ACS OMEGA

Open Access

http://pubs.acs.org/journal/acsodf

Article

Development and Evaluation of Conformal Prediction Methods for Quantitative Structure−Activity Relationship

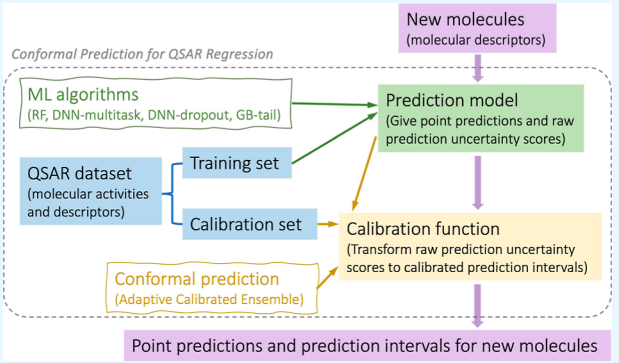
Yuting Xu,* Andy Liaw, Robert P. Sheridan, and Vladimir Svetnik

Cite This: *ACS Omega* 2024, 9, 29478−29490

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The quantitative structure−activity relationship (QSAR) regression model is a commonly used technique for predicting the biological activities of compounds using their molecular descriptors. Besides accurate activity estimation, obtaining a prediction uncertainty metric like a prediction interval is highly desirable. Quantifying prediction uncertainty is an active research area in statistical and machine learning (ML), but the implementation for QSAR remains challenging. However, most ML algorithms with high predictive performance require add-on companions for estimating the uncertainty of their prediction. Conformal prediction (CP) is a promising approach as its main components are agnostic to the prediction modes, and it produces valid prediction intervals under weak assumptions on the data distribution. We proposed computationally efficient CP algorithms tailored to the most widely used ML models, including random forests, deep neural networks, and gradient boosting. The algorithms use a novel approach to the derivation of nonconformity scores from the estimates of prediction uncertainty generated by the ensembles of point predictions. The validity and efficiency of proposed algorithms are demonstrated on a diverse collection of QSAR data sets as well as simulation studies. The provided software implementing our algorithms can be used as stand-alone or easily incorporated into other ML software packages for QSAR modeling.



INTRODUCTION

Quantitative structure−activity relationship (QSAR) regression models are routinely applied in the drug discovery process for predicting the biological activities of molecules from the molecular structure-based features. The predictions are used to prioritize candidate molecules for future experiments and help chemists gain better understanding of how structural changes affect activities.[1−3] While most of the previous efforts focus on improving the accuracy of point predictions, quantifying the uncertainty in the prediction will add valuable insights.[4−8] For regression tasks, prediction intervals (PIs) are often used as quantitative measures of the reliability or confidence in the point prediction at a given probability. A well-calibrated PI contains future observations with a prespecified probability, which is called nominal coverage. The width of a calibrated PI gives users an intuitive estimate of the precision of the prediction, with a wider interval indicating less precision (more uncertainty) than a narrower interval. For example, an interval of 2.0 to 3.0 at 95% probability for compound A means that the true value of the activity has a 95% chance of falling within 2.0 and 3.0. In comparison, a wider interval of 1.0 to 4.0 at 95% for compound B would indicate that B is less reliably predicted than A.

There are various methods for estimating prediction uncertainties. Some methods directly estimate the quantile or variance of prediction errors, while others provide relative uncertainty scores between different molecules, which requires further calibration for obtaining valid PIs.[6] Quantile regression methods[9−11] have been explored with various machine learning (ML) algorithms, including neural networks,[12] random forests (RFs),[13] and light gradient boosting (GB) machine;[14] however, the PIs constructed from the quantile estimates often lead to an under-coverage, i.e., the fraction of true activities falling within the intervals is smaller than the specified nominal coverage. Bayesian methods estimate the posterior distribution of molecular activity given a molecular structure. Thus, the PIs could be computed using the variance of the prediction errors under certain assumption of their distribution. An example of this approach is studied in Feng et al.,[13] the Bayesian additive regression trees that simultaneously estimate molecular activity and the error variance as a measure of prediction uncertainty, under the assumption that the errors are normally distributed. This assumption, however, is often violated in many QSAR applications.[15−17] A practical concern with some methods is the

