Modeling Physico-Chemical ADMET Endpoints With Multitask Graph Convolutional Networks

Floriane Montanari, Lara Kuhnke, Antonius ter Laak, Djork-Arné Clevert

Submitted date: 01/08/2019 • Posted date: 02/08/2019
Licence: CC BY-NC-ND 4.0

Citation information: Montanari, Floriane; Kuhnke, Lara; ter Laak, Antonius; Clevert, Djork-Arné (2019): Modeling Physico-Chemical ADMET Endpoints With Multitask Graph Convolutional Networks. ChemRxiv. Preprint.

Simple physico-chemical properties like logD, solubility or serum albumin binding have a direct impact on the likelihood of success of compounds in clinical trials. Here, we collected all the Bayer in house data related to these properties and applied machine learning techniques to predict them for new compounds. We report that, for the endpoints studied here, a multitask graph convolutional network appears a highly competitive choice. The new model shows increased predictive performance on all endpoints compared to previous modeling methods.

File list (1)

MTNN_paper_ChemRxiv.pdf (870.96 KiB)
Modeling physico-chemical ADMET endpoints with multitask graph convolutional networks

Floriane Montanari¹, Lara Kuhnke¹, Antonius ter Laak¹, Djork-Arné Clevert¹

AUTHOR ADDRESS
¹ Digital Technologies, Bayer AG, Berlin, Germany

KEYWORDS: ADMET prediction, multitask learning, graph convolutional networks, solubility, QSAR.

ABSTRACT: Simple physico-chemical properties like logD, solubility or melting point can reveal a lot about how a compound under development might later behave. These data are typically measured for most compounds in drug discovery projects in a medium throughput fashion. Collecting and assembling all the data related to these properties allowed us to apply powerful machine learning techniques to predict the outcome of those assays for new compounds. In this paper, we report our finding that, especially for predicting physicochemical ADMET endpoints, a multitask graph convolutional approach appears a highly competitive choice. For seven endpoints of interest, we compared the performance of that approach to fully connected neural networks and different single task models. The new model shows increased predictive performance compared to previous modeling methods and will allow early prioritization of compounds even before they are synthesized. In addition, our model follows the generalized solubility equation without being explicitly trained under this constraint.

1. INTRODUCTION
Properties such as solubility, logD, or serum albumin binding have a direct impact on the likelihood of a compound to be successful in clinical trials [1] [2]. While measuring these endpoints can be done in a relatively high throughput fashion, it still requires the compounds to be synthesized. In silico prediction tools allow to rank and prioritize compounds before they are synthesized, limiting the amount of experiments performed and thereby saving time and money in drug discovery projects.

Machine learning approaches are typically used to map the structure of compounds to their properties, a method called quantitative structure-activity relationship (QSAR). Common algorithms include multiple linear regression, random forest or support vector machine in combination with circular fingerprints or molecular properties to describe the molecules [3] [4] [5] [6] [7]. Water solubility and melting point are two endpoints for which a lot of previous modeling was published [8] [9] [10] [11] [12] [4] [3] [5] [13].

Membrane affinity and human serum albumin (HSA) binding, on the other hand, are rarely the subject of QSAR publications. In 2002, Kratochwil and colleagues collected a dataset of 138 compounds with known HSA binding and built a partial least squares (PLS) model on top of similarity matrices obtained by comparing pharmacophoric features of the training set. The maximum cross-validation $R^2$ was reported at 0.48 [14]. Another study by Ghaforian and Amin [7] used a public dataset of 792 compounds and tried different QSAR methods (linear models, regression trees, random forest, etc.) and reported a boosted tree model with validation $R^2$ of 0.65 as best performer.

Since the Merck molecular activity challenge in 2012 (www.kaggle.com/c/MerckActivity), it became clear that Deep Learning can increase the performance of QSAR models [15] [16], at least when sufficient data is available.

The idea to learn jointly over multiple endpoints, or multitask learning, is neither new nor specific to the cheminformatics field. Multitask neural networks share parameters in (some of) their hidden layers between all tasks, forcing the learning of a joint representation of the input that will be useful to all tasks. The main advantages of multitask learning are i) a regularization effect, as the model has to use the same amount of parameters to learn more tasks ii) a transfer learning effect, whereby learning related tasks helps extracting features that are useful in a more general way iii) a dataset augmentation effect, as smaller tasks can be combined together with large tasks avoiding overfitting on the small task [17].
In 2014, Dahl et al. published the first deep multitask approach to classify the bioactivity of compounds in about 20 different assays. They observed an increased ROC AUC performance of 0.04 on average compared to random forest models, with improvements on individual tasks ranging from no improvement to up to +0.17 [15].

In 2016, Kearnes and colleagues [18] published a related work whereby 22 undisclosed ADMET endpoints from Vertex Pharmaceuticals were modeled together in a large multitask neural network. The endpoints had between a few hundred and a few thousand data points. The authors showed that single task neural networks would outperform random forest AUC performance by 0.05 on average, while combining all endpoints in one model increased the average baseline AUC performance by 0.1.

The previously mentioned studies used pre-established molecular descriptors to encode the chemical structures in a computer-readable way, such as circular fingerprints [19]. Recent progresses in handling graph data in neural networks [20] [21] [22] have been exploited in the cheminformatics field. In 2015, Duvenaud and colleagues proposed an algorithm to learn molecular fingerprints using convolutional networks on a graph representation of the compounds. In this paradigm, atoms correspond to nodes in the graph and bonds are the edges connecting the nodes. The features are learned at the node level, using the adjacency matrix of the graph to communicate information between neighboring nodes [23].

Concurrently to our work, Feinberg et al. published on their approach at Merck to model ADMET properties using a type of graph convolutional networks named PotentialNet [24]. They found great improvement over classical approaches using fingerprints and Random Forest models for many endpoints, including protein plasma binding, solubility and logD. The average absolute improvement in $R^2$ over 31 tasks was 0.14 in temporal splits. PotentialNet is a type of graph convolutional networks that have been designed to predict protein-ligand affinities based in gated graph neural networks. They distinguish different edge types and use a gated recurrent unit to learn information selectively [25].

