iScience

CellPress
OPEN ACCESS

Review

Uncertainty quantification: Can we trust artificial intelligence in drug discovery?

Jie Yu,[1,2,3] Dingyan Wang,[1,2,3] and Mingyue Zheng[1,2,*]

SUMMARY

The problem of human trust is one of the most fundamental problems in applied artificial intelligence in drug discovery. In silico models have been widely used to accelerate the process of drug discovery in recent years. However, most of these models can only give reliable predictions within a limited chemical space that the training set covers (applicability domain). Predictions of samples falling outside the applicability domain are unreliable and sometimes dangerous for the drug-design decision-making process. Uncertainty quantification accordingly has drawn great attention to enable autonomous drug designing. By quantifying the confidence level of model predictions, the reliability of the predictions can be quantitatively represented to assist researchers in their molecular reasoning and experimental design. Here we summarize the state-of-the-art approaches to uncertainty quantification and underline how they can be used for drug design and discovery projects. Furthermore, we also outline four representative application scenarios of uncertainty quantification in drug discovery.

INTRODUCTION

Artificial intelligence (AI) and other data-driven approaches are reshaping drug discovery and design processes. For tasks with large amounts of training data, supervised learning can effectively map the relationship between inputs and outputs. A typical scenario is predicting protein structure based on primary sequence, where AlphaFold2 (Jumper et al., 2021) is believed to have solved this half-century problem (Buel and Walters, 2022). However, in most drug design tasks, the amounts of available training data are often limited (Altae-Tran et al., 2017). The inconsistency between the distribution of training data and test data may cause the model to produce unreliable outputs, which may have adverse consequences on decision-making procedure of drug design (Begoli et al., 2019). Unfortunately, classical deep learning (DL) models do not provide confidence estimation for their outputs. For regression tasks, the output is a single deterministic value without any uncertainty measurement. For classification tasks, the output is a probability distribution, which can be taken as the prediction confidence to some extent but is often poorly calibrated (Mervin et al., 2020). To illustrate this more vividly, we built a toy dataset, in which x is a real number ranging from 0 to 20 and y is a binarized label indicating whether $\frac{1+\sin(x)}{2}$ is larger than 0.5 (y = 1), or otherwise (y = 0). As shown in Figure 1A, the toy dataset is split into the training part ($x < 12$) and the test part ($x \geqq 12$). A neural network with 2 hidden layers and the Softmax output layer was trained on the training set. Figure 1B shows the probabilities given by the model on the training set and the test set. As shown, the model is well fitted on the training part, but gives overconfident false prediction on the test part. It is observed that the probability solely given by the Softmax function cannot be taken as the confidence of the prediction reliably. Thus, novel UQ strategies that are more effective, well-calibrated, and compatible with the different structures of neural networks are highly demanded. (Mervin et al., 2021a).

Evaluating the quality of a UQ method is tricky owing to the requirement of taking application scenarios and objectives of users into consideration, but in general, the ranking and calibration ability of UQ methods are the most two aspects that we are concerned. Ranking ability is intended to characterize the correlation between uncertainty and error. A UQ method with an ideal ranking ability should assign higher uncertainty values to predictions with larger errors. For regression tasks, appropriate correlation coefficients (e.g., Spearman correlation coefficient) can be used to quantitatively describe the correlation between prediction error and uncertainty. For classification tasks, it is expected that the wrong predicted samples could be intelligently prioritized by uncertainty. Specifically, the samples that are incorrectly and correctly classified can be regarded as positives and negatives, respectively, and then the ranking ability of UQ methods

[1]Drug Discovery and Design Center, State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica, Chinese Academy of Sciences, 555 Zuchongzhi Road, Shanghai 201203, China

[2]University of Chinese Academy of Sciences, No. 19A Yuquan Road, Beijing 100049, China

[3]These authors contributed equally

*Correspondence: myzheng@simm.ac.cn

https://doi.org/10.1016/j.isci.2022.104814

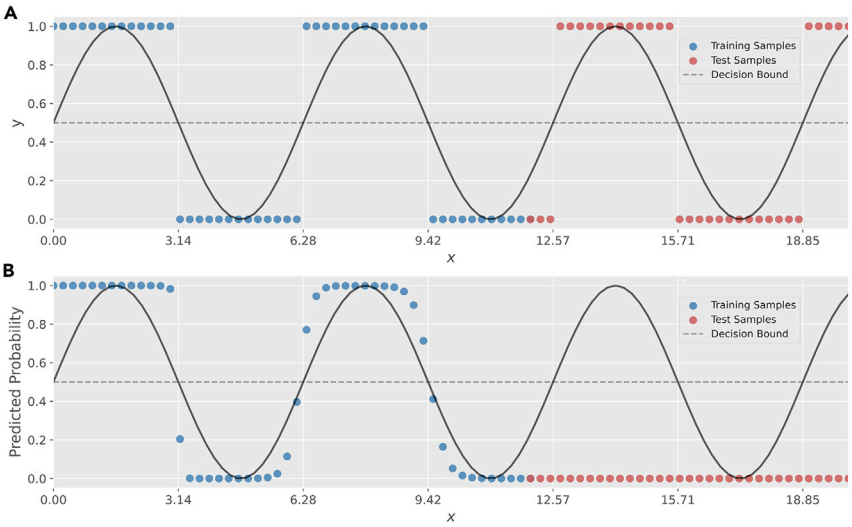

Figure 1. The probability given by the Softmax function cannot be taken as the confidence of the prediction reliably

(A) A toy dataset is built for illustration, in which x is a real number ranging from 0 to 20 and y is a binarized label indicating whether $\frac{1}{2}+\frac{1}{2}\sin(x)$ is larger than 0.5 ($y=1$), or otherwise ($y=0$). The dataset is split into the training part ($x < 12$) and the test part ($x \geq 12$). A neural network with 2 hidden layers and the Softmax output layer was trained on the training set.

(B) The figure shows the probability given by the model on the training set and the test set. As it can be seen, the model is well fitted on the training part, but gives overconfident false predictions on the test part.

can be quantified by auROC (area under the receiver operating characteristic curve) or auPRC (area under the precision—recall curve). Calibration ability is intended to characterize the ability to indicate the error distribution. For example, under the regression setting, it is expected that a UQ model could precisely estimate the variance of the error distribution, which is useful and important for confidence interval estimation.

In the chemistry community, there have been some concepts similar to uncertainty quantification for a long time, among which the most common one is the definition of the AD (applicability domain) (Sheridan, 2012, 2013, 2015) of QSAR (quantitative structure-activity relationship) models. In the following content, we will clearly specify the relationship between the two in this review to avoid confusion. UQ and AD share the same purpose: to help researchers determine whether the prediction result of a sample is reliable. Predictions for compounds outside the application domain will be thought to be less reliable (corresponding to higher uncertainty), and vice versa. Thus, UQ and AD are closely linked. Compared with UQ, traditional applicability domain definition methods are more input-oriented, generally considering the feature space or sub-feature space of samples, less considering the structure of the model itself. Correspondingly, the concept of UQ is broader and can refer to all the methods used to determine whether a prediction is reliable or not in general. As a result, AD definition methods are conceptually covered by UQ. Here, some classical AD definition methods are classified as similarity-based UQ methods and will be introduced in the "similarity-based approaches" section.

In this article, we intend to give a review of the concept, methods, and applications of UQ in the current drug design and discovery paradigm. It is worth noting that we will not thoroughly cover the available UQ strategies out of the context of drug design, especially considering that the review of Abdar et al. (Abdar et al. (2021) has conducted this job. Instead, we pay more attention to specific application cases of UQ and explain the underlying principles of the methods used, and we hope this review will give insights and practical guidance for deploying trustworthy AI models in drug design.

SOURCES OF UNCERTAINTY IN DRUG DISCOVERY

According to different sources, uncertainty can be broadly divided into three categories: approximation, aleatoric and epistemic uncertainties (Kiureghian and Ditlevsen, 2009). Approximation uncertainty

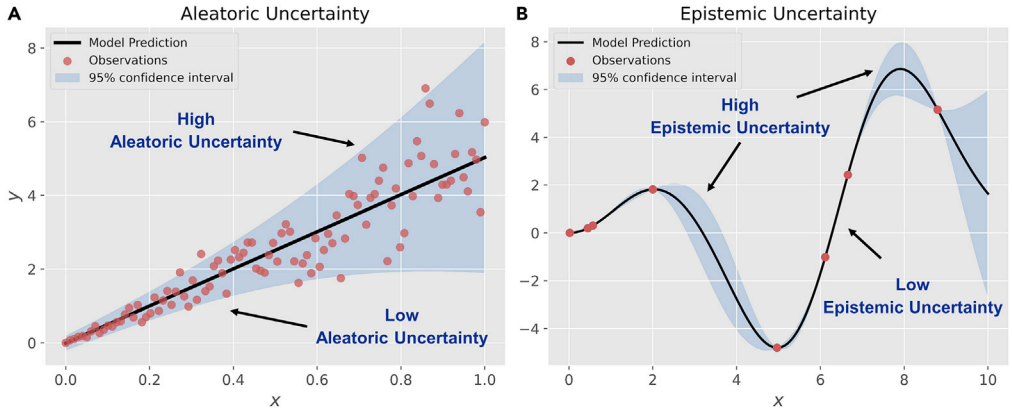

Figure 2. Illustration of aleatoric uncertainty and epistemic uncertainty

The fitted model is represented as a black solid line and the observed data are represented as red points. The blue area means the 95% confidence interval (uncertainty measurement).

