Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning

Yao Zhang and Alpha A. Lee
Cavendish Laboratory, University of Cambridge, Cambridge CB3 0HE, United Kingdom

Predicting bioactivity and physical properties of small molecules is a central challenge in drug discovery. Deep learning is becoming the method of choice but studies to date focus on mean accuracy as the main metric. However, to replace costly and mission-critical experiments by models, a high mean accuracy is not enough: Outliers can derail a discovery campaign, thus models need reliably predict when it will fail, even when the training data is biased; experiments are expensive, thus models need to be data-efficient and suggest informative training sets using active learning. We show that uncertainty quantification and active learning can be achieved by Bayesian semi-supervised graph convolutional neural networks. The Bayesian approach estimates uncertainty in a statistically principled way through sampling from the posterior distribution. Semi-supervised learning disentangles representation learning and regression, keeping uncertainty estimates accurate in the low data limit and allowing the model to start active learning from a small initial pool of training data. Our study highlights the promise of Bayesian deep learning for chemistry.

INTRODUCTION

Predicting physiological properties and bioactivity from molecular structure – quantitative structure-property relationships (QSPR) – underpins a large class of problems in drug discovery. Classical QSPR workflows separate descriptor generation – mapping a 2D or 3D molecular structure into a vector of real numbers using some handcrafted rules – and machine learning method that connects descriptors to property. Pioneering advances in machine learning such as graph neural networks directly take a molecular graph as input and infer the optimal structure-to-descriptor map from data, outperforming classical machine learning methodologies with handcrafted descriptors.

Nonetheless, graph neural networks are usually developed using frequentist maximum likelihood inference, with the benchmark being the mean error on a test set. However, if the goal of QSPR is to replace mission-critical but expensive experiments, a low mean error is insufficient: the user needs to have an estimate of uncertainty and know when the model is expected to fail. This is because typically only a small number of top-ranked predictions are selected to test experimentally, thus outliers can ruin a discovery campaign. Moreover, cost limits the number of experiments that can be run, thus an approach that judiciously designs the training set to maximise information gained is needed.

Uncertainty quantification, or domain applicability, has been extensively considered in the QSPR literature but not in the context of graph neural networks and not in a statistically complete way. Previous works estimate uncertainty of prediction as the distance in descriptor space between the input molecule and the training set, or training an ensemble of models and evaluating the variance. More recent works consider conformal regression, which trains two models, one for the molecular property and one for the error. However, there are two sources of uncertainty: Epistemic uncertainty arises due to insufficient data in the region of chemical space that the model is asked to make predictions on. Aleatoric uncertainty arises due to noise in the measurements themselves (e.g. noisy biochemical assays). Distance to the training set and variance within a model ensemble approximately capture epistemic uncertainty, whilst employing an ancillary model for prediction error approximately captures aleatoric uncertainty. We will show that the Bayesian statistical framework captures both sources of uncertainty in a unified and statistically principled manner.

Active learning strategies have been considered in the drug discovery literature. However, those pioneering works considered a priori defined molecular descriptors, and estimate uncertainty via variance within an ensemble of models. Notwithstanding the shortcomings with incomplete modelling of uncertainty discussed above, employing graph neural networks in active learning presents unique opportunities and challenges: High model accuracy in the big data limit comes at the cost of being data-hungry. As the descriptor is fully data-driven, the model cannot estimate how “far” a compound is from the test set in the low-data limit, leading to poor uncertainty estimate and breaking down the active learning cycle. Low-data drug discovery has been considered in the context of one-shot learning, which estimates distance in chemical space by pulling data from related tasks. Nonetheless, this approach requires a priori knowledge on which tasks are related. Works on generative molecular design overcome this problem by starting the active learning cycle with < 1000 quantitative measurements, which impose a significant upfront experimental cost.

In this paper, we combine Bayesian statistics – a principled framework for uncertainty estimation – with semi-supervised learning which learns the representation from unlabelled data. We show that Bayesian semi-supervised
graph convolutional neural networks can robustly estimate uncertainty even in the low data limit and drive an active learning cycle, and overcome dataset bias in the training set. Further, we demonstrate that the quality of posterior sampling is directly related to accuracy of the uncertainty estimates. As different Bayesian inference methods can be mixed and matched with different models, our study opens up a new dimension in the design space of uncertainty-calibrated QSPR models.