In this paper, we combine ten different physico-chemical ADMET endpoints into a single multitask graph convolutional regression model. Our graph convolutional networks are simple in that they do not distinguish bond types and do not contain a recurrent unit, but rather follow the Duvenaud algorithm. We show how changing the learning paradigm (from regular machine learning methods to deep learning), the way to describe compounds (from traditional circular fingerprints to end-to-end learned features) and combining endpoints into one model help improve the performance for most of the endpoints.

We also show that adding helper tasks (three endpoints for which no good prediction is required) can help boost the performance on more difficult, smaller endpoints like solubility. Similarly, we show that for very easy and large tasks, combining them into a multitask model does not bring them further predictivity. All validations are performed in varied settings beyond random splits, mimicking real world use cases such as time splits or cluster splits.

2. METHODS

2.1 Dataset

In this work, we collected in-house data for the following ADMET endpoints: logD in neutral and acidic pH, solubility (various assay settings), melting point, membrane affinity and human serum albumin binding (Table 1).

Biological data corresponding to a given assay was pre-processed the following way: when the same compound is measured more than once for that assay, then the average of the measurements is taken as final experimental value. In case a measurement is preceded by an unequal sign ($<10 \mu M$ for example), we report either the double of the value (in case the qualifier is ‘>’) or half of the value (in case the qualifier is ‘<’). For human serum albumin binding and membrane affinity, the log$_{10}$ of the experimental value is taken. For melting point and logD, the values are taken as reported in the experiment. For solubility, the reported value in mg/L is first transformed to mol/L then a log$_{10}$ transformation is applied.

For the chemical data, we used the Standardize Molecule tool from Pipeline Pilot, selecting “Standardize Charges”, “Keep largest fragment” and “Clear stereo”. Canonical tautomers are generated, then molecules are standardized as neutral by deprotonating bases and protonating acids.

2.2 Model validation

- Data splits

Models were evaluated in both a cross-validation and a separate test set fashion. Different splitting strategies were applied. We clustered the compounds of the combined datasets using the k-means algorithm ($K=10$) and different versions of the ECFC6 fingerprints. Clusters not containing compounds of every task were merged into larger clusters. One cluster was chosen as a test set while the others served as the different folds for the cross-validation set up. Random splits were also performed where compounds would be assigned to a fold randomly, but keeping the folds of the same sizes as those obtained by the clustering procedure and ensuring that each fold contains representatives from each task.

Time splits could not be performed in a cross-validation fashion because we would have needed to find up to 10 measurement dates for which each endpoint would have enough data measured. Instead, we used one temporal split to separate training from test sets. We distinguish two different types of time splits: one where a measurement date is fixed and for each task independently, later measurements are taken as test sets while earlier measurements belong to the training set.
Table 1. ADMET datasets used to train the models.

| Endpoint                              | Code | # compounds | Data transformation | Helper task |
|---------------------------------------|------|-------------|---------------------|-------------|
| LogD (pH7.5)                          | LOD  | 76 548      | n.a.                | no          |
| LogD (pH2.3)                          | LOA  | 236 280     | n.a.                | no          |
| Membrane affinity                     | LOM  | 64 506      | log₁₀               | no          |
| Human serum albumin binding           | LOH  | 61 398      | log₁₀               | no          |
| Melting point                         | LMP  | 90 589      | n.a.                | no          |
| Solubility (DMSO)                     | LOO  | 38 841      | log₁₀(mol/L)        | no          |
| Solubility (powder)                   | LOP  | 2 334       | log₁₀(mol/L)        | no          |
| Solubility (nephelometry)             | LON  | 88 301      | log₁₀(mol/L)        | yes         |
| Solubility (DMSO not fully dissolved) | LOX  | 7 392       | log₁₀(mol/L)        | yes         |
| Solubility (no assay annotation)      | LOQ  | 50 016      | log₁₀(mol/L)        | yes         |

This is later referred to as taskwise time split as it ignores the fact that a compound measured earlier in one assay might be measured later in another. We also propose a strict time split where the training sets are further filtered to remove any compound that would occur in the test set of another task. For one task, logD at acidic pH (LOA), no test dates were available, and in this case compounds were split randomly even in the time split settings.

- Performance measures

The models predict continuous values. The performance of such regression models is evaluated by the coefficient of determination $r^2$ (which measures the concordance between predicted and experimental values) and the Spearman correlation coefficient $\rho$ (which measures the ranking capabilities of the models). In the case of cross-validation, the individual fold performances are averaged and reported.

2.3 Machine learning models

- Random forest:

Single task models were built using Random Forest regression as implemented in Pipeline Pilot v.18.1. The input features are extended connectivity fingerprint counts of diameter 6 (hereafter referred to as ECFC6) folded to 1024 or 2048 [19].

- Fully connected single task network:

For the fully connected neural networks, we used a pyramidal architecture with 4 hidden layers (of dimensions 2000, 1000, 500 and 100 respectively) and as input features we used ECFC6 fingerprint counts folded to 1024 or 2048. The activation function used in the hidden units was ReLU, following the observation from Zhou et al. that ReLU seems superior to sigmoid for regression tasks [26]. A decreasing amount of dropout was applied to each layer (50% in the first two hidden layers, 25% in the next, and none in the last hidden layer). Weights were initialized using He’s method [27]. Biases in the hidden layers were initialized to 0 and to -1 for the output layer.