(A) A probabilistic neural network is built to provide a confidence interval of the prediction. The model assigns a lower aleatoric uncertainty to the data points in a regular pattern (low noise data), and a higher aleatoric uncertainty to the data points in a random pattern (high noise data).

(B) A Gaussian Regression Process model is used to provide a confidence interval of the prediction. The predictions in the space with no (or lack of) observed data points are assigned higher epistemic uncertainty, but the predictions in the space with observed data points are assigned lower epistemic uncertainty.

accounts for the errors caused by the incompetence of simplistic models to fit complex data, such as the error made by a linear model fitting a sinusoidal curve (Tagasovska and Lopez-Paz, 2019). However, because deep neural networks are known to be universal approximators, approximation uncertainty is always assumed to be negligible. More details are directed to a study by Lazic et al., which provides an introduction to sources of uncertainty, including the approximation uncertainty (Lazic and Williams, 2021). In this section, we will focus on the introduction of aleatoric and epistemic uncertainties.

Aleatoric uncertainty

Aleatoric uncertainty (derived from the Latin *alea*, which means the rolling of dice) describes the intrinsic random nature (noise) of data to be modeled (Tagasovska and Lopez-Paz, 2019). In Figure 2A, the fitted model is represented as a black solid line, and the observed data are represented as red points. As it can be seen, the model assigns a lower aleatoric uncertainty to the data points in a regular pattern (low noise data), and a higher aleatoric uncertainty to the data points in a random pattern (high noise data). As an inherent attribute of data, aleatoric uncertainty cannot be reduced by collecting more training data. In drug discovery projects, the data noise is always derived from the different experimental measurements that are complicated by two main sources of error: systematic error and random error (Kolmar and Grulke, 2021). Hence, aleatoric uncertainty is often used to estimate whether the maximal performance of a model has been reached (i.e., when models approximate experimental error) (Beker et al., 2020), which will be detailed in the "improving model accuracy and robustness" section.

Epistemic uncertainty

Epistemic uncertainty (derived from Greek *episteme*, which means "knowledge") represents the errors associated with the lack of knowledge of the trained model in certain regions of the sample space (e.g., the chemical space outside AD of the model) (Tagasovska and Lopez-Paz, 2019). As shown in Figure 2B, the predictions in the space with no (or lack of) observed data points are assigned higher epistemic uncertainty, but the predictions in the space with observed data points are assigned lower epistemic uncertainty. Hence, unlike aleatoric uncertainty, epistemic uncertainty can be neutralized by collecting the data in those low-density regions. Samples with higher epistemic uncertainty can provide more informative insights into models (e.g., novel structure-activity relationship). Therefore, epistemic uncertainty can be used to guide experiment design to annotate data with less experimental cost while maximizing a model's performance gain (Ding et al., 2021). The corresponding application is referred to as active learning (AL), which will be detailed in the "active learning" section.

Table 1. The summary of the uncertainty quantification methods

UQ methods	Core idea	Representative methods[a]	Example applications[a]
Similarity-based	If a test sample is too dissimilar to training samples, the corresponding prediction is likely to be unreliable.	1. Box Bounding (Netzeva et al., 2005) 2. Convex Hull (Jaworska et al., 2005) 3. DM (Sheridan et al., 2004) 4. SDC score (Liu et al., 2018) 5. NNAS (Allen et al., 2020)	1. Virtual screening (Berenger and Yamanishi, 2019) 2. Anticancer peptide activity prediction (Chen et al., 2021) 3. SARS-CoV 2 inhibitor prediction (Gawriljuk et al., 2021) 4. Toxicity prediction (Jiang et al., 2021)
Bayesian	Parameters and outputs are treated as random variables and maximum a posteriori (MAP) estimation is adopted according to Bayes' theorem.	1. VI (MC-dropout) (Gal and Ghahramani, 2016) 2. BNN (Goan and Fookes, 2020) 3. GP-MGK (Xiang et al., 2021) 4. MVE (Nix and Weigend, 1994) 5. Bayesian GCN (Ryu et al., 2019)	1. Molecular property prediction (Zhang and Lee, 2019) 2. Virtual screening (Ryu et al., 2019) 3. Protein-ligand interaction prediction (Kim et al., 2021)
Ensemble-based	The consistency of the predictions from various base models is an estimate of confidence.	1. Bootstrapping (Scalia et al., 2020) 2. RF (Sheridan, 2012) 3. DeltaDelta (Jimenez-Luna et al., 2019) 4. Deep ensemble (Lakshminarayanan et al., 2017) 5. MC-dropout (Gal and Ghahramani, 2016)	1. Drug-likeness prediction (Beker et al., 2020) 2. Molecular property prediction (Scalia et al., 2020) 3. Lead optimization (Jimenez-Luna et al., 2019)

[a]The representative methods and example applications are not exhaustive.

METHODS OF UNCERTAINTY QUANTIFICATION

A large number of UQ methods have been deployed in drug discovery projects. Here, we put forward a new taxonomy to track the development path of various UQ methods. By focusing on the theoretical foundations of these UQ methods, we categorize them into three types: similarity-based, Bayesian, and ensemble-based approaches. For clarity, we summarized their core ideas, representative methods, and example applications in Table 1. These UQ methods and associated concepts are reviewed in the following sections.

Similarity-based approaches

Similarity-based approaches basically adopt the concept that if a test sample is too dissimilar to training samples, the corresponding prediction is likely to be unreliable. In practice, users should first choose or define a method to measure the distance between the test samples and the training samples, and then the distance can be regarded as the estimated uncertainty of the prediction. Some of these approaches have been widely used to define the AD for QSAR models.

A simple similarity-based approach named Bounding Box defines a range of acceptable values for each descriptor based on the distribution of its values in the training set (Netzeva et al., 2005). For a query sample, if the value of at least one descriptor falls out of the defined range, the sample is regarded as "out-of-distribution." The more descriptors that break the criteria, the more uncertain the prediction is. Instead of directly using the raw descriptor, sometimes a reduced-dimension strategy, like PCA (principal components analysis), will be performed first to reduce the feature space (Carrio et al., 2014). The lower bound and the upper bound of the acceptable range is usually decided by the minimum and maximum value of the descriptor in the training set, but sometimes the top and bottom 5th percentiles are used. Another similarity-based approach considers the activity space similarity rather than feature space (Keefer et al., 2013). It is assumed that if the predicted value of a query sample is not consistent with the labels of the structurally similar training samples, which indicates that the SAR (structure-activity relationship) landscape is not smooth, the prediction is then considered unreliable. Generally, these approaches suffer from the shortcoming of too strong assumption for the distribution of features (independent variables x) or labels

(independent variable y). For example, most of these methods assume that features are independent of each other, and can only provide binarized or discrete uncertainty estimation (reliable or unreliable) results, which limits their application.

Different from the methods mentioned above, another kind of similarity-based approach considers the overall distance between samples, usually called the distance-to-model (DM) method. The application of the DM method should define the distance between two samples $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ first, which depends on the format of features. If features are Boolean vectors, Tanimoto similarity (also called Jaccard index) D_T is often used:

$$D_T\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \frac{\sum \boldsymbol{x}_m^{(i)} \boldsymbol{x}_m^{(j)}}{\sum \boldsymbol{x}_m^{(i)} \boldsymbol{x}_m^{(i)} + \sum \boldsymbol{x}_m^{(j)} \boldsymbol{x}_m^{(j)} - \sum \boldsymbol{x}_m^{(i)} \boldsymbol{x}_m^{(j)}} \quad \text{(Equation 1)}$$

where $\boldsymbol{x}_m^{(i)}$ is the m-th feature value of molecule $\boldsymbol{x}^{(i)}$. If $\boldsymbol{x}^{(i)}$ is a continuous vector, Euclidean distance D_E is often used:

$$D_E\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \sqrt{\sum_{m=1}^{M} \left(\boldsymbol{x}_m^{(i)} - \boldsymbol{x}_m^{(j)}\right)^2} \quad \text{(Equation 2)}$$

where M is the total feature length. Once it has decided how to calculate the distance between samples, we can further define the distance between a test sample and the training set, which can be further taken as predictive uncertainty. Many strategies can be applied in this procedure, for example, the average distance to the nearest k training samples (Sheridan et al., 2004) or the distance to the representative average of the training set (Berenger and Yamanishi, 2019). The threshold for defining AD can be decided by analyzing the training data distribution (Sahigara et al., 2013).

Instead of computing distances, some methods define an acceptable high-dimensional range and assume that the query sample within this range can be readily predicted. An example is the Convex Hull strategy, which defines the smallest convex area that covers the training points (Jaworska et al., 2005). It can also be taken as an extension of the Bounding Box method.

Recently, some more complex similarity-based approaches have emerged. For example, the SDC score proposed by Liu et al. (Liu et al., 2018; Liu and Wallqvist, 2019) uses the contribution of all training molecules to estimate the reliability of a prediction, in which the training sample contribution is weighted down exponentially by the distance.

It is noticed that the above-mentioned similarity-based approaches are highly dependent on how to feature samples. However, by engineering raw features, DL models could project samples into a mission-specific latent space in which distances can also be treated as an uncertainty metric. Janet et al. tested this idea on two diverse chemical datasets and found that latent space distance outperformed other well-established uncertainty metrics without any additional training cost (Janet et al., 2019). In a similar way, Allen et al. proposed NNAS (neural network activation similarity), a kind of latent space distance, to increase prediction confidence in toxicity safety evaluation. They found that NNAS outperformed Tanimoto similarity and RFS (random forest similarity) regarding similarity searching (Allen et al., 2020).

