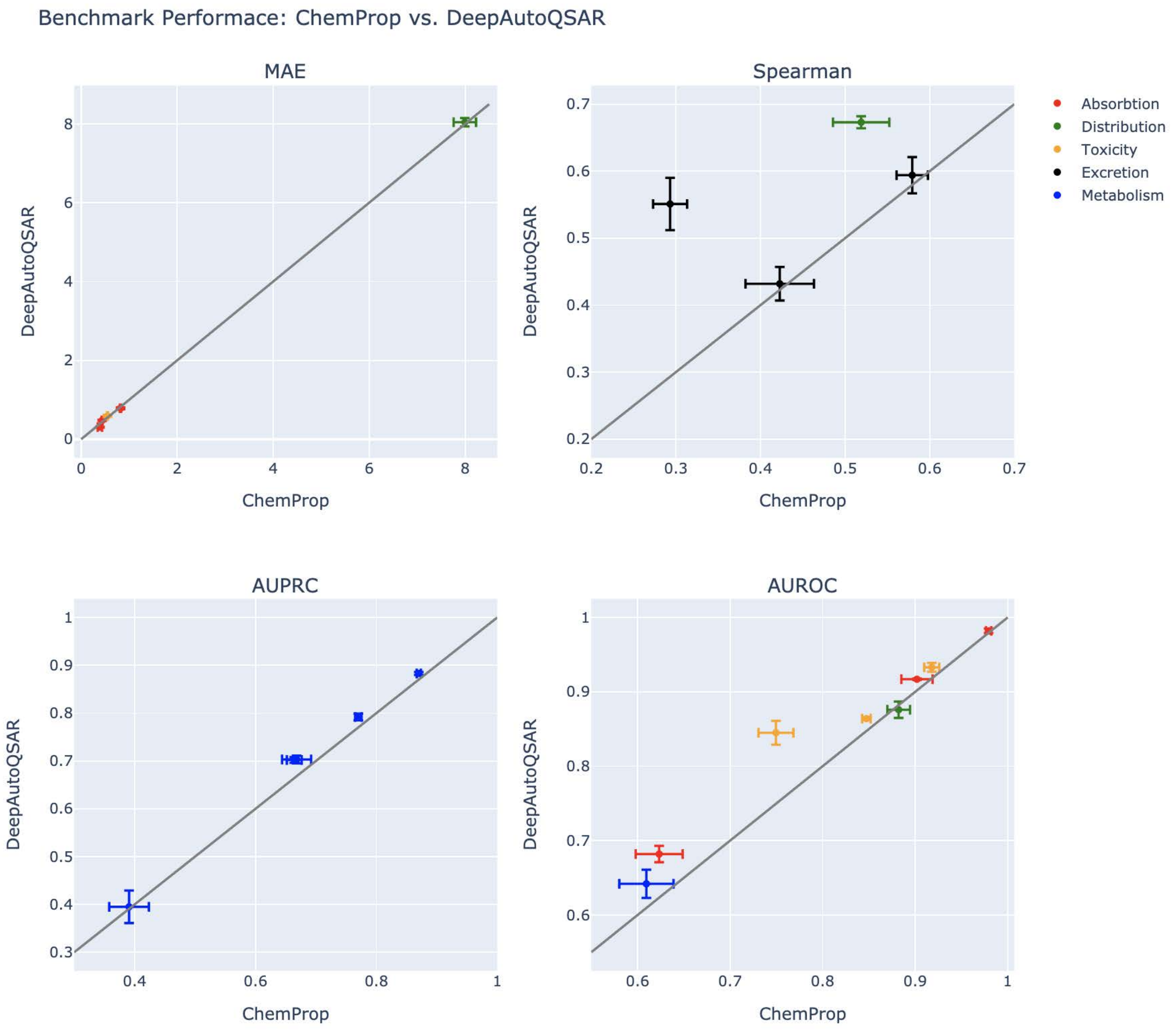
For a global view of relative performance across ML models, Figure 2 shows the distribution of ranking positions for each of the five models across all TDC endpoints. The plot gives the lower and upper (first and third) quartiles of the data distribution as the edges of the bounding boxes, with whiskers showing the lower and upper fences of the distributions. Outliers are also plotted as points. The rankings show DeepAutoQSAR performs the best on average, followed by ChemProp, DeepPurpose RDKit2D + MLP, DeepPurpose Morgan + MLP, and then DeepPurpose CNN. All methods except for DeepPurpose Morgan + MLP rank first in at least one task, and all methods except DeepAutoQSAR rank last in at least one task.