Because fingerprints are typically sparse, using dropout on the input features would not have a lot of effect during training. Input noise is used instead to effectively randomly "drop in" chemical features at training time. For this, we generate positive integers by rounding and taking the absolute value from samples of a normal distribution with zero mean and standard deviation of 3 to mimic fingerprint counts. Then, the real inputs are replaced by these noisy fingerprints with a probability $p$ (in our experiments, $p=0.01$ or $p=0.02$ worked best). To smooth out the inputs, we apply the hyperbolic tangent ($\tanh$) function directly after the input noise step (Figure 1). The mean squared error is used as a loss function.

![Figure 1. Input feature preprocessing and architecture of the fully connected neural networks. When only one output unit exists, then we talk about single task neural networks (STNN).](image-url)
The models were implemented in Tensorflow version 1.2.1. Hyperparameters such as hidden layer dimensions, learning rate, weight decay, learning rate scheduling, etc. were optimized on the melting point endpoint and then applied to all other tasks.

- Fully connected multitask network:
For the multitask version, the same architecture (4 hidden layers of dimensions 2000, 1000, 500, 100 respectively, see Figure 1) as for the single task networks was used. The input noise probability was reduced to $p=0.01$ and the learning rate, batch size, learning rate scheduling, number of epochs and weight decay were adjusted using cross-validation in the training set.

To learn in a multitask fashion, the loss function corresponds to a weighted average of individual tasks' mean squared errors. This means that endpoints with different output ranges (for example, melting points Celsius degrees and can reach over 200) could potentially participate differently in the global loss. To avoid this problem, we scale each endpoint values to zero mean and unit standard deviation using standard scaling. Because of the unequal sizes of the different tasks, we explore different ways of averaging the individual task losses: in the “simple” setting, each task receives a weight of $1/N$ with $N$ being the number of tasks represented in the training minibatch. In the “balanced” setting, tasks with fewer examples are proportionally upweighted compared to tasks with more examples in the minibatch. Missing values (input examples without label for some of the tasks) are ignored and do not participate in the task’s individual losses. If a task does not have any training example in a minibatch then it is ignored and does not participate in the overall loss.

- Graph convolutions

Graph convolutional networks learn node features by propagating features from neighboring nodes and learning affine transformations that will help for the task at hand [22]. In case of molecules, the nodes are the atoms and the edges are the bonds of the molecular graph. A training example is a whole graph and the task is a classification or regression at the graph level. Here, we use the implementation of the Duvenaud algorithm [23] in DeepChem v.1.2.1 [28]. We keep the architecture and hyperparameters suggested by the authors for ADMET predictions, namely:

- 75 input atomic features (see SI1 for details)
- 2 graph convolution steps with a feature dimension of 128 each, with ReLU activation function
- a dense layer with 256 units and ReLU activation function

These operations lead to learned continuous atomic features of dimension 256. To make a prediction at the molecule level, the individual atom features have to be aggregated. For this, the feature values are averaged across atoms (mean feature) and the maximum value across atoms is also taken (max feature). These two representations are concatenated and a tanh activation function is applied to give rise to a final molecule representation of size 512.

The learning rate was set to 0.001, and a batch size of 128 was used. The models were trained for 40 epochs. Adam optimization was performed with learning rate decay every 1000 steps.

The same architecture and hyperparameters were used in the multitask setting. Endpoint values were standardized with zero mean and unit standard deviation like in the fully-connected multitask counterpart. The loss is a task-weighted MSE.

3. RESULTS AND DISCUSSION

3.1 Datasets sizes and overlaps

Table 1 reports the different datasets used in this study. The smallest dataset is solubility measured from powder material, while the largest is logD at acidic pH. Figure 2 reports the pairwise correlations between datasets using shared compounds between endpoints. As expected, all solubility endpoints are correlated between them. LogD in acid and neutral pH are also correlated. Human serum albumin binding (LOH) is correlated with solubility, while logD is anticorrelated with solubility. Melting point (LMP) is not very strongly correlated with any other endpoints in this study. Value distributions of the ten endpoints can be found in the supplementary materials (SI2).

![Figure 2. Pearson’s correlation coefficients between pairs of endpoints. When less than 25 compounds were measured in both members of the pairs, no correlation is reported. Endpoint codes are listed in Table 1.](image)

In total, the datasets together contain 537,443 unique compounds, of which about 79% occur only in one end-
point, 11% are shared between two endpoints, 9% are shared between three and 1% between four or more. The pairs of endpoints with most overlapping compounds are membrane affinity (LOM) with human serum albumin binding (LOH), LOM with the solubility without assay annotation (LOQ), LOM with the nephelometric solubility assay (LON), logD (LOD) with DMSO solubility (LOO), LON with LOH and LOQ with LOH.

3.2 Performance of single task models
Three types of single task models were built: Random Forest (RF) and fully-connected, feed-forward neural networks (STNN), as well as graph convolutional networks (GCNN). Both RF and STNN are built upon circular fingerprints, while graph convolutional networks learn their feature representation in an end-to-end fashion, starting from the molecular graph and 75 simple atomic descriptors as initial node features.

Table 2 shows the leave-cluster-out cross-validation performance for the different endpoints in different modeling situations. The first three columns correspond to the single task case, where a model is built for each endpoint independently from the others. For all tasks except melting point and solubility from powder, fully connected neural networks greatly outperform random forest. On average, \( R^2 \) is improved by 0.06 and Spearman’s rho is improved by 0.05 in the cluster cross-validation setting. These improvements are in line with previous observations [15], [18]. The endpoints that are best modeled by the STNN are the two logD, membrane affinity and solubility from nephelometry. These are large tasks (between 64 000 and 230 000 datapoints, Table 1). Solubility from powder and from DMSO not fully dissolved and melting point are the less well predicted endpoints. The two solubility endpoints have the least data and the low performance in a cluster split setting can be explained by overfitting, but melting point is actually one of the largest tasks with 90 000 training examples, so the reason for the poorer performance might have to be found somewhere else. Melting point is notoriously difficult to predict [29] even though the experimental data is very accurate.