Bayesian approaches

The training process of a neural network can be taken as learning the optimal parameters $\boldsymbol{\theta}$ for a probabilistic model $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})$. Frequentists and Bayesians adopt different strategies for solving this problem, and their differences are visualized in Figure 3. As shown in Figures 3A and 3C, for frequentists, the parameters are fixed but with unknown quantities, and can be estimated by the maximum likelihood estimation (MLE). This corresponds to the standard training protocol that minimizes the empirical loss (Nix and Weigend, 1994; Scalia et al., 2020). On the other hand, as shown in Figures 3B and 3D, Bayesians treat parameters as random variables and adopt maximum a posteriori (MAP) estimation or directly give the posterior distribution of parameters according to Bayes' theorem. This is called the Bayesian neural network (BNN), where model weights and outputs are both distributions instead of determined values (Goan and Fookes, 2020). Different from the standard neural network, BNN has the advantage of directly capturing the uncertainty of the prediction (Olivier et al., 2021). To briefly show this, assuming that model parameters follow the prior distribution $p(\boldsymbol{\theta})$ (e.g., a normal distribution), and model likelihood is $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta})$, where $\boldsymbol{X}$ refers to the

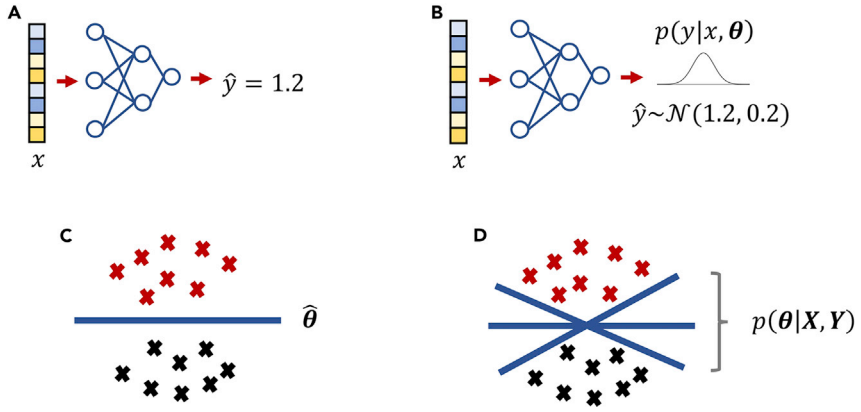

Figure 3. Comparison between the traditional neural network and Bayesian neural network

The outputs and parameters of the traditional neural network are deterministic values (A and C), while in the Bayesian neural network they are distributions (B and D).

feature vectors and $\boldsymbol{Y}$ refers to the label vector, we obtain the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y})$ according to the concept of Bayesian inference Equation (3):

$$p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) = \frac{p(\boldsymbol{\theta})p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})}{\int p(\boldsymbol{\theta})p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta})d\boldsymbol{\theta}} \quad \text{(Equation 3)}$$

where $(\boldsymbol{X}, \boldsymbol{Y})$ corresponds to the training set as "seen" by the model, and the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y})$ is the joint probability distribution of model weights learned (conditioned) by fitting the training set. Once the distribution of model weights is determined, for a query sample $\boldsymbol{x}^*$, its prediction $\hat{y}^*$, a distribution, can be calculated using Equation (4):

$$p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}) = \int p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y})d\boldsymbol{\theta} \quad \text{(Equation 4)}$$

where the final prediction $p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y})$ could be understood as a "weighted sum" of each prediction $p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})$ for each set of possible model weights $\boldsymbol{\theta}$, and the probability of $\boldsymbol{\theta}$ depends on the training set $(\boldsymbol{X}, \boldsymbol{Y})$. For regression tasks, as $\hat{y}^*$ is a variable following a distribution instead of a deterministic number in the Bayesian neural network, we can now define the uncertainty of $\hat{y}^*$ as its variance, which can be calculated according to Equation (5):

$$\underbrace{\mathrm{var}[\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}]}_{\textit{Total Uncertainty}} = \underbrace{\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})}[\mathrm{var}[\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}]]}_{\textit{Aleatoric Uncertainty}} + \underbrace{\mathrm{var}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})}[\mathbb{E}[\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}]]}_{\textit{Epistemic Uncertainty}} \quad \text{(Equation 5)}$$

It can be seen that the total uncertainty is decomposed into the former term aleatoric uncertainty and the latter term epistemic uncertainty, which has been introduced in the "sources of uncertainty in drug discovery" section. Directly using Equation (5) to calculate $\mathrm{var}[\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}]$ (total uncertainty) faces two problems. First, it is required to define the likelihood $p(y|\boldsymbol{x}, \boldsymbol{\theta})$. For regression tasks, mean-variance estimation (MVE) is often used (Nix and Weigend, 1994). In MVE, the output of a neural network (with determined model weights $\boldsymbol{\theta}$) is defined as a Gaussian distribution, and the task of the neural network is to give the mean $\mu(\boldsymbol{x}^*, \boldsymbol{\theta})$ and variance $v(\boldsymbol{x}^*, \boldsymbol{\theta})$ of the distribution:

$$[\mu(\boldsymbol{x}^*, \boldsymbol{\theta}), v(\boldsymbol{x}^*, \boldsymbol{\theta})] = f_{\boldsymbol{\theta}}(\boldsymbol{x}^*) \quad \text{(Equation 6)}$$

$$p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}) = \mathcal{N}(\mu(\boldsymbol{x}^*, \boldsymbol{\theta}), v(\boldsymbol{x}^*, \boldsymbol{\theta})) \quad \text{(Equation 7)}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{x}^*)$ refers to the model output. In practice, the output layer of the neural network is branched into two predictions (a 2-dimensions vector), the mean $\mu(\boldsymbol{x}^*, \boldsymbol{\theta})$ and the variance $v(\boldsymbol{x}^*, \boldsymbol{\theta})$. Owing to the non-negativity of variance, we generally predict its log value. In addition to this minimal modification on model output layer, the loss function should be changed as the form shown as Equation (9), which is obtained by performing MAP inference on Gaussian probability density function [Equation (8)].

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\, v(\boldsymbol{x}, \boldsymbol{\theta})}} \exp\left(\frac{-[y - \mu(\boldsymbol{x}, \boldsymbol{\theta})]^2}{2v(\boldsymbol{x}, \boldsymbol{\theta})}\right) \quad \text{(Equation 8)}$$

where y is the true label of the sample $\boldsymbol{x}$.

$$\mathcal{L}(\boldsymbol{\theta}) \propto \frac{1}{N}\sum_{i=1}^{N}\left(\frac{[y_i - \mu(\boldsymbol{x}_i,\boldsymbol{\theta})]^2}{2v(\boldsymbol{x}_i,\boldsymbol{\theta})} + \frac{1}{2}\ln[v(\boldsymbol{x}_i,\boldsymbol{\theta})]\right) \quad \text{(Equation 9)}$$

where N is the number of training samples, and y_i is the true label of i-th training sample $\boldsymbol{x}_i$. In MVE, labels are assumed to carry underlying Gaussian errors that indicate the noise in the labels. Pytorch (Paszke et al., 2019), a popular deep-learning python library, has implemented a function (torch.nn.functional.gaussian_nll_loss, version 1.11.0) for conveniently calculating MVE loss.

The second problem is that the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})$ cannot be calculated analytically owing to the intractable calculation of $p(\boldsymbol{Y}|\boldsymbol{X})$. Some strategies are often used to make an approximation, for example, variational inference (VI) (Blei et al., 2017), which constructs a variational distribution $q(\boldsymbol{\theta})$ to approximate $p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})$ by minimizing the Kullback-Leibler divergence between $q(\boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})$. VI methods constitute a standard technique in Bayesian modeling. However, its high computational cost still limits its application. Thus, some approximate ways have been implemented to circumvent its computational intractability, such as an ensemble that consists in training the same network multiple times with random initialization. The process of training a model could be deemed as taking a sampling of the real distribution of model weights $p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})$. Here, these approximate ways are classified as "Ensemble-based approaches" and will be detailed and introduced in next section.

After the acquisition of sampled weights from $q(\boldsymbol{\theta})$, $\mathrm{var}[\widehat{y}^*|\boldsymbol{x}^*,\boldsymbol{X},\boldsymbol{Y}]$ can be approximated as Equation (10), as proposed by Kendall et al. (Kendall and Gal, 2017):

$$\underbrace{\mathrm{var}[\widehat{y}^*|\boldsymbol{x}^*,\boldsymbol{X},\boldsymbol{Y}]}_{\textit{Total Uncertainty}} \approx \underbrace{\frac{1}{T}\sum_{t=1}^{T} v(\boldsymbol{x}^*,\boldsymbol{\theta}_t)}_{\textit{Aleatoric Uncertainty}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\mu(\boldsymbol{x}^*,\boldsymbol{\theta}_t)^2 - \left(\frac{1}{T}\sum_{t=1}^{T}\mu(\boldsymbol{x}^*,\boldsymbol{\theta}_t)\right)^2}_{\textit{Epistemic Uncertainty}} \quad \text{(Equation 10)}$$

where $\{\boldsymbol{\theta}_t\}_{t=1}^{T} \sim q(\boldsymbol{\theta})$ are sampled model weights. Zhang et al. benchmarked this approach in the context of molecular property prediction based on 6 datasets (Zhang and Lee, 2019). Results showed that the total uncertainty is a better estimate of error than any single source of uncertainty. Scalia et al. drew the same conclusion in a recent benchmarking test of molecular property prediction, again highlighting the importance of considering both sources of uncertainty (Scalia et al., 2020).