**METHODS AND DATA**

A machine learning method has two independent components: model and inference. The model is a function with parameters that relate the input to the output. Inference pertains to the methodology by which the model parameters are inferred from data. In terms of model, we focus on graph convolutional neural network models that take molecular graphs as input. In terms of inference, we focus on the Bayesian methodology.

**Supervised graph convolutional neural network**

Our baseline model is the graph convolutional fingerprint model \[22\]. The salient idea is the message passing operation \[22\], which creates a vector that summarises the local atomic environment around each atom while respecting invariance with respect to atom relabelling. A molecule is described by a graph, where the nodes are atoms and the edges are bonds. Atom \(i\) is described by a vector of atomic properties \(x_i\), and a bond connecting \(i\) and \(j\) is described by bond properties \(e_{ij}\). The algorithm is iterative: at step \(t\), each atom has a hidden state \(h^t_i\), which depends on “messages” \(m^t_v\) received from surrounding atoms as well as \(h^{t-1}_v\). The hidden states can be interpreted as descriptors of local atomic environment, and the messages allow adjacent atoms to comprehend the environment of its neighbours. Each atom is initialised to its atomic features, \(h^0_v = x_v\), and

\[
m^t_v = \sum_{w \in \mathcal{N}(v)} (h^{t-1}_w, e_{vw}),
\]

and

\[
h^t_v = \sigma(H_{t-1}^{\text{deg}(v)} m^t_v),
\]

where \(\mathcal{N}(v)\) denotes the set of atoms bonded to atom \(v\), \(\sigma(\cdot)\) is the sigmoid function, \(H_{t-1}^{\text{deg}(v)}\) is a learned matrix for each step \(t\) and vertex degree \(N\). The algorithm is run \(T\) times, with \(T\) being a hyperparameter. In the final step, the output is given by a multilayer neural network \(f(\cdot)\) that takes a weighted average of the hidden states at each step as input return a prediction,

\[
y = f \left( \sum_{v,t} \text{softmax}(W_t h^t_v) \right)
\]

where \(W_t\) are learned readout matrices, one for each step \(t\).

We use the implementation reported in the repository \[23\]. In all experiments, we consider \(T = 3\), hidden layer at each level has 128 units, fingerprint length 256 (i.e. \(W_t \in \mathbb{R}^{128 \times 256}\)), and \(f(\cdot)\) is a two layer neural network with 128 units each and \(\text{relu}\) units.

**Semi-supervised graph convolutional neural network**

The fully supervised approach learns molecular descriptors directly from data. This is an advantage if one has a lot of data but a disadvantage in the data-limited settings such as active learning applications, where the objective is to design informative experiments starting from a small pool of initial training data.

The insight behind the semi-supervised approach is that significant amount of chemical knowledge is contained within the molecular structures themselves, without any associated molecular properties (i.e. unlabelled data). Thermodynamic stability puts constrains on what bonds are possible, and tends to put certain bonds near each other, forming persistent chemical motifs. For example, just by looking at drug molecules, one would immediately spot ubiquitous motifs such as amide group, benzene rings etc., and some motifs often occur together as scaffolds \[24\]. The key assumption is that those persistent chemical motifs contribute to the molecular property that we want to predict. We can make mathematical progress by constructing a descriptor akin to Equation \[11 - 3\]. However, the objective is no longer trying to fit a particular property. Rather, the hidden states \(h^t_v\), which summarises the atomic environment around atom \(v\) within radius \(t\), are constructed such that they are predictable from the the hidden states of the surrounding atoms. Therefore, the model learns a descriptor that clusters similar environments.

Specifically, we use the semi-supervised approach developed by Nguyen et al. \[25\], which builds on the paragraph vector approach in natural language processing \[26\]. Given a set of molecular structures \(\mathcal{M}\), the hidden states \(h^t_v\) maximise the log-likelihood

\[
L = \sum_{m \in \mathcal{M}} \sum_{v \in m} \sum_{t=1}^T \log P(h^t_v | u_m),
\]

\[
P(h^t_v | u_m) = \frac{\exp((h^t_v)^T u_m)}{\sum_{n \in \mathcal{M}} \exp((h^t_v)^T u_n)},
\]

where
Bayesian deep learning

In Bayesian inference, the aim is to determine the distribution of model parameters that conforms to the data, the so-called posterior distribution. Let $\theta$ be model parameters, $x_i$ the dependent variables and $y_i$ the independent variable, such that

$$y_i = F(x_i, \theta) + \epsilon_i.$$  

(6)

where $\epsilon_i \sim N(0, \sigma_i^2)$ is the measurement noise. Bayes theorem states the posterior distribution, $P(\theta|\{x_i\}, \{y_i\})$, is related to the likelihood, $P(\{y_i\}|\theta, \{x_i\})$ and the prior $P(\theta)$ via

$$P(\theta|\{x_i\}, \{y_i\}) = \frac{1}{Z} P(\{y_i\}|\theta, \{x_i\})P(\theta),$$  