Figure 2: The boxplots above show the distributions of rankings for each method across the 18 ADME and 4 Tox TDC datasets. DeepAutoQSAR ranks first most often, followed by ChemProp and DeepPurpose methods.



To highlight the relative performances of ChemProp and DeepAutoQSAR across all of the TDC tasks, we provide comparison plots in Figure 3. These plots show direct comparisons between the methods on a per-metric basis; clockwise from top-left, we show performance by mean absolute error (MAE), Spearman correlation coefficient, area under the precision-recall curve (AUPRC), and area under the receiver-operator characteristic curve (AUROC). Individual datasets from the benchmark are color-coded corresponding to the associated ADMET classification, as is denoted in the above tables. Uncertainty via score standard deviation is also denoted via whiskers in the correlation plots. Note that subplots differ in scales and units. We provide additional comparison plots between ChemProp and DeepPurpose methods in the supplementary material.

Figure 3: Scatter plots show relative performance of DeepAutoQSAR and ChemProp on all endpoints with associated uncertainties. Top left shows performance on tasks measured by MAE where lower is better. Top right and bottom row plots show performance measured by Spearman Correlation, AUPRC, and AUROC where higher is better. Colors indicate the ADME/Tox designations of the datasets.



Conclusions

As the utilization rate and subsequent popularity of machine learning approaches in small molecule discovery rises, it is increasingly important to maintain an informed understanding of different approaches' accuracy and stability in predicting relevant molecular properties. To this end, ADME and toxicity assay prediction remains a fundamental challenge for adopting machine learning in drug discovery and development, and ML model performance on these tasks is a core factor in determining the technology's successful application. Therefore, we provide benchmarking figures to demonstrate the relative performance of multiple state-of-the-art methods for molecular property prediction on ADME/Tox tasks, showing how existing tools may faithfully model chemical data. We also illustrate the strengths of Schrödinger's offering DeepAutoQSAR by highlighting its improved modeling performance over the open source ML/AI tools ChemProp and DeepPurpose.

Considering these benchmark results, we aim to continually develop DeepAutoQSAR to further improve overall accuracy, generalization to new chemistry, and label efficiency to adapt to the demands of ML in real world drug discovery. We offer these results to the growing field of machine learning on molecules as a whole, as recognizing the abilities of existing methods can enable easier comparison for others to use in their development of ML tools. Ultimately, we hope that our work as a machine learning community will enable the discovery and deployment of new therapeutics through faster and more cost effective means with tangible benefit to patients.

Supplemental Materials

DeepAutoQSAR Model Hyper-parameters and References:

Table 9 contains a list of ML model architectures and associated hyper-parameters explored for model selection during DeepAutoQSAR model training. The models are a mix of classical ML models operating on vectorial data, like Random Forests and XGBoost, and more specialized graph convoluted models focused on representation learning for graph structured data. For each model, we provide a list of free hyper-parameters which we optimize using Bayesian Optimization; these parameters take float, integer, and boolean values, all of which are cast to continuous latent variables for BO with Gaussian Process Regressors.

We also give a reference link (or several links) for each model architecture. These links either directly describe the methods implemented in DeepAutoQSAR (eg. XGBoost, GIN), or contain information which was used indirectly for the construction of custom methods (eg. TorchGraphConv). In terms of implementation library, Random Forest models use sklearn's implementation[15], XGBoost comes from the XGBoost library [16], and all other models are implemented in PyTorch. The GNN's are all implemented with PyTorch Geometric [17]. For the full DeepAutoQSAR model search, we also perform BO over other components (data transformations, additional molecular features), but we omit those details.

Table 9: Architectural specifications for ML models eligible for inclusion in DeepAutoQSAR's predictive ensemble. We provide the name of the model architecture, the range of possible hyper-parameters searched over, and a link to literature describing each model.