For classification problems, label y^* can be expressed as:

$$y^* \in [e_1, \ldots e_c, \ldots, e_C] \quad \text{(Equation 11)}$$

where e_c is a one-hot encoded vector whose c-th element is 1 and other positions are zeros. For example, for a typical binary classification problem, y^* can be either [0, 1] or [1, 0]. The likelihood function, or the predicted probability of the model that the sample belongs to the c-th class, is given by:

$$p(y = e_c|\boldsymbol{x},\boldsymbol{\theta}) = \frac{\exp\left(f_{\boldsymbol{\theta}}^{(c)}(\boldsymbol{x})\right)}{\sum_{i=1}^{C}\exp\left(f_{\boldsymbol{\theta}}^{(i)}(\boldsymbol{x})\right)} = p_c \quad \text{(Equation 12)}$$

where $f_{\boldsymbol{\theta}}^{(c)}(\boldsymbol{x})$ is the c-th element of the pre-activated model output and the probability vector $\mathbf{p} = [p_1 \ldots, p_c, \ldots, p_C] = softmax(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$ is the final output.

There exist several different methods for UQ in classification settings. Here we introduce two of them. The first was proposed by Kwon et al. (Kwon et al., 2020) which aimed at calculating the total variance of prediction as we have conducted in the regression setting:

$$\underbrace{\mathrm{var}[\widehat{y}^*|\boldsymbol{x}^*,\boldsymbol{X},\boldsymbol{Y}]}_{\textit{Total Uncertainty}} \approx \underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(diag(\mathbf{p}_t) - (\mathbf{p}_t)(\mathbf{p}_t)^T\right)}_{\textit{Aleatoric Uncertainty}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\mathbf{p}_t - \overline{\mathbf{p}})(\mathbf{p}_t - \overline{\mathbf{p}})^T}_{\textit{Epistemic Uncertainty}} \quad \text{(Equation 13)}$$

where $\overline{\mathbf{p}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{p}_t$ is the predictive mean and $\mathbf{p}_t = softmax(f_{\boldsymbol{\theta}_t(\boldsymbol{x}^*)})$ is the prediction of a single model whose weights are sampled from $q(\boldsymbol{\theta})$, as is conducted in Equation (10). Ryu et al. applied this method to develop a

Bayesian graph convolutional network (GCN) for molecular property prediction (Ryu et al., 2019). They demonstrated that the usage of Bayesian GCN in quantifying prediction uncertainty improves the virtual screening accuracy and can quantitatively evaluate training data quality. Kim et al. also used this method to develop a Bayesian neural network for protein-ligand interaction prediction, which showed better performance than previous baselines (Kim et al., 2021). Beker et al. applied this method for decomposing the total error within predictions of drug-likeness into the aleatoric and epistemic components (Beker et al., 2020).

For the second method, instead of variance, the entropy of $\mathbf{p}$ is used to quantify the uncertainty of probability distribution (Shannon, 1948).

$$H(\mathbf{p}) = -\sum_{c=1}^{C} p_c \log_2 p_c \quad \text{(Equation 14)}$$

However, the entropy of a single model output does not distinguish between aleatoric and epistemic uncertainties. To achieve this goal, Smith et al. (Smith and Gal, 2018) proposed that the predictive entropy $H[p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y})]$ can be taken as the total uncertainty, the expected entropy $\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})}[H[p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})]]$ as the aleatoric uncertainty, and the mutual information $MI[\hat{y}^*; \boldsymbol{\theta}|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}]$ as the epistemic uncertainty. Once the ensemble $\{p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}_t)\}_{t=1}^{T}$ has obtained, these terms can be approximated as the following equation:

$$\underbrace{MI[\hat{y}^*; \boldsymbol{\theta}|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}]}_{\textit{Epistemic Uncertainty}} = \underbrace{H\left[p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y})\right]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{Y})}\left[H\left[p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta})\right]\right]}_{\textit{Aleatoric Uncertainty}} \approx \underbrace{H\left[\frac{1}{T}\sum_{t=1}^{T}\mathbf{p}_t\right]}_{\textit{Total Uncertainty}} - \underbrace{\frac{1}{T}\sum_{t=1}^{T}H\left[\mathbf{p}_t\right]}_{\textit{Aleatoric Uncertainty}} \quad \text{(Equation 15)}$$

Yildirim et al. used this method to filter out false positive predictions in the semantic segmentation of particle instances in EM images (Yildirim and Cole, 2021).

Except for BNN, the Gaussian process (GP) is another classical Bayesian machine-learning approach that can provide native uncertainty for its predictions (Williams and Rasmussen, 1996). The most obvious similarity between GP and BNN is that the predictions of these models are both probabilistic and can be used to infer predictive uncertainty or compute empirical confidence intervals. Taking regression as an example, Gaussian Process Regression (GPR) models the inputs and outputs using Equation (16):

$$y = f(\boldsymbol{x}) + \varepsilon \quad \text{(Equation 16)}$$

where f is a latent function and ε is a noise term, which is typically assumed to be normally distributed with zero mean and noise variance σ_n^2. Instead of explicitly modeling f using the neural network architecture, as is conducted in BNN, in GPR the latent function f is supposed to be drawn from a Gaussian Process prior with mean function $\mu(\cdot)$ and covariance function $\kappa(\cdot)$. Same as the MVE method [Equation (7)], in GPR the predictive distribution of $\hat{y}^*$ for a test sample $\boldsymbol{x}^*$ also follows a Gaussian distribution:

$$p(\hat{y}^*|\boldsymbol{x}^*, \boldsymbol{X}, \boldsymbol{Y}) = \mathcal{N}(\mu_*, \sigma_*^2) \quad \text{(Equation 17)}$$

in which the mean μ_* and variance σ_*^2 can be calculated using Equations (18) and (19):

$$\mu_* = \mathbf{k}_*^{\mathrm{T}}\left(\mathbf{K} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{Y} \quad \text{(Equation 18)}$$

$$\sigma_*^2 = k_{**} - \mathbf{k}_*^{\mathrm{T}}\left(\mathbf{K} + \sigma_n^2\mathbf{I}\right)^{-1}\mathbf{k}_* + \sigma_n^2 \quad \text{(Equation 19)}$$

where $\mathbf{K} = \kappa(\mathbf{X}, \mathbf{X})$, $\mathbf{k}_* = \kappa(\mathbf{X}, \boldsymbol{x}^*)$ and $k_{**} = \kappa(\boldsymbol{x}^*, \boldsymbol{x}^*)$. This process equals marginalizing over the infinite possible latent functions f. The same as Equation (5), here σ_*^2 can be taken as the uncertainty of the prediction. As a non-parametric model (function form of f is not specified), GP is more flexible than BNN, but suffers the burden of storing training data points for computing the covariance matrix (Li et al., 2021). The machine-learning package scikit-learn (Pedregosa et al., 2011) provides a convenient API for building GP models.

The application of GP in computational chemistry and chemoinformatics has been well studied (Deringer et al., 2021). DiFranzo et al. proposed a nearest neighbor Gaussian process model for QSAR modeling. They found that the variance of model output provides calibrated uncertainty estimation

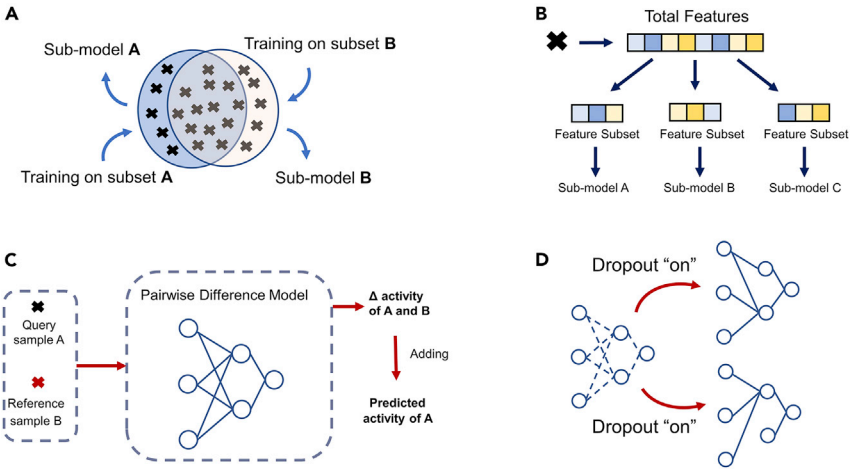

Figure 4. Illustration of ensemble-based UQ methods

(A) Data perturbation. Sub-models are trained based on different subsets of the original training set.

(B) Features perturbation. Sub-models are trained based on different subsets of the original sample features.

(C) Outputs perturbation. The output of the model is no longer a deterministic value, but a difference.

(D) Weights perturbation. The sub-models are generated by keeping dropout open in the prediction process.

(DiFranzo et al., 2020). Musil et al. presented a scheme based on subsampling and sparse GP regression for fast and reliable uncertainty estimation in the task of atomic and molecular property prediction (Musil et al., 2019). Xiang et al. proposed a GP model with a hybrid kernel, GP-MGK, for molecular property prediction (Xiang et al., 2021). They found that GP-MGK outperformed D-MPNN, a kind of graph convolutional neural network, regarding uncertainty quantification. These examples have demonstrated the usefulness of GP in chemical modeling and uncertainty estimation. However, more benchmarking tests are still needed for the comparison of GP with other state-of-the-art deep learning models (Hirschfeld et al., 2020).