(7)

where $Z$ is a normalising constant. The prediction for an unknown input $\hat{x}$ is obtained by averaging over the posterior

$$\langle \hat{y} \rangle = \int P(\theta|\{x_i\}, \{y_i\})F(\hat{x}, \theta) \, d\theta.$$  

(8)

The uncertainty of model predictions can be readily derived from this Bayesian formalism. There are two types of uncertainties [17]. First, the epistemic uncertainty, is given by the variance of the prediction with respect to the posterior

$$\text{var}(\hat{y}) = \int P(\theta|\{x_i\}, \{y_i\})(\langle \hat{y} \rangle - F(\hat{x}, \theta))^2 \, d\theta.$$  

(9)

Second, the aleatoric uncertainty, is the intrinsic noise of the measurement $\sigma_i^2$. This aleatoric noise can depend on the input, $\sigma_i^2 = \sigma(x)^2$, as certain areas of the chemical space can be intrinsically more variable.

We note that the log posterior is, up to a constant,

$$- \log P(\theta|\{x_i\}, \{y_i\}) = \sum_i \left( \frac{1}{2\sigma_i^2}(y_i - F(\hat{x}, \theta))^2 + \frac{1}{2} \log \sigma_i^2 \right) - \log P(\theta),$$  

(10)

which is exactly the mean-squared loss if $\sigma_i$ is constant, with $\log P(\theta)$ being the regulariser. Therefore, maximum likelihood inference is a special case of Bayesian inference.

The Bayesian formalism is easy to state but computationally expensive. The numerical bottleneck is the numerical evaluation of the high dimensional integrals [26]. A plethora of approximate numerical methods have been developed in the literature to overcome this bottleneck. However, there is no free lunch, and methods which approximate the posterior well are usually computationally expensive. In this paper, we will consider two approximate methods spanning the cost-accuracy spectrum.

### Dropout variational inference

Variational inference seeks to approximate the posterior distribution by a distribution that is much easier to sample from. Refs [17, 28] show that a popular way to regularise neural networks – dropout – is equivalent to approximate Bayesian inference. The algorithm is simple: The neural network is forked at the last layer to have two outputs, the predicted aleatoric uncertainty $\sigma_i^2$ and dependent variable $y_i$, and trained to minimise the loss [10]. However, during training, each unit has a probability $p$ of being set to 0.

For a neural network with $M$ units, refs [17, 28] show that the above algorithm is approximately equal to finding parameters $\theta = (\Theta_1, \Theta_2 \cdots \Theta_M)$ that fit the distribution

$$q(\theta) = \prod_{m=1}^{M} \Theta_m z_m, \ z_m \sim \text{Bernoulli}(p),$$  

(11)

to the posterior distribution $P(\theta|\{x_i\}, \{y_i\})$, where $\Theta_m$ is the parameter vector associated with the $m^{th}$ unit.

Distribution [11], although not the same as the true posterior distribution, is significantly easier to sample: In the prediction phase, the model is run $N$ times, and akin to the training phase each unit has probability $p$ of being set to 0. The final prediction and uncertainty is taken to be the mean over $N$ different predictions of depending variable and variance, $\{y^i, (\sigma^i)^2\}_{i=1}^{N}$,

$$\langle y \rangle = \frac{1}{N} \sum_{m=1}^{N} y^m, \ \text{var}(y) = \frac{1}{N} \sum_{m=1}^{N} (y^m - \langle y \rangle)^2 + \frac{1}{N} \sum_{m=1}^{N} (\sigma^m)^2.$$  

(12)

In our numerical experiments, dropout is applied to every unit that is trained using supervised learning, i.e.
every unit in the supervised graph convolutional neural network is subjected to dropout, whereas for the semi-supervised case the layers on top of the hidden states are trained with dropout.