Switching from a fixed compound representation (circular fingerprints) to learnt features (graph convolutional networks) allows us to further gain predictive performance in many cases. On average, \( R^2 \) is improved by 0.06 (over STNN) and by 0.12 (over RF) while Spearman’s rho is improved by 0.05 (over STNN) and by 0.29 (over RF) in the cluster cross-validation setting. The only task for which graph convolutional features seem to be detrimental is solubility from powder (LOP) which shows a drop in \( R^2 \). In random split cross-validation though, the performance of the graph convolutional network for LOP is on par with the ones of RF and STNN (see Supplementary information S3). We therefore assume that the training of graph convolutional networks tends to overfit on smaller training sets. This would explain why the performance in random split appears high (compounds in the test splits are likely similar to compounds in the training set, so the learnt features work well also for the test data) while the performance in cluster splits drops significantly (cluster split cross-validation shows performance in chemical areas that are far away from the training set, where the learnt overfitted features generalize poorly).

It is worth mentioning that intensive hyperparameter selection was not necessary in our case. For Random Forest with ECFC6 fingerprints, we used the default settings from Pipeline Pilot, which is nowadays one of the go-to methods for QSAR models in Computational Molecular Design at Bayer. For STNN, a few pyramidal architectures were tested, dropout and input noise were included for controlling overfitting, and the details of batch size, learning rate, etc. were tuned on a cross-validation split for the task melting point only and applied to all other endpoints. As can be seen from Table 2, those parameters seem to perform well on the other ADMET datasets tested, which is something already observed by Ma and colleagues [16].

Note that our final settings follow the guidance provided by the authors: most of our endpoints are logtransformed, we use 4 hidden layers of decreasing sizes with decreasing amounts of dropout and ReLU as activation function. Two main differences are our usage of input noise followed by a tanh transformation to counteract the fact that dropout at input is not appropriate for our sparse fingerprint data and the choice of a bias of -1 for the output layer which we found experimentally to improve performance. In another study, Zhou et al. evaluated different parameters and architectures for single task models for industrial ADME endpoints and found that a pyramidal architecture, dropout and weight decay were beneficial, that models built with ReLU were less sensitive to other hyperparameters, and that regression tasks require smaller learning rates than classification tasks [26]. The graph convolutional STNN settings were also taken as recommended in DeepChem and not further optimized due to the lengthy training process, but the good performance of the trained models shows here again a practical robustness to adjustable parameters.

3.3 Performance in multitask setting
Since many of the endpoints of interest have some biological relations and actual correlations (Figure 1), we hypothesized that learning all the tasks together would bring further performance improvement. Indeed, by learning simultaneously several tasks, the model has to learn feature representations that are useful for all tasks (regularization aspect) and smaller tasks will benefit from the chemical space coverage of the larger tasks.

We built fully connected multitask networks and graph convolutional multitask networks (Table 2, last two columns). In the fully connected version (MTNN), the task that sees most improvement is the small solubility from powder endpoint (LOP, 0.29 increase in \( R^2 \) and 0.16 increase in Spearman’s rho). Most of the larger tasks are...
Table 2. Performance of different learning algorithm in the ten ADMET endpoints. We report the average of cluster split cross-validation folds (not used for parameter tuning) as well as standard deviations of those folds. The best performing method is given in bold (as well as those for which standard deviations overlap).

|                | Random Forest | STNN<sup>a</sup> | STNN graph conv<sup>b</sup> | MTNN<sup>c</sup> | MTNN graph conv<sup>d</sup> |
|----------------|---------------|-------------------|---------------------------|-----------------|-----------------------------|
| LOD<sup>e</sup> | 0.63 ± 0.03   | 0.78 ± 0.02       | 0.87 ± 0.02               | 0.75 ± 0.01     | 0.88 ± 0.01                 |
| LOA<sup>f</sup> | 0.49 ± 0.01   | 0.72 ± 0.02       | 0.94 ± 0.02               | 0.64 ± 0.01     | 0.91 ± 0.00                 |
| LOM<sup>g</sup> | 0.43 ± 0.01   | 0.53 ± 0.02       | 0.68 ± 0.02               | 0.51 ± 0.02     | 0.71 ± 0.02                 |
| LOH<sup>h</sup> | 0.39 ± 0.01   | 0.49 ± 0.02       | 0.56 ± 0.02               | 0.49 ± 0.02     | 0.65 ± 0.02                 |
| LMP<sup>i</sup> | 0.39 ± 0.01   | 0.31 ± 0.02       | 0.51 ± 0.02               | 0.35 ± 0.02     | 0.51 ± 0.02                 |
| LOO<sup>j</sup> | 0.43 ± 0.01   | 0.47 ± 0.02       | 0.47 ± 0.02               | 0.49 ± 0.02     | 0.59 ± 0.02                 |
| LOP<sup>k</sup> | 0.09 ± 0.01   | 0.03 ± 0.02       | -0.17 ± 0.02              | 0.32 ± 0.02     | 0.56 ± 0.02                 |
| LON<sup>l</sup> | 0.50 ± 0.01   | 0.53 ± 0.02       | 0.59 ± 0.02               | 0.54 ± 0.02     | 0.68 ± 0.02                 |
| LOX<sup>m</sup> | 0.33 ± 0.01   | 0.37 ± 0.02       | 0.33 ± 0.02               | 0.48 ± 0.02     | 0.58 ± 0.02                 |
| LOQ<sup>n</sup> | 0.46 ± 0.01   | 0.51 ± 0.02       | 0.58 ± 0.02               | 0.53 ± 0.02     | 0.69 ± 0.02                 |

<sup>a</sup> single task neural network, <sup>b</sup> single task graph convolutional network, <sup>c</sup> multitask neural network, <sup>d</sup> multitask graph convolutional network, <sup>e</sup> logD, <sup>f</sup> logD in acidic pH, <sup>g</sup> membrane affinity, <sup>h</sup> human serum albumin binding, <sup>i</sup> melting point, <sup>j</sup> solubility from DMSO, <sup>k</sup> solubility from powder, <sup>l</sup> solubility from nephelometry, <sup>m</sup> solubility from DMSO not fully dissolved, <sup>n</sup> solubility no assay information.

either not affected or show poorer performance in multitask than in single task approach.