ML Model	Model Hyper-parameters	Architecture Reference
Dense Neural Network	# hidden layers: 1-3 Hidden size 1: 64-1000 Hidden size 2: 64-256 Hidden size 3: 32-64 Dropout: 0.0-0.5 Epochs: 50-150 ECFP size: 500-2060 ECFP radius: 2-4	N/A
Random Forest Regressor	# estimators: 10-200 ECFP size: 500-1050 ECFP radius: 2-4	18
XGBoost	# estimators: 10-200 ECFP size: 500-1050 ECFP radius: 2-4	19
TorchGraphConv	# Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dense layer size: 128-256 Epochs: 50-150	20
GCN	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F	21
GraphSAGE	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F	22
GIN	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F Train GIN eps: T/F	23
TopK	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F TopK ratio: 0.1-0.6	24

SAGPool	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F SAGPool ratio: 0.1-0.6	25
EdgePool	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F	26
GlobalAttention	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F	23
Set2Set	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F	27
SortPool	#Conv layers: 2-3 Hidden size 1: 64-256 Hidden size 2: 64-128 Hidden size 3: 32-64 Dropout: 0.0-0.5 Dense layer size: 128-256 Epochs: 50-150 Jumping Knowledge: T/F Nodes held: 3-10	28

Figure 4: Comparison of per-dataset performance between DeepAutoQSAR and DeepPurpose Kit2D + MLP. See associated chart in main text for plot details.

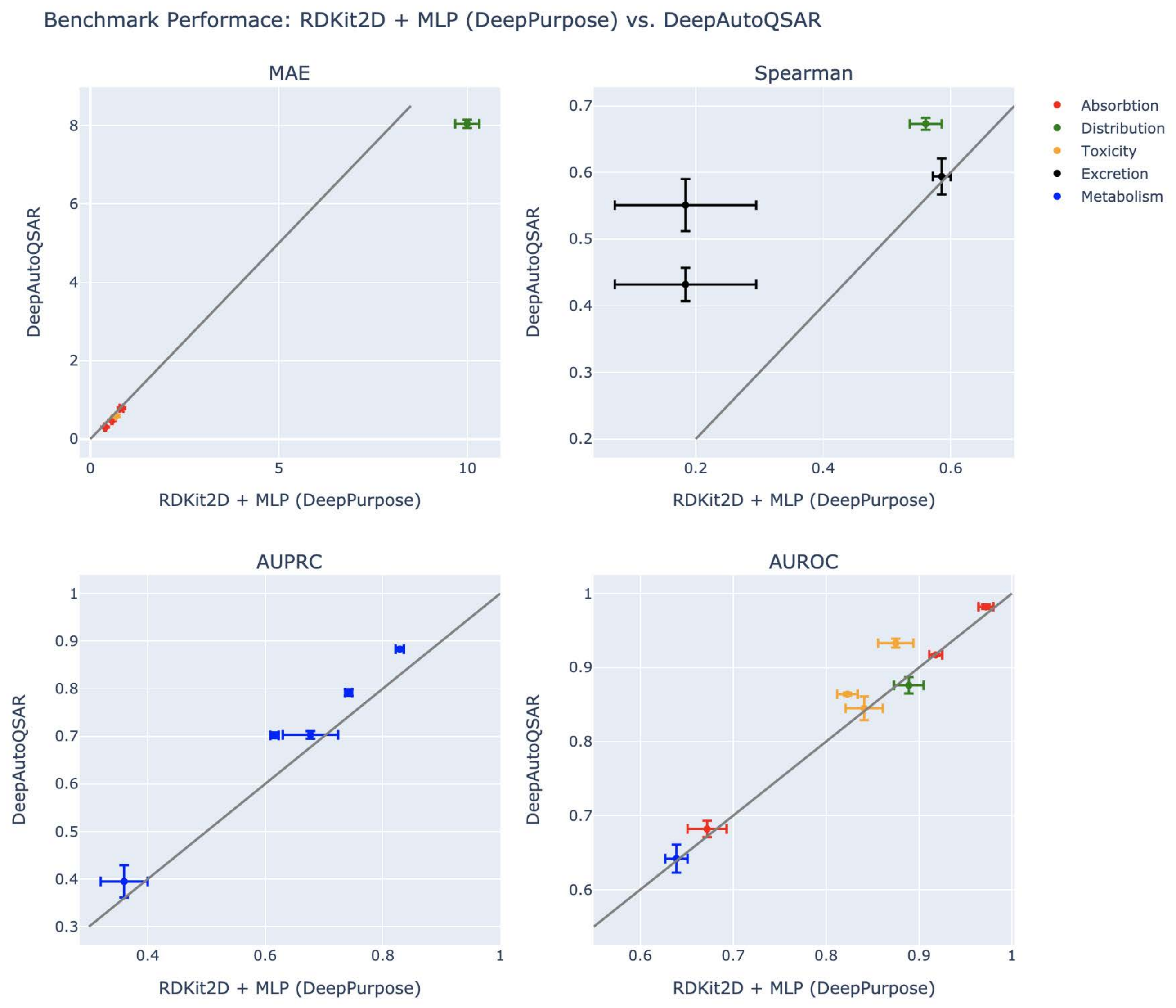


Figure 5: Comparison of per-dataset performance between DeepAutoQSAR and DeepPurpose CNN See associated chart in main text for plot details.