**Stein Variational Gradient Descent**

Rather than fitting a distribution to the posterior, Stein Variational Gradient Descent [29] directly draws samples from the posterior via gradient descent. Specifically, let \( \{ \theta_i \}_{i=1}^N \) be parameters randomly and independently initialised in parameter space. We want to evolve parameters such that, after \( T \) steps, \( \{ \theta_i^T \}_{i=1}^N \) are \( N \) independent samples drawn from \( P(\theta_j | \{ x_i \}, \{ y_i \}) \). Ref [29] shows that the following dynamical system does the trick:

\[
\theta_i^{t+1} = \theta_i^t + \eta \phi(\theta_i^t),
\]

where

\[
\phi(\theta) = \frac{1}{N} \sum_{j=1}^N \left[ k(\theta_j, \theta) \nabla_{\theta_j} \log P(\theta_j | \{ x_i \}, \{ y_i \}) + \nabla_{\theta_j} k(\theta_j, \theta) \right],
\]

and \( k(\cdot, \cdot) \) is a generic kernel function and \( \eta \) is the learning rate. Equations (13) - (14) can be interpreted as free energy minimisation of an interacting particle system: a “particle” (parameter vector) is subjected to a “force” \( \phi(\theta) \), which drives particles to regions of low energy (low loss), whilst forcing the particles apart to maximise entropy.

The key advantage of Equation (13) - (14) is that it is a well-defined approximation: Frequentist inference (c.f. Equation (10)) is recovered if \( N = 1 \), whereas when \( N \to \infty \) the system exactly samples from the posterior. Therefore, for finite \( N \), the algorithm interpolates between frequentist and full Bayesian inference. The computational cost and memory demands increase with \( N \), and in this paper we use \( N = 50 \).

**Datasets**

We consider a set of common regression benchmarks for physical properties prediction and bioactivity prediction. The MeltingPoint dataset is a collection of 3041 melting point measurements of drug-like molecules used in a benchmark study [30]. The ESOL dataset is a set of 1128 measured aqueous solubilities of organic molecules [31], and the FreeSolv dataset is a set of 642 hydration free energy measurements small molecules in water [32]. The ESOL and FreeSolv datasets are used in the MoleculeNet benchmark [10]. The CatS dataset comprises log IC\(_{50}\) measurements of 595 molecules against Cathepsin S, taken from the D3R Grand Challenge 3 and 4 [33]. The Malaria dataset is a set of in vitro half-maximal effective concentration (log EC\(_{50}\)) measurement of 10,000 molecules against a sulfide-resistant strain of P. falciparum, the parasite that causes malaria [34] used in benchmark [1]. The p450 dataset is a dataset of half-maximal effective concentration measurements of 11,082 molecules against Cytochrome P450 3A4, a key enzyme for metabolism and clearance of xenobiotics, taken from the PubChem assay AID 1851.

**RESULTS**

**Uncertainty quantification**

We first consider how well can the model estimate its own uncertainty given the full dataset, split into training (80%) and test (20%) sets. The quality of the uncertainty estimate is operationalised by asking what is the model accuracy when the most uncertain predictions are removed, with uncertainty quantified by the variance computed from Equation (12). Our baseline method is graph convolution with dropout, which has recently been implemented in the DeepChem package as a feature [10], although to our knowledge this is the first study that benchmarks Bayesian graph convolutional neural networks in terms of uncertainty quantification.

Figure 1 shows that semi-supervised learning with Stein Variational Gradient Descent accurately estimates uncertainty and significantly outperforms the baseline on every dataset. The plots show how the error varies as a function of confidence percentile – i.e. what is the error if we only consider the top-n % of compounds ranked by confidence; the shaded region is one standard deviation, estimated by analysing 5 random partitions of the data into training and test sets. In every case, the error is a decreasing function of model confidence, thus the model successfully estimates which predictions are likely to be correct and which predictions are outliers.