This confirm previous observations that larger tasks are negatively affected by joint training [16]. On the other hand, all solubility endpoints get better predicted. This can be explained by the high correlation between the different solubility assays (Figure 1). The best MTNN model used balanced task weighting when calculating the loss, meaning that tasks with large amount of training data would see their loss down weighted with respect to less represented tasks. One consequence is that the model is allowed to make more errors in the larger tasks, a phenomenon that can be seen when looking at the performance of our largest endpoint, logD in acidic pH (LOA). This could explain the lower performance in MTNN for this particular task (0.08 decrease in $R^2$ and 0.03 decrease in Spearman's rho).

We saw in the single task approach that graph convolutional networks showed higher performance than fully-connected networks, and the same is true in the multitask learning approach: on average, $R^2$ increased by 0.17 with respect to the non-convolutional network and Spearman's rho by 0.09. Comparing the single task with the multitask convolutional networks leads to similar observations as when comparing STNN with MTNN in the non-convolutional setting. The average improvement in performance is 0.14 in $R^2$ and 0.06 in Spearman's rho, but the endpoint-by-endpoint picture is more nuanced. Endpoints like logD (LOD) or melting point (LOP) show no change in performance, while the acidic logD (LOA, our largest task) is negatively impacted in the multitask setting. The tasks benefiting the most from the joint training are the two smaller solubility endpoints (solubility from powder LOP and solubility from DMSO not fully dissolved, LOX). We also notice that the standard deviations of both reported metrics are the smallest for the multitask graph convolutional model, meaning that learning is very stable even across potentially very different cross-validation folds (we report in Table 2 cluster cross-validation results, which in practice contains folds of unequal sizes and difficulty).

3.4 Effect of helper tasks

Not all endpoints under consideration here are of interest to medicinal chemists, our end users. The nephelometric assay (LON) is not in use anymore and therefore the
training set contains only historical data. The solubility from DMSO not fully dissolved (LOX) probably contains a lot of artefacts. And finally, the other solubility assay where no assay information was recorded (LOQ) also contains mostly historical data (see Table 1). It means that the actual performance of the models on those three datasets is of little importance, but we included them for completeness and because, in joint training, they might help train the other solubility endpoints (LOO and LOP). We compared the effect of including or not those three helper tasks into the multitask graph convolutional model.

From Table 3, one observes that the endpoints’ performances stay stable without the helper tasks. The two solubility endpoints, which we would assume to benefit most from the helper tasks (recall that these are different solubility assays), show indeed a slightly lower performance in the absence of helper tasks (-0.02 $R^2$ for the DMSO solubility and -0.02 Spearman’s rho for the powder solubility). We deduce that adding the helper tasks is not detrimental to the proper learning of the model but will help reaching more accurate predictions in solubility. Beyond considerations on the performance level, one can also argue that adding more related endpoints will also enrich the chemical space covered by the training set, helping the graph convolutions to learn meaningful atom representations and increasing the generalization capability of the model.

Table 3. Performance of the multitask graph convolutional model without helper tasks. Average of cluster split cross-validation folds. In parenthesis, difference with the results from the multitask graph convolutional model in Table 2.

|        | $R^2$     | Spearman |
|--------|-----------|----------|
| LOD$^a$| 0.87 (-0.01) | 0.94     |
| LOA$^b$| 0.92 (+0.01) | 0.96     |
| LOM$^c$| 0.71       | 0.84     |
| LOH$^d$| 0.65       | 0.83 (+0.01) |
| LMP$^e$| 0.52 (+0.01) | 0.73     |
| LOO$^f$| 0.57 (-0.02) | 0.76 (-0.01) |
| LOP$^g$| 0.56       | 0.74 (-0.02) |

$^a$ logD, $^b$ logD in acidic pH, $^c$ membrane affinity, $^d$ human serum albumin binding, $^e$ melting point, $^f$ solubility from DMSO, $^g$ solubility from powder

3.5 General Solubility Equation

The aqueous solubility of a small molecule is linked to its melting point and octanol-water partition coefficient by the general solubility equation (GSE) [30]:

$$\log S_w = -0.01 \times (LMP - 25) - \log(K_{ow}) + 0.5$$

where $\log S_w$ is the logarithm of base 10 of the aqueous solubility in mol/L, LMP is the melting point in Celsius degrees and $K_{ow}$ is the partition coefficient.

We applied this formula to our dataset. In total, 105 compounds had measurements for all three endpoints LOO, LMP and LOD. The Pearson correlation between the predicted $\log S_w$ using the GSE with the original 105 LOO data points is 0.75.

We compared this correlation with the one obtained on the cluster split test set by our multitask graph convolutional network. On almost 4000 LOO data points not seen by the model (and in a different chemical space than the training set), the Pearson correlation coefficient between predictions and measurements is 0.81. We also used the model to predict melting point and LogD for these 4000 test set datapoints and see whether the model predictions also follow the GSE. For this, we used the predictions of the model for melting point and LogD, obtained the aqueous solubilities according to the Yalkowsky equation and compared these with the predicted LOO. The Pearson correlation is here 0.83, meaning that our model follows globally the GSE model of aqueous solubility without actually being trained on that constraint. Correlations plots can be found in supplementary material SL5. We saw from the endpoints correlation matrix (Fig. 2) that LogD is clearly anti-correlated with solubility. To check that the GSE property of our network is not simply due to the correlation of logD and solubility in the training data, we also computed the Pearson correlation between the predicted logD and the predicted solubility for the 4000 test datapoints: this correlation is 0.71, a clear drop in magnitude with respect to the correlation when taking into account both the predicted logD and melting point and following the GSE formula.