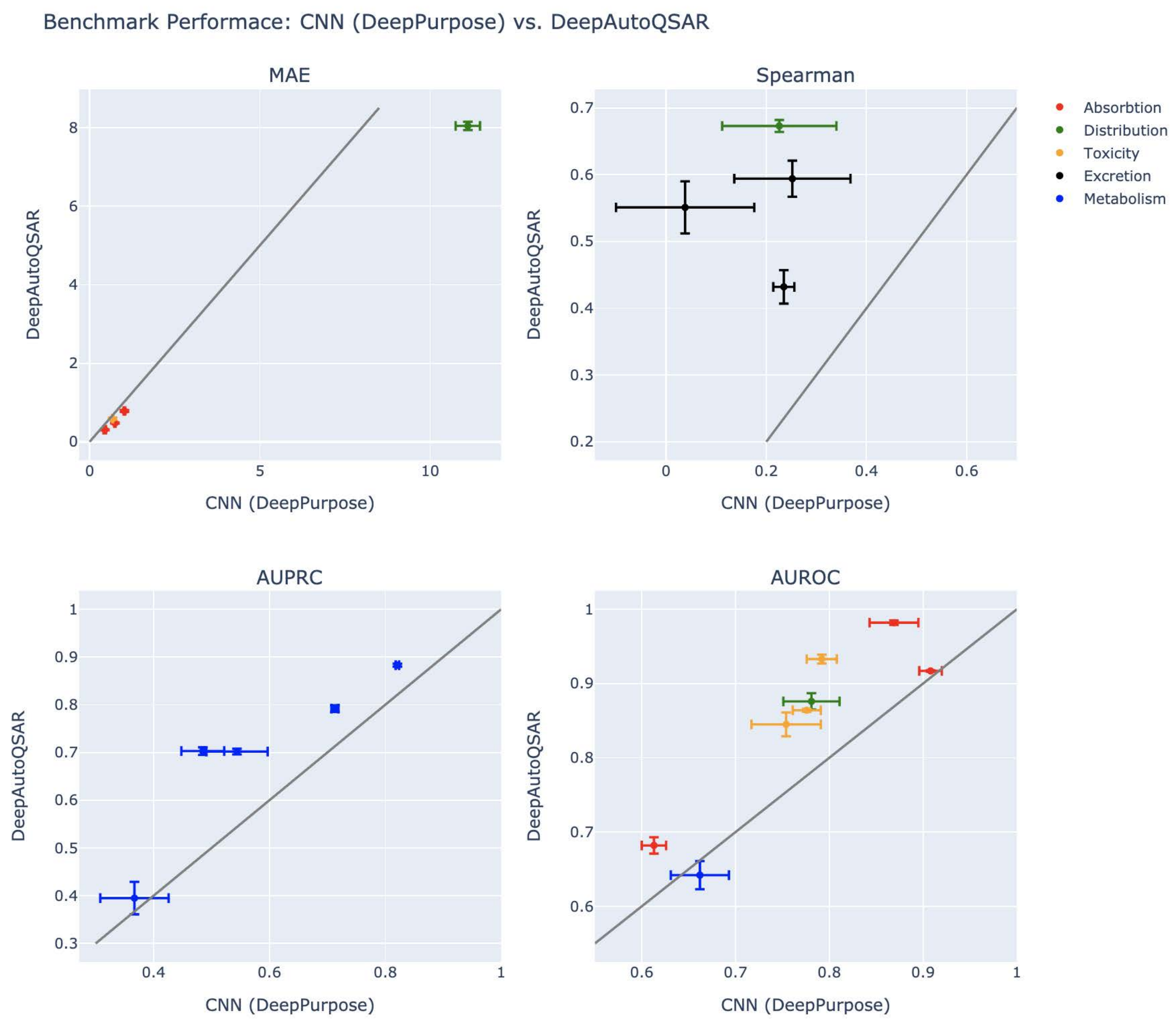