Ensemble-based approaches

It has long been observed that ensemble learning improves predictive performance (Dietterich, 2000). Except for this, however, ensemble learning can also be used for UQ (Lakshminarayanan et al., 2017). Ensemble learning aims at constructing multiple similar but different base learners. In general, the predictions of the base learners are integrated into the final prediction (e.g., mean, median, and so forth) and their variance of them is deemed as an estimate of epistemic uncertainty. Here, we take random forest (for regression) as an example to illustrate the usage of ensemble-based UQ approaches in practice. For a query sample $\boldsymbol{x}^*$, the prediction $\widehat{y}^*$ is provided as the average of the predictions of all decision trees (base learners) $\{\widehat{y}_1^*, \widehat{y}_2^*, \ldots, \widehat{y}_T^*\}$, and the uncertainty of this sample $\mathrm{U}(\boldsymbol{x}^*)$ can be provided by the variance of the predictions of all decision trees.

$$\widehat{y}^* = \frac{1}{T}\sum_{t=1}^{T}\widehat{y}_t^* \qquad \text{(Equation 20)}$$

$$\mathrm{U}(\boldsymbol{x}^*) = \frac{1}{T}\sum_{t=1}^{T}\left(\widehat{y}_t^* - \widehat{y}^*\right)^2 \qquad \text{(Equation 21)}$$

where T is the number of decision trees . Different base learners will tend to output similar prediction values when the inputs are similar to the observed training data because each base learner's weights, even if different, are optimized for those data. In contrast, as inputs become less similar to the training data, the outputs of each base learner tend to be more sensitive to the specificities of the suboptimal solution reached, thus the higher variance (Scalia et al., 2020). Given this, it seems clear that diversity in the base learners should be promoted for uncertainty improvement. The general idea for promoting diversity is to introduce randomness into the training process, and the commonly used methods could be categorized into four styles: data, features, outputs, and weights perturbations. For clarity, they are visualized in Figure 4. These perturbation methods and associated UQ methods are reviewed in the following sections.

Data perturbation

Dataset perturbation is usually based on sampling. Given an initial dataset, different subsets could be sampled and then used to train different base learners for increasing diversity (Figure 4A). For example, bootstrapping (also referred to as bagging) is a popular technique where base learners are trained on different bootstrap samples of the original training set (Scalia et al., 2020). Dataset perturbation is highly efficient with some types of base learners such as neural networks that are sensitive to training data, but it may also impair the predictive performance of neural networks owing to the shrinkage of training data.

Features perturbation

For ML models, training samples are always represented by a set of attributes (e.g., molecular descriptors or molecular fingerprints) that could be thought of as a feature space, and different feature subspaces could provide various perspectives on samples. As shown in Figure 4B, features perturbation aims at describing samples from different feature subspaces to increase the diversity of the trained base learners. One of the most representative models is random forest (Saxe et al., 2021). The diversity of the base learners in the RF algorithm not only derives from data perturbation (bootstrap sampling), but also from features perturbation. Accordingly, the generalization ability of the final model could be improved and the variance of the predictions of these base learners could be regarded as predictive uncertainty (Sheridan, 2012).

Some data augmentation methods used in deep learning also share similarities with features perturbation. For example, considering that SMILES (Simplified molecular input line entry system) of a molecule are not unique, Kimber et al. used different SMILES to represent the same molecule for data augmentation, where SMILES are the input format of their model (Kimber et al., 2021). Similar to features perturbation, different SMILES can provide different perspectives on the same molecule. Based on this data augmentation method, they found that in addition to the benefit in the model performance, the variance of the predictions of the SMILES corresponding to the same molecule could also be taken as an estimate of uncertainty.

Outputs perturbation

Outputs perturbation (Figure 4C) enhances diversity by replacing the original task with other related tasks. For example, DeltaDelta, a pairwise difference regression model proposed by Jimenez-Luna et al., replaces the absolute activity (pIC_{50}) of a ligand with the activity difference (ΔpIC_{50}) between a pair of ligands as output (Jimenez-Luna et al., 2019). For DeltaDelta, a predicted pIC_{50} value of a new ligand could be recovered by first predicting the ΔpIC_{50} between the new ligand and any previously seen (pIC_{50} known) reference ligand, and then adding back in the pIC_{50} value of the reference molecule. By conducting this prediction procedure for all reference ligands and the new ligand, multiple predicted values of its pIC_{50} could be obtained and the variance of these predicted values could be regarded as an estimate of the uncertainty. Tynes et al. transferred this idea to molecular property prediction and observed similar results (Tynes et al., 2021).

Weights perturbation

Compared with other perturbation methods, weights perturbation methods force the base learners to get different weights more directly. Two representative examples are Deep Ensemble (Lakshminarayanan et al., 2017) and MC-dropout (Gal and Ghahramani, 2016; Kendall and Gal, 2017). Deep Ensemble is designed to train multiple base learners of the same structure with random initialization of model weights. Thus, different solutions can be easily reached by the base learners given their non-convexity and the suboptimal optimization strategies employed. MC-dropout consists in training a network with dropout before every layer and then, in the inference process, keeping dropout open to sample multiple outputs with different random masks (Figure 4D). Owing to the model-agnostic nature and ease of implementation, weights perturbation methods can be considered state-of-the-art for epistemic UQ in neural networks (Soleimany et al., 2021).

APPLICATION OF UNCERTAINTY QUANTIFICATION IN DRUG DISCOVERY

Estimation of model maximum achievable accuracy

The performance of in silico models depends on the quality of the training data (Saxe et al., 2021), and in most drug discovery projects, the labels of training data are always defined by experimental measurements with inherent variability (Kolmar and Grulke, 2021). As a result, the intrinsic label uncertainty or noise in the

training data determines the maximum achievable accuracy (MAA) of models (Kramer et al., 2012). Estimating the MAA of models based on the currently available data is highly instructive for follow-up machine learning studies. For example, if the accuracy of a model has approached the possible MAA, we should pay more attention to expanding the dataset or improving the quality of the training data rather than considering more sophisticated model architecture.

Given the close relationship between the label uncertainty of training data and the MAA of models described above, the problem of how to estimate the MAA of a model can be divided into two sub-problems: (1) How to estimate the label uncertainty in the currently available data, and (2) how to quantify the relationship between the label uncertainty and the MAA. A previous work by Kramer et al. provided the paradigm for the first sub-problem (Kramer et al., 2012). They first extracted all the high-quality Ki data from the ChEMBL database (Gaulton et al., 2012) through a series of data filtering steps. After that, they analyzed the differences between the published Ki measurements of identical protein-ligand systems to estimate the experimental error in the Ki data. Their experimental (or label) uncertainty estimation yielded a mean error of 0.44 pKi units with a standard deviation of 0.54 pKi units, which means that if the average error of a model based on heterogeneous (i.e., various laboratories, assay conditions, assay methods) sources of data are less than 0.44 pKi units, it is very likely that the model is overtrained. This work inspired a series of follow-up similar studies, such as quantitative estimation of label uncertainty in IC_{50} (Kalliokoski et al., 2013) and cytotoxicity data set (Cortes-Ciriano and Bender, 2016).

For the second sub-problem, several studies have attempted to artificially add simulated noises (usually sampled from normal distributions with different variances) to the labels of dataset to study the correlation between the label uncertainty of modeling data and model performance (Kolmar and Grulke, 2021; Sheridan et al., 2020). In this way, the originally unknown data noise is turned into a controllable variable with a known value. Kolmar et al. added 15 levels of simulated Gaussian distributed random error to 8 different QSAR datasets, and systematically evaluated the impact of random errors in the datasets on model performance using 5 different algorithms (Kolmar and Grulke, 2021). They found that the model performance did deteriorate with the introduction of label noise, and different kinds of machine learning models show varying degrees of robustness to noise.

In addition to directly estimating the average error of data, another strategy to infer the MAA of models is uncertainty quantification. Specifically, in the Bayesian system, total uncertainty can be divided into aleatoric and epistemic uncertainty according to different sources. The former is the result of irreducible and inherent data noise. The latter is caused by the insufficiency of knowledge provided by the training set. A more detailed description of them has been provided in the "sources of uncertainty in drug discovery" section. Therefore, the proportion of predicted aleatoric uncertainty in the total predicted uncertainty can be used to estimate whether a model has reached the possible MAA. Beker et al. systematically evaluated the performance of various AI models on the prediction of molecular drug-likeness using different types of molecular descriptors (Beker et al., 2020), where Deep Ensemble is used for uncertainty quantification. Based on the result that total uncertainty is comparable with its aleatoric contribution, they infer that the classification accuracy reported in their work (0.93) is probably the upper limit achievable with the current collection of known drugs.

Active learning

Owing to the time- and resource-intensive nature of biological and chemical experiments, how to generate new data to improve model performance more efficiently is a key problem in drug discovery (Yu et al., 2021). To address this issue, active learning (AL), an uncertainty-guided algorithm, has begun to show promise and has increasingly been used (Ding et al., 2021; Gong et al., 2021; Jansen et al., 2019; Yang et al., 2021). In AL, a model is typically initialized with a limited training set (e.g., currently available samples). Then, batches of unlabeled samples are iteratively selected based on a pre-defined query strategy (also referred to as selection function), labeled through associated experiments, and gradually added to the training set. The model is subsequently retrained using this expanded training set, with the expectation of more gains in prediction results on a held-out test set.

The query strategy is usually referred to a sampling method to decide which samples should be selected and labeled for each iteration, which is one of the most important components of AL. Depending on the query strategy used, AL could be divided into three categories: exploration-oriented AL,

exploitation-oriented AL, and hybrid AL (Ren et al., 2020). Exploration-oriented AL aims to select samples with the greatest predictive uncertainty. These samples may possess novel structures relative to their counterparts in the original training set. As a result, the AD of the retrained model could be enlarged effectively owing to the introduction of novel SAR. For example, Ding et al. explored the effectiveness of four UQ methods in exploration-oriented AL through a case study on the plasma exposure of orally administered drugs, and they found that the query strategy based on entropy is the most sample-efficient strategy (Ding et al., 2021). Besides, through complete experimental verification, their work also highlighted the effectiveness of the exploration-oriented AL in expanding the AD of models and guiding the experiment design.