3.6 Performance in time splits

All the results previously commented were obtained by clustering the compounds by structure, then validating the models on left-out clusters of compounds (leave-cluster-out cross-validation). This type of validation shows how well the model generalizes and performs on unseen chemical space. Another way to evaluate models in an industrial setting is to apply time splits. In this approach, all measured data up to a given date are used for training while all recent data are used as a separate test set. We retrained our MTNN graph convolutional model on such a historical subset of our assays, and used all data measured after June 2014 as test set. Table 4 reports the obtained performance on the test sets. Note that, for LOA, no test date could be retrieved so the split is random.

The number of data points for each endpoint vary, as some assays are not often used anymore (melting point, membrane affinity) while others are intensively requested in the course of ongoing drug discovery projects.
In terms of performance, we observe slightly lower values in the time split than in the average of the leave-cluster-out cross-validations ($R^2$ dropped by 0.06 and Spearman’s rho by 0.03). Still, the performance of our multitask model is solid also in this prospective type of validation. A similar table in the supplementary information shows the results for the strict time splits (see Methods section for more details), where a compound measured in several endpoints can only occur either in training or in test for all endpoints (SI4). From the relatively robust performance in leave-cluster-out and prospective time split validation, we conclude that a weekly retraining of the model to aggregate newly measured data points is not mandatory in the production phase of the model.

3.7 Experimental error: are we nearly there?

An ideal model would predict the data to an accuracy similar to the experimental accuracy. To evaluate how far from ideal performance our model actually is, we retrieved all measurements available for i) a same compound or ii) a pair of stereoisomers with only one stereo-center) for each of the assays of interest. Experimental error is then reported as an average standard deviation of those multiple measurements. Table 5 compares the experimental error thus computed with the root mean square error (RMSE) of the model for each of the endpoints.

It is clear that our best model, although way improved over the ones previously used in house, is still far from the optimal accuracy for all of the endpoints.

We note that our Bayer experimental variability is very low for solubility compared to previously reported experimental errors of up to 0.6 log units [31]. Our RMSE for predicting solubility is around 0.8, which is in line with observations from Boobier on different public solubility datasets using a multilayer perceptron. Human experts were asked to predict solubility for 25 compounds and the best participant had an RMSE of 0.94 [32]. We deduce from this that our model for predicting solubility is probably as good or better than a human expert, and present the advantage of being fast, inexpensive and immune to tiredness.

Table 5. Experimental error (average standard deviation of multiple measurements) compared to average prediction error (in RMSE) in cluster split cross-validation. In parenthesis, the number of pairs of compounds for which multiple measurements were available.

|                | RMSE final model | Experimental error (stereoisomers) | Experimental error (multiple measurements) |
|----------------|------------------|-----------------------------------|-------------------------------------------|
| LOD$^a$        | 0.34             | 0.06 (#112)                       | 0.14 (#299)                               |
| LOA$^b$        | 0.36             | 0.04 (#91)                        | 0.28 (#300)                               |
| LOM$^c$        | 0.51             | 0.11 (#176)                       | 0.18 (#801)                               |
| LOH$^d$        | 0.50             | 0.12 (#172)                       | 0.17 (#764)                               |
| LMP$^e$        | 0.39             | 0.61 (#80)                        | 16.9 (#1037)                              |
| LOO$^f$        | 0.82             | 0.17 (#199)                       | 0.35 (#63)                                |
| LOP$^g$        | 0.79             | 0.28 (#14)                        | 0.14 (#15)                                |

$^a$ logD, $^b$ logD in acidic pH (random split), $^c$ membrane affinity, $^d$ human serum albumin binding, $^e$ melting point, $^f$ solubility from DMSO, $^g$ solubility from powder

4. CONCLUSIONS

In this work, we built a predictive model for seven ADMET assays corresponding to endpoints of high interest: logD, solubility, melting point, membrane affinity, and human serum albumin binding. Combining all the data available, we were able to apply Deep Learning methods to learn to predict these endpoints. We showed that, as previously observed, neural networks generally outperform Random Forest in the case of large physicochemical datasets, and that joint training approaches bring further performance improvements at least for the smaller endpoints. Moving away from classical compound representations, we showed that graph convolutional networks are a very powerful method that seems particularly suited for more “physico-chemical” endpoints. The best model, a multitask graph convolutional model with 3 additional helper tasks, showed very robust performance both in cluster splits and temporal splits. This does not mean that ADMET modeling is a solved problem, since in our experience graph convolutional approaches did not work as well for more complex endpoints like Caco2 permeation or in vitro metabolic stability (validations not shown). Also, multitask modeling is still pretty much a trial-and-error type of work, where it is not clear beforehand which tasks should be combined together nor which hyperparameters would work for a particular task combination.
ASSOCIATED CONTENT

We provide the following supporting information:
- Details on atomic features used as input to the graph convolutional networks
- Distribution of training data
- Performance of the methods in random cross-validation
- Performance of the multitask graph convolutional model in strict time splits
- Correlations between solubility and calculated solubility using the GSE

This material is available free of charge via the Internet at http://pubs.acs.org.

AUTHOR INFORMATION

Corresponding Author
* floriane.montanari@bayer.com

Author Contributions
The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

ACKNOWLEDGMENT

DC and FM acknowledge funding from the Bayer AG Life Science Collaboration (“DeepMinDS”).

ABBREVIATIONS

ADMET, Absorption, Distribution, Metabolism, Excretion, Toxicity; QSAR, quantitative structure-activity relationship; AUC, area under the curve; ROC receiver operating characteristic; DMSO, dimethyl sulfoxide; ECFC6, extended connectivity fingerprint counts of diameter 6; STNN, single task neural network; GCNN, graph convolutional neural network; RF, random forest; MTNN, multitask neural network; ReLU, rectified linear unit; tanh, hyperbolic tangent; GSE, General Solubility Equation; LOA, logD in acidic pH; LOD, logD; LOM, membrane affinity; LOH, human serum albumin binding; LMP, melting point; LOO, solubility from powder; LON, solubility from nephelometry measurement; LOX, solubility not fully dissolved; LOQ, solubility no assay information; RMSE, root mean squared error.