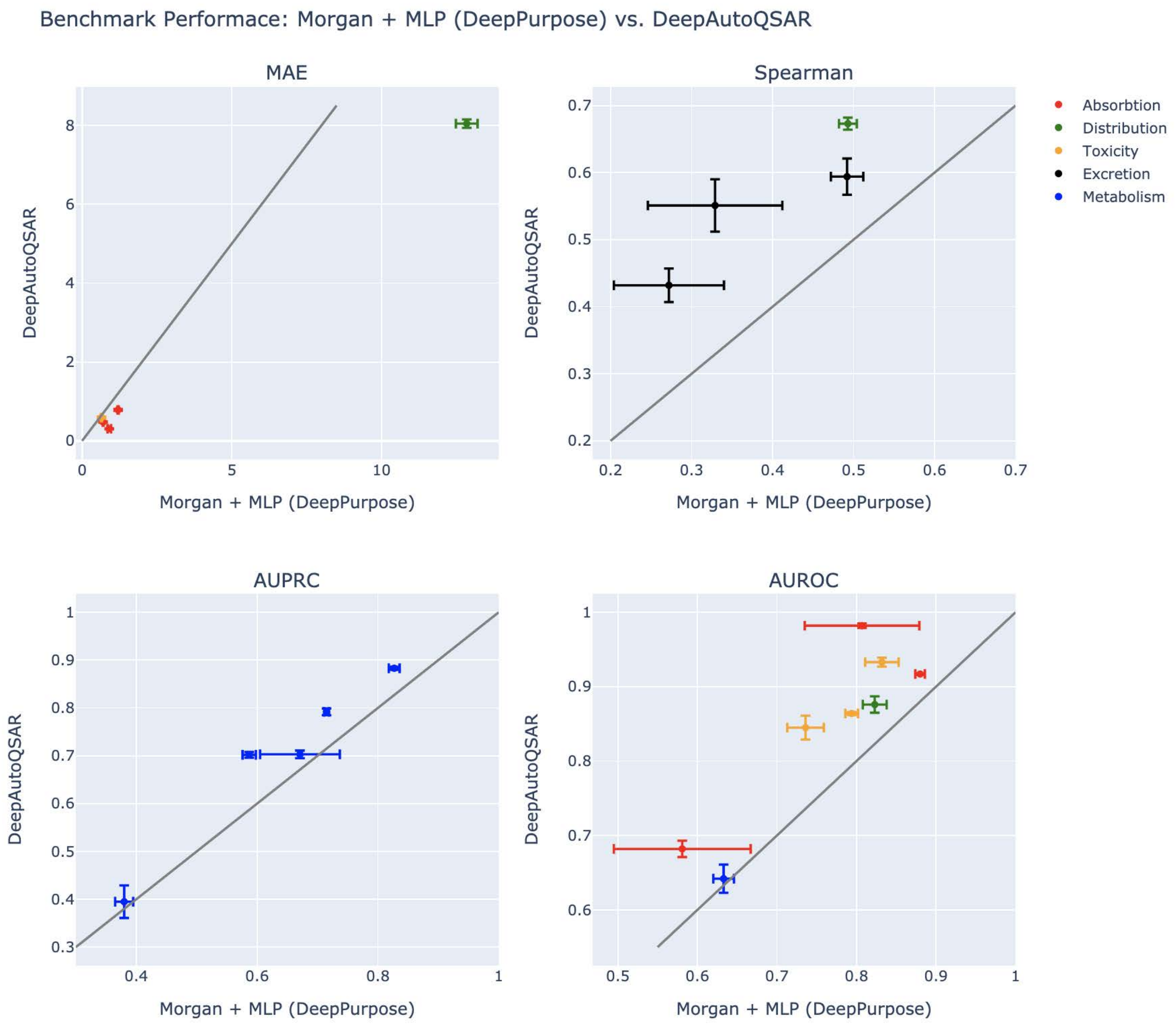