Another metric that we can evaluate is the shape of the confidence-error curve. For ESOL and FreeSolv, the error is a steeply decreasing at the low confidence limit before plateauing, suggesting that most predictions are accurate but for a few outliers, which the Bayesian method can identify. The situation is different for MeltingPoint, Malaria and p450 – the error is slowly decreasing at the low confidence limit before sharply decreasing when it approaches the 100% confidence percentile limit (see also insets of Figure 1). This suggests that a few predictions are very accurate, and Bayesian method can pick out those accurate predictions amid many less accurate ones. We note that our Bayesian model is well-suited for virtual screening applications, where the challenge is ensuring that the top-ranked actives picked out by the model are indeed actives, since only a very small proportion of the compounds ranked will actually be screened.
FIG. 1. Bayesian semi-supervised learning accurately predicts molecular properties and uncertainty. The plots show the model accuracy as a function of confidence percentile. The inset highlights the fact that predictions which the model are confident about are indeed accurate.

experimentally (the "early recognition problem") \cite{35,36}. A lingering question whether the quality of the uncertainty estimate is due to a set of good descriptors (obtained via semi-supervised learning) or accuracy of the Bayesian methodology. Figure 1 also shows that replacing Stein Variational Gradient Descent with dropout significantly reduces model performance. At the same confidence percentile, Stein Variational Gradient Descent consistently outperforms dropout. This suggests that the quality of posterior sampling drastically impacts the quality of uncertainty estimation.

Previous works on domain applicability have focused on either building an auxiliary model to predict the error \cite{15,16}, or estimating the uncertainty of a prediction via the distance of the input to the training set \cite{11,12,14}. The former models aleatoric uncertainty whereas the latter approximately captures epistemic uncertainty. Our Bayesian method captures both sources of uncertainty in a statistically principled manner. However, our model also provides independent estimates of epistemic and aleatoric uncertainties. As such, we can ask the question: is knowing epistemic or aleatoric uncertainty alone sufficient to estimate whether a prediction is accurate?

Figure 2 shows that the confidence-error curve for semi-supervised learning with Stein Variational Gradient Descent obtained by considering both epistemic and aleatoric uncertainty is below epistemic or aleatoric uncertainty alone. In other words, epistemic uncertainty and aleatoric uncertainty are distinct sources of uncertainty that contribute to prediction error, and neglecting either one of them leads to inferior model performance. This is particularly pronounced for \textit{ESOL}, \textit{CatS} and \textit{p450}, but also apparent in \textit{MeltingPoint} and \textit{Malaria}: the different uncertainties are the same within error for \textit{FreeSolv}. Moreover, there is no consistent trend as to whether epistemic or aleatoric uncertainty is more important – for \textit{ESOL}, epistemic uncertainty is a better estimate of error than aleatoric uncertainty, whereas the opposite is true for \textit{p450} and \textit{CatS}.

**Overcoming dataset bias**

Our Bayesian methodology also overcomes dataset bias, which has be noted in the recent literature as the leading cause for overly optimistic results on benchmarks \cite{37}. Most ligand-based benchmarks are biased in the sense that the molecules reported are tightly clustered around a few important chemical scaffolds, such that when the dataset is randomly split into training set and test set, the molecules in the test set are structurally very similar to the training set. Therefore, a model that only memorises the training set will still achieve a high accuracy on the test set yet cannot generalise to other regions of chemical space. Methods such as scaffold splitting \cite{10} and attribution \cite{38} attempt to estimate what would be the true performance of the model if the dataset were...
FIG. 2. Epistemic uncertainty and aleatoric uncertainty are distinct sources of uncertainty, and a combination of them is needed to obtain a good estimate of model error. We plot the confidence-error curve for semi-supervised learning with Stein Variational Gradient Descent where the confidence is estimated from combining epistemic and aleatoric uncertainty, epistemic uncertainty alone, and aleatoric uncertainty alone.

FIG. 3. Dataset bias can be mitigated with Bayesian uncertainty estimation. We consider a toy problem of predicting computationally calculated log $P$ using Stein Variational Gradient Descent and semi-supervised learning, with a biased dataset comprising all beta-lactams or a benzodiazepines from ChEMBL. (Left) The model performance when the test set is also drawn from beta-lactams and benzodiazepines. (Middle) The model performance when the test set is all steroids from ChEMBL. (Right) The distribution of predicted uncertainty for the model applied to steroids and the model applied to beta-lactams and benzodiazepines.

not biased. However, bias is fundamental in chemical data – an “uniform distribution” in chemical space does not exist because chemical space does not have a well-defined metric. Regardless of how one preprocesses the dataset or train the model, model predictions will always be awry for scaffolds that are not represented in the dataset. As such, rather than “unbiasing” the data, the practical question is whether the model can estimate whether it is likely to make a correct prediction for an unseen molecule given a biased training set.