REFERENCES

[1] M. J. Waring, J. Arrowsmith, A. R. Leach, P. D. Leeson, S. Mandrell, R. M. Owen, G. Paireaudou, W. D. Pennie, S. D. Pickett, J. Wang, O. Wallace und A. Weir, „An analysis of the attrition of drug candidates from four major pharmaceutical companies,“ Nature Reviews Drug Discovery, Bd. 14, Nr. 7, pp. 475-486, 19.7 2015.
[2] M. P. Gleeson, A. Hersey, D. Montanari und J. Overington, „Probing the links between in vitro potency, ADMET and physicochemical parameters,“ Nature Reviews Drug Discovery, Bd. 10, Nr. 3, pp. 197-208, 1.3 2011.
[3] Q. Zang, K. Mansouri, A. J. Williams, R. S. Judson, D. G. Allen, W. M. Casey und N. C. Kleinstreuer, „In Silico Prediction of Physicochemical Properties of Environmental Chemicals Using Molecular Fingerprints and Machine Learning,“ Journal of Chemical Information and Modeling, Bd. 57, Nr. 1, pp. 36-49, 23.1 2017.
[4] M. Watkins, N. Sizochenko, B. Rasulev und J. Leszcynski, „Estimation of melting points of large set of persistent organic pollutants utilizing QSPR approach,“ Journal of Molecular Modeling, Bd. 22, Nr. 3, p. 53, 13.3 2016.
[5] I. V. Tetko, D. M. Lowe und A. J. Williams, „The development of models to predict melting and pyrolysis point data associated with several hundred thousand compounds mined from PATENTS,“ Journal of Cheminformatics, Bd. 8, Nr. 1, p. 2, 22.12 2016.
[6] B. Bhattachary, W. Teetz, T. Liu, T. Øberg, N. Jeliakova, N. Kochev, O. Pukalov, I. V. Tetko, S. Kovarich, E. Papa und P. Gramatica, „CADASTER QSPR Models for Predictions of Melting and Boiling Points of Perfluorinated Chemicals,“ Molecular Informatics, Bd. 30, Nr. 2-3, pp. 189-204, 14.3 2011.
[7] T. Ghafoorian und Z. Amin, „QSAR models for the prediction of plasma protein binding,“ BiomImpacts : BI, Bd. 3, Nr. 1, pp. 21-7, 2013.
[8] T. Cheng, Q. Li, Y. Wang und S. H. Bryant, „Binary Classification of Aqueous Solubility Using Support Vector Machines with Reduction and Recombination Feature Selection,“ Journal of Chemical Information and Modeling, Bd. 51, Nr. 2, pp. 229-236, 28.2 2011.
[9] S. E. Fioretti, D. E. Baceco, C. Rojas, J. F. Aranda und P. R. Duchowicz, „Conformation-independent quantitative structure-property relationships study on water solubility of pesticides,“ Ecotoxicology and Environmental Safety, Bd. 171, pp. 47-53, 30.4 2019.
[10] H. Sun, P. Shah, K. Nguyen, K. R. Yu, E. Kerns, M. Kabir, Y. Wang und X. Xu, „Predictive models of aqueous solubility of organic compounds built on A large dataset of high integrity,“ Bioorganic & Medicinal Chemistry, 27 5 2019.
[11] C. A. Bergström und P. Larsson, „Computational prediction of drug solubility in water-based systems: Qualitative and quantitative approaches used in the current drug discovery and development setting,“ International Journal of Pharmaceutics, Bd. 540, Nr. 1-2, pp. 185-193, 5 4 2018.
[12] F. Nigsch, A. Bender, B. van Buuren, J. Tissen, E. Nigsch und J. B. O. Mitchell, „Melting Point Prediction Employing k-Nearest Neighbor Algorithms and Genetic Parameter Optimization,“ 2006.
[13] S. Chinta und R. Rengaswamy, „Machine Learning Derived Quantitative Structure Property Relationship (QSPR) to Predict Drug Solubility in Binary Solvent Systems,“ Industrial & Engineering Chemistry Research, Bd. 58, Nr. 8, pp. 3082-3092, 27.2 2019.
[14] N. A. Kratochwil, W. Huber, F. Müller, M. Kansy und P. R. Gerber, „Predicting plasma protein binding of drugs: a new approach,“ Biochemical pharmacology, Bd. 64, Nr. 9, pp. 1355-74, 11 2002.
[15] G. E. Dahl, N. Jaityl und R. Salakhutdinov, „Multi-task Neural Networks for QSAR Predictions,“ 4 6 2014.
[16] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl und V. Svetnik, „Deep Neural Nets as a Method for Quantitative Structure-Activity Relationships,“ Journal of Chemical Information and Modeling, Bd. 55, Nr. 2, pp. 263-274, 23.2 2015.
[17] R. Caruana, „Multitask Learning,“ *Machine Learning*, Bd. 28, Nr. 1, pp. 41-75, 1997.
[18] S. Kearnes, B. Goldman und V. Pande, „Modeling Industrial ADMET Data with Multitask Networks,“ 28 6 2016.
[19] D. Rogers und M. Hahn, „Extended-Connectivity Fingerprints,“ *Journal of Chemical Information and Modeling*, Bd. 50, Nr. 5, pp. 742-754, 24 5 2010.
[20] J. Bruna, W. Zaremba, A. Szlam und Y. LeCun, „Spectral Networks and Locally Connected Networks on Graphs,“ 20 12 2013.
[21] M. Henaff, J. Bruna und Y. LeCun, „Deep Convolutional Networks on Graph-Structured Data,“ 16 6 2015.
[22] T. N. Kipf und M. Welling, „Semi-Supervised Classification with Graph Convolutional Networks,“ 9 9 2016.
[23] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik und R. P. Adams, „Convolutional Networks on Graphs for Learning Molecular Fingerprints,“ 30 9 2015.
[24] E. N. Feinberg, R. Sheridan, E. Joshi, V. S. Pande und A. C. Cheng, „Step Change Improvement in ADMET Prediction with PotentialNet Deep Featureization,“ 28 3 2019.
[25] E. N. Feinberg, D. Sur, Z. Wu, B. E. Husic, H. Mai, Y. Li, S. Sun, J. Yang, B. Ramsundar und V. S. Pande, „PotentialNet for Molecular Property Prediction,“ *ACS Central Science*, Bd. 4, Nr. 11, pp. 1520-1530, 28 11 2018.
[26] Y. Zhou, S. Cahya, S. A. Combs, C. A. Nicolaou, J. Wang, P. V. Desai und J. Shen, „Exploring Tunable Hyperparameters for Deep Neural Networks with Industrial ADME Data Sets,“ *Journal of Chemical Information and Modeling*, Bd. 59, Nr. 3, pp. 1005-1016, 25 3 2019.
[27] K. He, X. Zhang, S. Ren und J. Sun, „Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,“ 6 2 2015.
[28] B. Ramsundar, P. Eastman, P. Walters, V. Pande, K. Leswing und Z. Wu, Deep Learning for the Life Sciences, O'Reilly Media, 2019.
[29] L. D. Hughes, D. S. Palmer, F. Nigsch und J. B. O. Mitchell, „Why Are Some Properties More Difficult To Predict than Others? A Study of QSAR Models of Solubility, Melting Point, and Log P,“ 2008.
[30] N. Jain und S. H. Yalkowsky, „Estimation of the aqueous solubility I: application to organic nonelectrolytes,“ *Journal of Pharmaceutical Sciences*, Bd. 90, Nr. 2, pp. 234-52, 2 2001.
[31] D. S. Palmer und J. B. O. Mitchell, „Is Experimental Data Quality the Limiting Factor in Predicting the Aqueous Solubility of Druglike Molecules?，“ *Molecular Pharmaceutics*, Bd. 11, Nr. 8, pp. 2962-2972, 4 8 2014.
[32] S. Boobier, A. Osbourn und J. B. O. Mitchell, „Can human experts predict solubility better than computers?，“ *Journal of Cheminformatics*, Bd. 9, Nr. 1, p. 63, 13 12 2017.
Supplementary information to “Modeling ADMET data with multitask graph convolutional networks”