Instead of selecting samples based on uncertainty, exploitation-oriented AL provides a framework to discover high-performing compounds (e.g., those with more favorable molecular properties) from a large search space by selecting the unlabeled samples with the highest scores in the iterative process. A typical application scenario of exploitation-oriented AL is structure-based virtual screening (VS) (Neves et al., 2018). As virtual libraries continue to grow [e.g., ZINC (Sterling and Irwin, 2015) now contains roughly 1 billion molecules], the computational resources necessary to conduct exhaustive virtual screening campaigns on these libraries are inaccessible to many academic researchers. Given this, combined with the AL algorithm, Graff et al. proposed a QSAR model to predict molecules' docking scores, which could enrich most of the molecules with high docking scores when only a few molecules were docked (Graff et al., 2021). However, they found that the chemical diversity of the molecules enriched by the QSAR model with purely exploitation-oriented AL is extremely low. To increase the chemical diversity, they employed a hybrid AL query strategy that incorporates both predicted docking scores and uncertainties to guide sample selection in the iterative process, which shows the unique status of UQ in the application of AL. Because of its flexibility in adjusting exploration-exploitation trade-off, hybrid AL query strategies (e.g., upper confidence bound) have gradually become the most widely used sampling methods in AL.

Virtual screening

High-throughput virtual screening has emerged as an important approach to identifying hit compounds from large chemical libraries (Shoichet, 2004). Among different types of VS strategies, DL-based VS has shown a promising hit rate and high throughput (Neves et al., 2018). In a typical workflow of DL-based VS, the drug-like compounds from a library are scored by a DL model, in which the top-scored ones are selected for further experimental verification. However, most commonly used chemical libraries cover extensive chemical space, most of which do not contain compounds with well-studied structures. It may cause a model to give overconfident predictions, accounting for the limited enrichment ability of conventional DL-based VS models. Incorporating UQ into the selection process to ensure the robustness of predictions is an intuitive way to deal with this problem. For example, if the DL model is trained to predict the pIC_{50} value (referred as $\widehat{y}$) and corresponding uncertainty (referred as $\widehat{u}$), Equation (22) can be used to prioritize compounds instead of directly using the descending order of $\widehat{y}$:

$$a = \widehat{y} - \beta\widehat{u} \quad \text{(Equation 22)}$$

where β is a user-defined parameter deciding the extent of uncertainty penalty, and a is the acquisition score. It should be noticed that the common practice is using pIC_{50} values as modeling tasks instead of the raw IC_{50} values. Compared with that of IC_{50} values, the distribution of pIC_{50} values in biological datasets is more in line with the Gaussian distribution, thus the conversion from IC_{50} to pIC_{50} can be taken as a kind of label scaling, making the prediction for both target values and uncertainties more accurate for machine-learning models.

Hie et al. valid the effectiveness of this strategy based on the task of modeling compound-kinase interaction (Hie et al., 2020). In this study, GP was used to conduct uncertainty quantification for model prediction. Compared with the predictions without uncertainty, they found that the one with UQ can prioritize interactions with lower K_d, while ignoring uncertainty will lead to higher false positive results. A retrospective virtual screening study by Soleimany et al. (Soleimany et al. (2021) also showed that filtering the results based on estimated uncertainty can increase the hit rate. In PIGNet, a DL-based drug-target interaction prediction model, MC-dropout is used to quantify the uncertainty and filter unreliable positive predictions (Moon et al., 2022). Except for considering the uncertainty in an explicit way as shown in Equation (14), some studies proposed that constructing the VS model using a BNN framework to eliminate the model uncertainty during prediction can also improve the VS model accuracy (Kim et al., 2021; Ryu et al., 2019).

Improving model accuracy and robustness

Most strategies we have introduced so far treat UQ as an independent module in the workflow of the model establishment. An important reason is that we hope to make a trade-off between model accuracy and explanation. It is less favorable to obtain model explanation at the expense of accuracy dropping. However, recent studies have shown that building models with the consideration of uncertainty may have a beneficial side effect of further improving the model accuracy. These kinds of models are called uncertainty-aware models. A typical example is MVE which has been introduced in Section 3.2. By changing the loss function, MVE is able to capture the aleatoric uncertainty inherent in data with heteroscedastic assumptions. It means that for data regions with high noise, the model can assign large uncertainty instead of overfitting them. Kwon et al. compared the MVE loss function with traditional mean squared error (MSE) loss in the task of reaction yield prediction (Kwon et al., 2022). They found that MVE loss slightly outperformed MSE loss regarding model prediction performance. Previously, we also observed a similar phenomenon in a work on building a hybrid uncertainty quantification method (Wang et al., 2021).

For regression problems, well-calibrated uncertainty can be treated as the variance of the error, thus there is an intuitive way to combine predictions and uncertainties into a more informative format, for example, the confidence interval. However, for classification problems, it is not easy to integrate these two parts together. To this end, it is essential to build an uncertainty-aware classification model architecture that could provide well-calibrated probabilities and avoid giving overconfident predictions for out-of-distribution samples. Han et al. recently proposed GNN-SNGP which can reduce overconfident misprediction by applying Gaussian Process and Spectral Normalization into model architecture (Han et al., 2021). Results on CardioTox, a cardiotoxicity dataset with a significant distribution shift, showed that GNN-SNGP can improve model accuracy and provide well-calibrated predictions. Mervin et al. presented a novel protein-ligand interaction classifier using Probabilistic Random Forest (PRF). In PRF, the original bioactivity value (e.g., $K_i = 8\mu M$) is converted to a probability (e.g., 0.63) as a label using the cumulative distribution function of a normal distribution to show how possible the compound can bind to a target. In this way, the labels are considered following as probability distribution rather than as deterministic values, and uncertainty of bioactivity labels are aleatorically introduced into model construction. Bioactivity prediction benchmarking tests showed that PRF outperformed traditional random forest regarding several common classification evaluation metrics, such as F1-score and balanced accuracy (Mervin et al., 2021b).

CONCLUSION AND PERSPECTIVE

In this review article, the background and sources of uncertainty are introduced first. Then three kinds of uncertainty quantification methods with different philosophical reasoning and four typical application scenarios where UQ is indispensable are explored in detail. We hope this content will be helpful and enlightening to readers who are not embedded in this field.

Current UQ also faces technical challenges. There is no consensus on optimal UQ methods. For different downstream tasks and task scenarios, the most appropriate UQ method is not consistent. Many UQ methods do not come as readily usable, but need to be tailored to each application scenario. Thus, designing benchmarking datasets with different degrees of domain shift is an urgent need for a fair and comprehensive comparison between different UQ methods. Different ML model architectures should also be benchmarked for the UQ methods that serve as an independent module, which will enable users to choose UQ methods more conveniently according to the specific model architecture they used in their projects. In the development process, uncertainty-aware models should be compared with conventional deep learning models without uncertainty measurements regarding accuracy and robustness to explore the potential benefits. In addition, many researches on UQ often focus on theoretical proof while ignoring practical considerations, which is one of the most concerned aspects of users. Therefore, it is highly recommended that subsequent UQ studies should summarize the differences from conventional ML models in the deployment process, and demonstrate the practicability of the proposed UQ methods with some applied case studies (e.g., virtual screening), as Soleimany et al. did in their work (Soleimany et al., 2021). Moreover, some UQ methods do not differentiate between aleatoric and epistemic uncertainty, which play different roles in the uncertainty domain. For example, aleatoric uncertainty can be used to infer model maximum achievable accuracy while epistemic uncertainty is able to guide sample selection in an AL setting. Thus, UQ methods that mix up these two types of uncertainty will be less ideal and their applications are limited. Finally, most of the UQ methods do not show evident calibration ability, especially for out-of-domain samples. Considering the ability is vital in inferring the

label range of test samples, more emphasis should be placed on the improvement of calibration ability when developing novel UQ methods.

According to the above discussion, an ideal UQ method requires the following properties: (1) supported by a solid theoretical foundation or a reasonable assumption, (2) easy to deploy, (3) disentangling aleatoric from epistemic uncertainty, (4) improvement on model accuracy, (5) possessing calibration ability, (6) low computational burden, although full compliance with these requirements may be difficult to achieve. Overall, we still have a long way to go in terms of UQ, before AI can play a more substantial role in decision making at different stages of drug development.

ACKNOWLEDGMENTS

This work was supported by the Lingang Laboratory (LG202102-01-02), the National Natural Science Foundation of China (81903639), and the Shanghai Municipal Science and Technology Major Project.

AUTHOR CONTRIBUTIONS

M.Z. directed the project. J.Y. and D.W. co-wrote the article. J.Y and D.W. contributed equally to this work.

DECLARATION OF INTERESTS

The authors declare no competing financial interest.

REFERENCES

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., et al. (2021). A review of uncertainty quantification in deep learning: techniques, applications and challenges. Inf. Fusion *76*, 243–297. https://doi.org/10.1016/j.inffus.2021.05.008.

Allen, T.E.H., Wedlake, A.J., Gelžinytė, E., Gong, C., Goodman, J.M., Gutsell, S., and Russell, P.J. (2020). Neural network activation similarity: a new measure to assist decision making in chemical toxicology. Chem. Sci. *11*, 7335–7348. https://doi.org/10.1039/d0sc01637c.

Altae-Tran, H., Ramsundar, B., Pappu, A.S., and Pande, V. (2017). Low data drug discovery with one-shot learning. ACS Cent. Sci. *3*, 283–293. https://doi.org/10.1021/acscentsci.6b00367.