To show that Bayesian uncertainty estimation overcomes dataset bias, we consider a toy problem where we know the ground truth and deliberately introduce bias: We consider the problem of predicting log $P$ values, and use computed ACD log $P$ values as a surrogate. We construct a dataset comprising all molecules on ChEMBL with either a beta-lactam or a benzodiazepine scaffold. The dataset is obviously very biased as it only contains 2 scaffolds. We then train a model using Stein Variational Gradient Descent with semi-supervised learning,
FIG. 4. Semi-supervised learning significantly outperforms full supervised learning in active learning. The model starts with 25% of the full training set, at each iteration 2.5% of the full training set is added to the training set. The curves show the model error as a function of iteration.

Low data active learning

Having considered the quality of the uncertainty estimates in the data-abundant limit, our next question whether we can estimate uncertainty in the low data limit and drive an active learning cycle. We consider the objective of obtaining a low model error with a small training set. The model is first trained from a small initial pool of data (25% of the full training set, picked randomly), the model then selects a batch of molecules (2.5% of the full training set) that it is the most uncertain about to put into the training set, and then the model is retrained to suggest other additions, and the cycle continues. The test set is always 20% of the full dataset, held out at the beginning of the experiment. We note that other acquisition functions have been suggested in the literature [39], and the objective function is problem-dependent [18, 19]. Nonetheless, the goal of our experiment is to evaluate the quality of uncertainty estimate, thus we focus on a simple objective and acquisition functions. Further, as active learning requires constant retraining of the model, and Stein Variational Gradient Descent is significantly more computationally intensive than dropout, we will only consider dropout variational inference.

Figure 4 shows that semi-supervised learning significantly outperforms full supervised learning in the low data limit. Moreover, in the case of full supervised learning, active learning is unable to deliver a better learning curve than random sampling, whereas for semi-supervised learning there is a sizeable gap between the learning curves of random sampling and active learning. This is because the full supervised method generates molecular descriptors directly from data. Therefore, in the low data limit, it is unable to learn descriptors that describe the structure of chemical space and chemical similarity between compounds, thus cannot generate meaningful uncertainty estimates to drive active learning.
CONCLUSION

We propose a novel method to quantify uncertainty in molecular properties prediction. We show that our methodology significantly outperforms the baseline on a range of benchmark problems, both in terms of model accuracy and in terms of uncertainty estimates. Our method also overcomes dataset bias by returning a large uncertainty estimate when the test set is drawn from a different region of chemical space compared to the training set. Moreover, our methodology can drive an active learning cycle, maximising model performance while minimising the size of the training set. The key to the success of our method is the combination of semi-supervised learning and Bayesian deep learning. Semi-supervised learning allows us to learn informative molecular representation in the low data limit. Bayesian deep learning allows us to estimate aleatoric and epistemic uncertainty in a statistically principled manner. We exemplified our methodology on regression as it is generally more challenging than classification, although it can be readily extended to classification problems.

Our observation that the choice of Bayesian inference methodology significantly impacts the quality of the uncertainty estimate suggests an evident followup that probes the mathematical limit of Bayesian inference – i.e. benchmarking approximate inference techniques against importance sampling of the posterior using Markov Chain Monte Carlo till convergence, which is computationally expensive but mathematically exact. Moreover, we note that most approximate inference techniques in the literature have been benchmarked in terms of RMSE error or log-likelihood [40], rather than explicitly considering the quality of the uncertainty estimate in a manner relevant for chemoinformatics such as the confidence-error curve. An open question is the design of appropriate approximate inference techniques for graph convolutional neural networks that solves the trilemma between computational cost, model accuracy, and the quality of uncertainty estimate.

AAL acknowledges the support of the Winton Programme for the Physics of Sustainability.