Figure 6: Comparison of per-dataset performance between DeepAutoQSAR and DeepPurpose Morgan + MLP. See associated chart in main text for plot details.



References

1. Göller, A. H.; Kuhnke, L.; Montanari, F.; Bonin, A.; Schneckener, S.; Ter Laak, A.; Wichard, J.; Lobell, M.; Hillisch, A. Bayer's in Silico ADMET Platform: A Journey of Machine Learning over the Past Two Decades. *Drug Discov. Today* **2020**, 25 (9), 1702–1709.

2. Huang, K.; Fu, T.; Gao, W.; Zhao, Y.; Roohani, Y.; Leskovec, J.; Coley, C. W.; Xiao, C.; Sun, J.; Zitnik, M. Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. *arXiv* [cs.LG], 2021.

3. Pejó, B.; Biczók, G. Quality Inference in Federated Learning with Secure Aggregation. *arXiv [cs.LG]*, 2020.

4. Sheridan, R. P. Time-Split Cross-Validation as a Method for Estimating the Goodness of Prospective Prediction. *J. Chem. Inf. Model.* **2013**, 53 (4), 783–790.

5. Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, 9 (2), 513–530.

6. ChEMBL Database https://www.ebi.ac.uk/chembl/ (accessed 2022 -01 -26).

7. Hutter, F.; Kotthoff, L.; Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges*; Springer International Publishing, 2019.

8. Kelley, B. Descriptastorus: *Descriptor Computation(chemistry) and (optional) Storage for Machine Learning*; Github.

9. Bayesian Optimization https://ax.dev/docs/bayesopt.html (accessed 2021 -12 -08).

10. Džeroski, S.; Ženko, B. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Mach. Learn.* **2004**, 54 (3), 255–273.

11. Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model*. **2019**, 59 (8), 3370–3388.

12. Huang, K.; Fu, T.; Glass, L. M.; Zitnik, M.; Xiao, C.; Sun, J. DeepPurpose: A Deep Learning Library for Drug-Target Interaction Prediction. *Bioinformatics* **2021**, 36 (22-23), 5545–5547.

13. ADMET Benchmark Group https://tdcommons.ai/benchmark/admet_group/overview/ (accessed 2021 -12 -08).

14. Wang, N.-N.; Dong, J.; Deng, Y.-H.; Zhu, M.-F.; Wen, M.; Yao, Z.-J.; Lu, A.-P.; Wang, J.-B.; Cao, D.-S. ADME Properties Evaluation in Drug Discovery: Prediction of Caco-2 Cell Permeability Using a Combination of NSGA-II and Boosting. *J. Chem. Inf. Model*. **2016**, 56 (4), 763–773.

15. 1.11. Ensemble methods https://scikit-learn.org/stable/modules/ensemble.html (accessed 2022 -01 -26).

16. Introduction to Boosted Trees — xgboost 1.6.0-dev documentation https://xgboost.readthedocs.io/en/latest/tutorials/model.html (accessed 2022 -01 -26).

17. PyG Documentation — pytorch_geometric 2.0.4 documentation https://pytorch-geometric.readthedocs.io/en/latest/ (accessed 2022 -01 -26).

18. 1.11. Ensemble methods https://scikit-learn.org/stable/modules/ensemble.html (accessed 2021 -12 -08).

19. Introduction to Boosted Trees — xgboost 1.6.0-dev documentation https://xgboost.readthedocs.io/en/latest/tutorials/model.html (accessed 2021 -12 -08).

20. Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *arXiv [cs.LG]*, 2015.

21. Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv [cs.LG]*, 2016.

22. Hamilton, W. L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs. *arXiv [cs.SI]*, 2017.

23. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful Are Graph Neural Networks? *arXiv [cs.LG]*, 2018.

24. Knyazev, B.; Taylor, G. W.; Amer, M. R. Understanding Attention and Generalization in Graph Neural Networks. *arXiv [cs.LG]*, 2019.

25. Lee, J.; Lee, I.; Kang, J. Self-Attention Graph Pooling. *arXiv [cs.LG]*, 2019.

26. Diehl, F. Edge Contraction Pooling for Graph Neural Networks. *arXiv [cs.LG]*, 2019.

27. Vinyals, O.; Bengio, S.; Kudlur, M. Order Matters: Sequence to Sequence for Sets. *arXiv [stat.ML]*, 2015.

28. Zhang, M.; Cui, Z.; Neumann, M.; Chen, Y. An End-to-End Deep Learning Architecture for Graph Classification. *AAAI* **2018**, 32 (1).

Authored by: Zach Kaplan, Stephan Ehrlich and Karl Leswing

Learn more about Schrodinger’s automated engine for machine learning, AutoQSAR at:
www.schrodinger.com/products/autoqsar

Contact Us: sales@schrodinger.com