Begoli, E., Bhattacharya, T., and Kusnezov, D. (2019). The need for uncertainty quantification in machine-assisted medical decision making. Nat. Mach. Intell. *1*, 20–23. https://doi.org/10.1038/s42256-018-0004-1.

Beker, W., Wołos, A., Szymkuć, S., and Grzybowski, B.A. (2020). Minimal-uncertainty prediction of general drug-likeness based on Bayesian neural networks. Nat. Mach. Intell. *2*, 457–465. https://doi.org/10.1038/s42256-020-0209-y.

Berenger, F., and Yamanishi, Y. (2019). A distance-based boolean applicability domain for classification of high throughput screening data. J. Chem. Inf. Model. *59*, 463–476. https://doi.org/10.1021/acs.jcim.8b00499.

Blei, D.M., Kucukelbir, A., and McAuliffe, J.D. (2017). Variational inference: a review for statisticians. J. Am. Stat. Assoc. *112*, 859–877. https://doi.org/10.1080/01621459.2017.1285773.

Buel, G.R., and Walters, K.J. (2022). Can AlphaFold2 predict the impact of missense mutations on structure? Nat. Struct. Mol. Biol. *29*, 1–2. https://doi.org/10.1038/s41594-021-00714-2.

Carrió, P., Pinto, M., Ecker, G., Sanz, F., and Pastor, M. (2014). Applicability domain analysis (ADAN): a robust method for assessing the reliability of drug property predictions. J. Chem. Inf. Model. *54*, 1500–1511. https://doi.org/10.1021/ci500172z.

Chen, J., Cheong, H.H., and Siu, S.W.I. (2021). xDeep-AcPEP: deep learning method for anticancer peptide activity prediction based on convolutional neural network and multitask learning. J. Chem. Inf. Model. *61*, 3789–3803. https://doi.org/10.1021/acs.jcim.1c00181.

Cortés-Ciriano, I., and Bender, A. (2016). How consistent are publicly reported cytotoxicity data? Large-Scale statistical analysis of the concordance of public independent cytotoxicity measurements. ChemMedChem *11*, 57–71. https://doi.org/10.1002/cmdc.201500424.

Deringer, V.L., Bartók, A.P., Bernstein, N., Wilkins, D.M., Ceriotti, M., and Csányi, G. (2021). Gaussian process regression for materials and molecules. Chem. Rev. *121*, 10073–10141. https://doi.org/10.1021/acs.chemrev.1c00022.

Dietterich, T.G. (2000). Ensemble methods in machine learning. Lect Notes Comput Sc *1857*, 1–15. https://doi.org/10.1007/3-540-45014-9_1.

DiFranzo, A., Sheridan, R.P., Liaw, A., and Tudor, M. (2020). Nearest neighbor Gaussian process for quantitative structure-activity relationships. J. Chem. Inf. Model. *60*, 4653–4663. https://doi.org/10.1021/acs.jcim.0c00678.

Ding, X., Cui, R., Yu, J., Liu, T., Zhu, T., Wang, D., Chang, J., Fan, Z., Liu, X., Chen, K., et al. (2021). Active learning for drug design: a case study on the plasma exposure of orally administered drugs. J. Med. Chem. *64*, 16838–16853. https://doi.org/10.1021/acs.jmedchem.1c01683.

Gal, Y., and Ghahramani, Z. (2016). Dropout as a bayesian approximation: representing model uncertainty in deep learning. Preprint at arXiv. Pr. Mach. Learn. Res. 48. https://doi.org/10.48550/arXiv.1506.02142.

Gaulton, A., Bellis, L.J., Bento, A.P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., and Overington, J.P. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Res. *40*, D1100–D1107. https://doi.org/10.1093/nar/gkr777.

Gawriljuk, V.O., Zin, P.P.K., Puhl, A.C., Zorn, K.M., Foil, D.H., Lane, T.R., Hurst, B., Tavella, T.A., Costa, F.T.M., Lakshmanane, P., et al. (2021). Machine learning models identify inhibitors of SARS-CoV-2. J. Chem. Inf. Model. *61*, 4224–4235. https://doi.org/10.1021/acs.jcim.1c00683.

Goan, E., and Fookes, C. (2020). Bayesian neural networks: an introduction and survey. Lect. Notes Math. *2259*, 45–87. https://doi.org/10.1007/978-3-030-42553-1_3.

Gong, Y., Xue, D., Chuai, G., Yu, J., and Liu, Q. (2021). DeepReac plus : deep active learning for quantitative modeling of organic chemical reactions. Chem. Sci. *12*, 14459–14472. https://doi.org/10.1039/d1sc02087k.

Graff, D.E., Shakhnovich, E.I., and Coley, C.W. (2021). Accelerating high-throughput virtual screening through molecular pool-based active learning. Chem. Sci. *12*, 7866–7881. https://doi.org/10.1039/d0sc06805e.

Han, K., Lakshminarayanan, B., and Liu, J. (2021). Reliable graph neural networks for drug discovery under distributional shift. Preprint at arXiv. e-prints, arXiv:2111.12951. https://doi.org/10.48550/arXiv.2111.12951.

Hie, B., Bryson, B.D., and Berger, B. (2020). Leveraging uncertainty in machine learning accelerates biological discovery and design. Cell

Syst. *11*, 461–477.e9. https://doi.org/10.1016/j.cels.2020.09.007.

Hirschfeld, L., Swanson, K., Yang, K., Barzilay, R., and Coley, C.W. (2020). Uncertainty quantification using neural networks for molecular property prediction. J. Chem. Inf. Model. *60*, 3770–3780. https://doi.org/10.1021/acs.jcim.0c00502.

Janet, J.P., Duan, C., Yang, T., Nandy, A., and Kulik, H.J. (2019). A quantitative uncertainty metric controls error in neural network-driven chemical discovery. Chem. Sci. *10*, 7913–7922. https://doi.org/10.1039/c9sc02298h.

Jansen, J.M., De Pascale, G., Fong, S., Lindvall, M., Moser, H.E., Pfister, K., Warne, B., and Wartchow, C. (2019). Biased complement diversity selection for effective exploration of chemical space in hit-finding campaigns. J. Chem. Inf. Model. *59*, 1709–1714. https://doi.org/10.1021/acs.jcim.9b00048.

Jaworska, J., Nikolova-Jeliazkova, N., and Aldenberg, T. (2005). QSAR applicabilty domain estimation by projection of the training set descriptor space: a review. Altern. Lab. Anim. *33*, 445–459. https://doi.org/10.1177/026119290503300508.

Jiang, J., Wang, R., and Wei, G.W. (2021). GGL-tox: geometric graph learning for toxicity prediction. J. Chem. Inf. Model. *61*, 1691–1700. https://doi.org/10.1021/acs.jcim.0c01294.

Jiménez-Luna, J., Pérez-Benito, L., Martínez-Rosell, G., Sciabola, S., Torella, R., Tresadern, G., and De Fabritiis, G. (2019). DeltaDelta neural networks for lead optimization of small molecule potency. Chem. Sci. *10*, 10911–10918. https://doi.org/10.1039/c9sc04606b.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. Nature *596*, 583–589. https://doi.org/10.1038/s41586-021-03819-2.

Kalliokoski, T., Kramer, C., Vulpetti, A., and Gedeck, P. (2013). Comparability of mixed IC50 data - a statistical analysis. PLoS One, e61007. https://doi.org/10.1371/journal.pone.0061007.

Keefer, C.E., Kauffman, G.W., and Gupta, R.R. (2013). Interpretable, probability-based confidence metric for continuous quantitative structure-activity relationship models. J. Chem. Inf. Model. *53*, 368–383. https://doi.org/10.1021/ci300554t.

Kendall, A., and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems 30 (nips 2017) *30*. https://doi.org/10.48550/arXiv.1703.04977.

Kim, Q., Ko, J.H., Kim, S., Park, N., and Jhe, W. (2021). Bayesian neural network with pretrained protein embedding enhances prediction accuracy of drug-protein interaction. Bioinformatics *37*, 3428–3435. https://doi.org/10.1093/bioinformatics/btab346.

Kimber, T.B., Gagnebin, M., and Volkamer, A. (2021). Maxsmi: maximizing molecular property prediction performance with confidence estimation using SMILES augmentation and deep learning. Artificial Intelligence in the Life Sciences *1*, 100014. https://doi.org/10.1016/j.ailsci.2021.100014.

Kiureghian, A.D., and Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? Struct. Saf. *31*, 105–112. https://doi.org/10.1016/j.strusafe.2008.06.020.

Kolmar, S.S., and Grulke, C.M. (2021). The effect of noise on the predictive limit of QSAR models. J. Chem. informatics. *13*, 92. https://doi.org/10.1186/s13321-021-00571-7.

Kramer, C., Kalliokoski, T., Gedeck, P., and Vulpetti, A. (2012). The experimental uncertainty of heterogeneous public K-i data. J. Med. Chem. *55*, 5165–5173. https://doi.org/10.1021/jm300131x.

Kwon, Y., Lee, D., Choi, Y.S., and Kang, S. (2022). Uncertainty-aware prediction of chemical reaction yields with graph neural networks. J. Chem. informatics. *14*, 2. https://doi.org/10.1186/s13321-021-00579-z.

Kwon, Y., Won, J.-H., Kim, B.J., and Paik, M.C. (2020). Uncertainty quantification using Bayesian neural networks in classification: application to biomedical image segmentation. Comput. Stat. Data Anal. *142*, 106816. https://doi.org/10.1016/j.csda.2019.106816.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. Adv. Neural Inf. Process. Syst. *30*. Nips 2017. https://doi.org/10.48550/arXiv.1612.01474.