Received: February 29, 2024
Revised: June 10, 2024
Accepted: June 12, 2024
Published: June 27, 2024

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

high computational cost, which is especially high for Bayesian methods. Hirschfeld et al.[6] benchmarked several uncertainty quantification methods that are applicable to neural networks, including ensemble-based methods,[18,19] distance-based methods, mean-variance estimation,[20] and union-based methods.[21] This work recommended several top-performing methods that produced an estimate of the predicted error variances with the lowest negative log likelihood under the Gaussian distribution assumption for prediction errors. The likelihood-based evaluation metric, however, is highly sensitive to the distributional assumption. As was mentioned above, prediction errors in many QSAR applications are non-Gaussian.[15−17] The top recommended methods in Hirschfeld et al.[6] are union-based methods involving training a neural network for point prediction, and a second model, for example, the Gaussian Process or RF regression to estimate the variance of errors in prediction produced by the first model. Training of two models typically requires substantially more data. Comprehensive reviews on uncertainty quantification methods and/or applications could be found in Mervin et al.,[7] Abdar et al.,[22] Tian et al.,[23] and Yu et al.[8] And some additional recent work in this field could be found in Alvarsson et al.,[24] Bostrom et al.,[25,26] and McShane et al.[27]

We are seeking practical methods to provide PIs accompanying the point predictions for large QSAR data sets in an industrial-scale pharmaceutical drug discovery environment. A good algorithm should meet the following requirements:

(1) Marginal validity: The empirical marginal coverage for a test set is the proportion of PIs that contains these molecules' true activity measurements. Validity means that the empirical coverage must match the user-specified (nominal) coverage. Marginal, in this context, is understood as averaged over all molecules not used in model development.

(2) Conditional validity (also called informativeness or adaptivity): The conditional coverage, i.e., PI coverage for each molecule, should be close to the nominal coverage. In other words, the width of the PI should closely follow the uncertainty in the prediction of any specific molecule. Under the homoscedastic conditions where the variance of prediction errors does not depend on the molecular descriptors, PIs for all molecules will have the same width. In contrast, under heteroscedastic conditions, where the variance depends on the descriptors, the width of the PI should increase or decrease in accordance with the variance. Compared to marginal validity, conditional validity is a much more stringent requirement usually very difficult to fully achieve. However, we still aim to have PIs with widths adaptive to the prediction errors, in order to approximate this property.

(3) Efficiency or tightness of PIs: The width of the PIs should be as narrow as possible.

(4) Low computational cost: The computation of PIs should not involve a significant computational effort over generating the original point prediction model. In real application, it is crucial to ensure the computational efficiency, since the QSAR models need to be rebuilt regularly with accumulated experimental data.

(5) Independence of model or data distribution: The properties of the PIs, such as validity, efficiency, and adaptivity, do not depend on the choice of ML methods or any assumption about the distribution of prediction errors.

Many evaluation metrics of PIs have been explored in previous benchmark studies, and several recommendations have been proposed.[6,23,28,29] Although it is difficult to find a single method that is superior to others under all criteria, the conformal prediction (CP) framework could satisfy most of these practical considerations and has attracted increased attention in recent QSAR applications.[18,19,30−35] The PIs generated by CP methods are guaranteed to achieve valid marginal coverage, i.e., on average, the test set PIs to contain the measured molecular activities with a user-specified probability, under the assumption of data exchangeability, i.e., that the training set molecules and molecules being predicted are random selections from the same pool of molecules.[36,37] This marginal validity property does not depend on any distributional assumptions or model assumptions and holds for any sample size.[36] The CP is applied as a companion to any pretrained model for a QSAR regression task, and many conformal algorithms require little extra computational costs besides the training of the original prediction model. Construction of conformal PIs is based on a "nonconformity" score that quantifies how unusual a data point is relative to the training data.[38] We prefer to use nonconformity scores that do not require additional modeling efforts. The adaptivity property of PIs depends on the choice of a nonconformity score. A theory for selecting the nonconformity score is yet to be developed, and the choice needs to be evaluated with empirical studies. Several computationally efficient CP algorithms have been developed for QSAR regression;[18,19,34] however, a comprehensive evaluation of their properties, especially their adaptivity, i.e., ability to handle heteroscedasticity of prediction errors, remains to be done.