SI1: Input atomic features for the graph convolutional models

The following 75 features are encoded for each atom in the molecules:
- Atomic symbol as one-hot encoding from 44 possible choices
- Degree as one-hot encoding from 11 possible choices (0 to 10)
- Total number of hydrogens as one-hot encoding from 5 possible choices (0 to 4)
- Implicit valence as one-hot encoding from 7 possible choices (0 to 6)
- Formal charge
- Number of radical electrons
- Hybridization as one-hot encoding from 5 possible choices (SP, SP2, SP3, SP3D, SP3D2)
- Whether or not the atom is aromatic

SI2: Distribution of experimental values for the ADMET endpoints of interest

Membrane affinity, hSA binding and the solubility endpoints are log-transformed.

SI3: Performance of the different models in random split cross-validation

|       | Random Forest | STNN\(^a\) | STNN graph conv\(^b\) | MTNN\(^c\) | MTNN graph conv\(^d\) |
|-------|---------------|------------|------------------------|------------|------------------------|
|       | \(R^2\)       | Spearman   | \(R^2\)                | Spearman   | \(R^2\)                | Spearman   |
| LOD\(^e\) | 0.81          | 0.91       | 0.88                   | 0.94       | 0.84                   | 0.93       | 0.91          | 0.96         |
| LOAf     | 0.79          | 0.90       | 0.86                   | 0.94       | 0.80                   | 0.92       | 0.91          | 0.96         |
| LOM\(^g\) | 0.68          | 0.83       | 0.71                   | 0.85       | 0.69                   | 0.85       | 0.70          | 0.84         |
| LOH\(^h\) | 0.65          | 0.82       | 0.67                   | 0.84       | 0.67                   | 0.84       | 0.62          | 0.83         |
| LMP\(^i\) | 0.54          | 0.73       | 0.44                   | 0.75       | 0.49                   | 0.74       | 0.53          | 0.74         |
| LOO\(^j\) | 0.63          | 0.80       | 0.65                   | 0.82       | 0.67                   | 0.82       | 0.68          | 0.84         |
SI4: Performance of the multitask graph convolutional model in the strict time split test set.

|      | R²  | Spearman | Test set size |
|------|-----|----------|---------------|
| LODa | 0.86| 0.93     | 23 164        |
| LOAb | 0.90| 0.95     | 47 250        |
| LOMc | 0.62| 0.80     | 199           |
| LOHd | 0.56| 0.74     | 646           |
| LMPe | 0.50| 0.72     | 63 318        |
| LOQf | 0.62| 0.80     | 8 068         |
| LOPg | 0.50| 0.73     | 584           |

*a* logD, *b* logD in acidic pH, *c* membrane affinity, *d* human serum albumin binding, *e* melting point, *f* solubility from DMSO, *g* solubility from powder

SI5: Correlations between solubility in the data, solubility as deduced from the General Solubility Equation (GSE) and solubility predicted by the model. A. Correlation between the measured solubility in DMSO and the calculated solubility according to GSE for compounds having all necessary measurements (LogD, melting point and solubility). B. Correlations between predictions of the multitask graph convolutional model for solubility and calculated solubility according to GSE using the melting point and logD predicted by the model.