*aal44@cam.ac.uk

[1] A. Cherkasov, E. N. Muratov, D. Fourches, A. Varnek, I. I. Baskin, M. Cronin, J. Dearden, P. Gramatica, Y. C. Martin, R. Todeschini, et al., Journal of medicinal chemistry 57, 4977 (2014).
[2] M. Randić, Journal of Mathematical Chemistry 7, 155 (1991).
[3] O. Ivanciuc, Journal of chemical information and computer sciences 40, 1412 (2000).
[4] J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse, Journal of chemical information and computer sciences 42, 1273 (2002).
[5] D. Rogers, R. D. Brown, and M. Hahn, Journal of biomolecular screening 10, 682 (2005).
[6] R. D. Cramer, D. E. Patterson, and J. D. Bunce, Journal of the American Chemical Society 110, 5059 (1988).
[7] J. Verma, V. M. Khedkar, and E. C. Coutinho, Current topics in medicinal chemistry 10, 95 (2010).
[8] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, IEEE Transactions on Neural Networks 20, 61 (2009).
[9] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, in Advances in neural information processing systems (2015) pp. 2224–2232.
[10] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Gienisse, A. S. Pappu, K. Leswing, and V. Pande, Chemical science 9, 513 (2018).
[11] R. P. Sheridan, B. P. Feuston, V. N. Maiorov, and S. K. Kearsley, Journal of Chemical Information and Computer Sciences 44, 1912 (2004).
[12] I. Sushko, S. Novotarskyi, R. Ko rner, A. K. Pandey, A. Cherkasov, J. Li, P. Gramatica, K. Hansen, T. Schroeter, K.-R. Mu ller, et al., Journal of chemical information and modeling 50, 2094 (2010).
[13] R. P. Sheridan, Journal of chemical information and modeling 52, 814 (2012).
[14] M. Toplak, R. Moc nik, M. Polajnar, Z. Bosnic, L. Carlsson, C. Hasselgren, J. Dems ar, S. Boyer, B. Zupan, and J. Sta ring, Journal of chemical information and modeling 54, 431 (2014).
[15] U. Norinder, L. Carlsson, S. Boyer, and M. Eklund, Journal of chemical information and modeling 54, 1596 (2014).
[16] F. Svensson, U. Norinder, and A. Bender, Toxicology Research 6, 73 (2017).
[17] A. Kendall and Y. Gal, in Advances in neural information processing systems (2017) pp. 5574–5584.
[18] D. Reker and G. Schneider, Drug discovery today 20, 458 (2015).
[19] D. Reker, P. Schneider, G. Schneider, and J. Brown, Future medicinal chemistry 9, 381 (2017).
[20] H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande, ACS central science 3, 283 (2017).
[21] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, ACS central science 4, 268 (2018).
[22] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, arXiv preprint arXiv:1704.01212 (2017).
[23] https://github.com/debbiemarksLlab/neural_fingerprint – the neural fingerprint code.
[24] M. E. Welsch, S. A. Snyder, and B. R. Stockwell, Current opinion in chemical biology 14, 347 (2010).
[25] H. Nguyen, S.-i. Maeda, and K. Oono, arXiv preprint arXiv:1711.10168 (2017).
[26] Q. Le and T. Mikolov, in International Conference on Machine Learning (2014) pp. 1188–1196.
[27] https://github.com/pfnet-research/hierarchical – molecular fingerprint code.
[28] Y. Gal and Z. Ghahramani, arXiv preprint arXiv:1506.02157 (2015).
[29] Q. Liu and D. Wang, in Advances In Neural Information Processing Systems (2016) pp. 2378–2386.
[30] C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola, and K. F. Jensen, Journal of chemical information and modeling 57, 1757 (2017).
[31] J. S. Delaney, Journal of chemical information and computer sciences 44, 1000 (2004).
[32] D. L. Mobley and J. P. Guthrie, Journal of computer-aided molecular design 28, 711 (2014).
[33] Z. Gaieb, C. D. Parks, M. Chiu, H. Yang, C. Shao, W. P. Walters, M. H. Lambert, N. Nevins, S. D. Bembeneck, M. K. Ameriks, T. Mirzadegan, S. K. Burley, R. E. Amaro, and M. K. Gilson, Journal of Computer-Aided Molecular Design (2019).
[34] F.-J. Gamo, L. M. Sanz, J. Vidal, C. de Cozar, E. Alvarez, J.-L. Lavandera, D. E. Vanderwall, D. V. Green, V. Kumar, S. Hasan, et al., Nature 465, 305 (2010).
[35] A. Bender and R. C. Glen, Journal of chemical information and modeling 45, 1369 (2005).
[36] J.-F. Truchon and C. I. Bayly, Journal of chemical information and modeling 47, 488 (2007).
[37] I. Wallach and A. Heifets, Journal of chemical information and modeling 58, 916 (2018).
[38] K. McCloskey, A. Taly, F. Monti, M. P. Brenner, and L. Colwell, arXiv preprint arXiv:1811.11310 (2018).
[39] Y. Gal, R. Islam, and Z. Ghahramani, arXiv preprint arXiv:1703.02910 (2017).
[40] J. Mukhoti, P. Stenetorp, and Y. Gal, arXiv preprint arXiv:1811.09385 (2018).