Lazic, S.E., and Williams, D.P. (2021). Quantifying sources of uncertainty in drug discovery predictions with probabilistic models. Artificial Intelligence in the Life Sciences *1*, 100004. https://doi.org/10.1016/j.ailsci.2021.100004.

Li, Y., Rao, S., Hassaine, A., Ramakrishnan, R., Canoy, D., Salimi-Khorshidi, G., Mamouei, M., Lukasiewicz, T., and Rahimi, K. (2021). Deep Bayesian Gaussian processes for uncertainty estimation in electronic health records. Sci. Rep. *11*, 20685. https://doi.org/10.1038/s41598-021-01680-x.

Liu, R., and Wallqvist, A. (2019). Molecular similarity-based domain applicability metric efficiently identifies out-of-domain compounds. J. Chem. Inf. Model. *59*, 181–189. https://doi.org/10.1021/acs.jcim.8b00597.

Liu, R., Glover, K.P., Feasel, M.G., and Wallqvist, A. (2018). General approach to estimate error bars for quantitative structure-activity relationship predictions of molecular activity. J. Chem. Inf. Model. *58*, 1561–1575. https://doi.org/10.1021/acs.jcim.8b00114.

Mervin, L.H., Afzal, A.M., Engkvist, O., and Bender, A. (2020). Comparison of scaling methods to obtain calibrated probabilities of activity for protein-ligand predictions. J. Chem. Inf. Model. *60*, 4546–4559. https://doi.org/10.1021/acs.jcim.0c00476.

Mervin, L.H., Johansson, S., Semenova, E., Giblin, K.A., and Engkvist, O. (2021a). Uncertainty quantification in drug design. Drug Discov. Today *26*, 474–489. https://doi.org/10.1016/j.drudis.2020.11.027.

Mervin, L.H., Trapotsi, M.A., Afzal, A.M., Barrett, I.P., Bender, A., and Engkvist, O. (2021b). Probabilistic Random Forest improves bioactivity predictions close to the classification threshold by taking into account experimental uncertainty. J. Chem. informatics. *13*, 62. https://doi.org/10.1186/s13321-021-00539-7.

Moon, S., Zhung, W., Yang, S., Lim, J., and Kim, W.Y. (2022). PIGNet: a physics-informed deep learning model toward generalized drug-target interaction predictions. Chem. Sci. *13*, 3661–3673. https://doi.org/10.1039/d1sc06946b.

Musil, F., Willatt, M.J., Langovoy, M.A., and Ceriotti, M. (2019). Fast and accurate uncertainty estimation in chemical machine learning. J. Chem. Theory Comput. *15*, 906–915. https://doi.org/10.1021/acs.jctc.8b00959.

Netzeva, T.I., Worth, A., Aldenberg, T., Benigni, R., Cronin, M.T.D., Gramatica, P., Jaworska, J.S., Kahn, S., Klopman, G., Marchant, C.A., et al. (2005). Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. The report and recommendations of ECVAM Workshop 52. Altern. Lab. Anim. *33*, 155–173. https://doi.org/10.1177/026119290503300209.

Neves, B.J., Braga, R.C., Melo, C.C., Moreira, J.T., Muratov, E.N., and Andrade, C.H. (2018). QSAR-based virtual screening: advances and applications in drug discovery. Front. Pharmacol. *9*, 1275. https://doi.org/10.3389/fphar.2018.01275.

Nix, D.A., and Weigend, A.S. (1994). Estimating the mean and variance of the target probability distribution. 1994 Ieee International Conference on Neural Networks, *1–7*, pp. 55–60. https://doi.org/10.1109/ICNN.1994.374138.

Olivier, A., Shields, M.D., and Graham-Brady, L. (2021). Bayesian neural networks for uncertainty quantification in data-driven materials modeling. Comput. Method. Appl. M. *386*. https://doi.org/10.1016/j.cma.2021.114079.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z.M., Gimelshein, N., Antiga, L., et al. (2019). PyTorch: an imperative style, high-performance deep learning library. Proceedings of the 33rd International Conference on Neural Information Processing Systems, *721*, pp. 8026–8037.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: machine learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B.B., Chen, X., and Wang, X. (2020). A survey of deep active learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.2009.00236.

Ryu, S., Kwon, Y., and Kim, W.Y. (2019). A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. Chem. Sci. *10*, 8438–8446. https://doi.org/10.1039/c9sc01992h.

Sahigara, F., Ballabio, D., Todeschini, R., and Consonni, V. (2013). Defining a novel k-nearest neighbours approach to assess the applicability domain of a QSAR model for reliable predictions.

J. Chem. informatics. *5*. https://doi.org/10.1186/1758-2946-5-27.

Saxe, A., Nelli, S., and Summerfield, C. (2021). If deep learning is the answer, what is the question? Nat. Rev. Neurosci. *22*, 55–67. https://doi.org/10.1038/s41583-020-00395-8.

Scalia, G., Grambow, C.A., Pernici, B., Li, Y.P., and Green, W.H. (2020). Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. J. Chem. Inf. Model. *60*, 2697–2717. https://doi.org/10.1021/acs.jcim.9b00975.

Shannon, C.E. (1948). A mathematical theory of communication. Bell Syst Tech J *27*, 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Sheridan, R.P. (2012). Three useful dimensions for domain applicability in QSAR models using random forest. J. Chem. Inf. Model. *52*, 814–823. https://doi.org/10.1021/ci300004n.

Sheridan, R.P. (2013). Using random forest to model the domain applicability of another random forest model. J. Chem. Inf. Model. *53*, 2837–2850. https://doi.org/10.1021/ci400482e.

Sheridan, R.P. (2015). The relative importance of domain applicability metrics for estimating prediction errors in QSAR varies with training set diversity. J. Chem. Inf. Model. *55*, 1098–1107. https://doi.org/10.1021/acs.jcim.5b00110.

Sheridan, R.P., Feuston, B.P., Maiorov, V.N., and Kearsley, S.K. (2004). Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR. J. Chem. Inf. Comput. Sci. *44*, 1912–1928. https://doi.org/10.1021/ci049782w.

Sheridan, R.P., Karnachi, P., Tudor, M., Xu, Y., Liaw, A., Shah, F., Cheng, A.C., Joshi, E., Glick, M., and Alvarez, J. (2020). Experimental error, kurtosis, activity cliffs, and methodology: what limits the predictivity of quantitative structure-activity relationship models? J. Chem. Inf. Model. *60*, 1969–1982. https://doi.org/10.1021/acs.jcim.9b01067.

Shoichet, B.K. (2004). Virtual screening of chemical libraries. Nature *432*, 862–865. https://doi.org/10.1038/nature03197.

Smith, L., and Gal, Y. (2018). Understanding measures of uncertainty for adversarial example detection. Uncertainty in artificial intelligence. Preprint at arXiv, 560–569. https://doi.org/10.48550/arXiv.1803.08533.

Soleimany, A.P., Amini, A., Goldman, S., Rus, D., Bhatia, S.N., and Coley, C.W. (2021). Evidential deep learning for guided molecular property prediction and discovery. ACS Cent. Sci. *7*, 1356–1367. https://doi.org/10.1021/acscentsci.1c00546.

Sterling, T., and Irwin, J.J. (2015). ZINC 15-ligand discovery for everyone. J. Chem. Inf. Model. *55*, 2324–2337. https://doi.org/10.1021/acs.jcim.5b00559.

Tagasovska, N., and Lopez-Paz, D. (2019). Single-model uncertainties for deep learning. Preprint at arXiv *32*. Adv. Neur. In.. https://doi.org/10.48550/arXiv.1811.00908

Tynes, M., Gao, W., Burrill, D.J., Batista, E.R., Perez, D., Yang, P., and Lubbers, N. (2021). Pairwise difference regression: a machine learning meta-algorithm for improved prediction and uncertainty quantification in chemical search. J. Chem. Inf. Model. *61*, 3846–3857. https://doi.org/10.1021/acs.jcim.1c00670.

Wang, D.Y., Yu, J., Chen, L.F., Li, X.T., Jiang, H.L., Chen, K.X., Zheng, M.Y., and Luo, X.M. (2021). A hybrid framework for improving uncertainty quantification in deep learning-based QSAR regression modeling. J. Chem. informatics. *13*. https://doi.org/10.1186/s13321-021-00551-x.

Williams, C.K.I., and Rasmussen, C.E. (1996). Gaussian processes for regression. Adv. Neural Inf. Process. Syst. *8*, 514–520.

Xiang, Y., Tang, Y.H., Lin, G., and Sun, H. (2021). A comparative study of marginalized graph kernel and message-passing neural network. J. Chem. Inf. Model. *61*, 5414–5424. https://doi.org/10.1021/acs.jcim.1c01118.

Yang, Y., Yao, K., Repasky, M.P., Leswing, K., Abel, R., Shoichet, B.K., and Jerome, S.V. (2021). Efficient exploration of chemical space with docking and deep learning. J. Chem. Theory Comput. *17*, 7106–7119. https://doi.org/10.1021/acs.jctc.1c00810.

Yildirim, B., and Cole, J.M. (2021). Bayesian particle instance segmentation for electron microscopy image quantification. J. Chem. Inf. Model. *61*, 1136–1149. https://doi.org/10.1021/acs.jcim.0c01455.

Yu, J., Li, X., and Zheng, M. (2021). Current status of active learning for drug discovery. Artif. Intell. Life Sci. *1*, 100023. https://doi.org/10.1016/j.ailsci.2021.100023.

Zhang, Y., and Lee, A.A. (2019). Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. Chem. Sci. *10*, 8154–8163. https://doi.org/10.1039/c9sc00616h.