In this work, our first contribution is to develop a computationally efficient CP algorithm, named adaptive calibrated ensemble (ACE) that is based on a specifically designed nonconformity score. The ACE algorithm can be used with any model that provides both point prediction and an uncertainty estimate for each molecule. For any ensemble-based model, the mean prediction over the ensemble is the "prediction" (i.e., the point estimate) and the standard deviation of predictions can be used as the prediction uncertainty estimate. The point estimate and prediction uncertainty estimate are inputs for ACE. The ACE nonconformity score is computed from the point prediction and the data-driven transformation of the prediction uncertainty score. If the uncertainty scores effectively represent the relative uncertainty between molecules, the ACE PIs would achieve not only a prespecified marginal coverage but also an approximate conditional coverage.

Deep neural networks (DNNs) and GB models are among the most predictive descriptor-based ML methods.[39−41] Our second contribution is proposing novel approaches for generating uncertainty estimates for these methods, named DNN-multitask and GB-tail, respectively. DNN-multitask is compared to the state-of-the-art DNN-dropout method for generating uncertainty estimates from ensembles.[8,42]

Our third contribution is a comprehensive analysis of a diverse collection of real QSAR data sets, with special attention to the analysis of conditional validity and efficiency.

Conditional validity and adaptivity of PIs in the heteroscedastic case is one of our work's primary objectives and that of other researchers. The proposed methods include modifying the nonconformity scores by normalizing them to the conditional

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

measures of prediction uncertainty. These methods and relevant references are discussed later. The method proposed in our paper also uses conformity score modification in a novel way, efficiently handling heteroscedasticity that is demonstrated with a large volume of diverse QSAR data. Different approaches to handling heteroscedasticity or, in general, obtaining PIs satisfying conditional validity have been proposed recently and are based on quantile regression,[43] conditional histograms,[44] and CP distributions.[45,46] To our knowledge, these approaches have yet to be used in the QSAR literature and are currently being investigated by the authors of this paper.

The paper is organized as follows. The Data Sets and Molecular Descriptors section describes QSAR data sets and their molecular descriptors. In the Methods section, we briefly review the CP framework and applications to QSAR, introduce the proposed ACE CP algorithm, define evaluation metrics, and explain in detail how to obtain ensemble predictions or raw prediction uncertainty scores for several ML algorithms. The Results section presents the results of applying ACE jointly with several popular QSAR predictive algorithms and shows the validity, efficiency, and adaptivity of proposed methods on diverse QSAR data sets.

DATA SETS AND MOLECULAR DESCRIPTORS

Two collections of QSAR data sets are used in this study:

- ChEMBL: IC_{50} data sets for 23 diverse protein targets and receptors from the ChEMBL database. The data sets were obtained from Cortes-Ciriano et al.,[18] excluding the smallest data set "A2a" with only 203 molecules. We generated molecular descriptors according to the procedures described in Cortes-Ciriano et al.[19] The circular Morgan fingerprints[47] were computed using RDkit[48] (version 2021.03.2) with the radius set to 2, and the output fingerprint length was 2048.
- Kaggle: The 15 QSAR data sets used in the 2012 "Merck Molecular Activity Challenge" Kaggle competition and released in Ma et al.[39] These data sets are of various sizes for either on-target potency, off-target activity, or absorption, distribution, metabolism, and excretion (ADME) properties. The molecular descriptor is the union of the "atom pair" (AP) descriptors from Carhart et al.[49] and "donor−acceptor pair" (DP) descriptors.[50] Each data set was provided in two parts: the time-split training set and test set. In this work, we only use the training set in each data set, since investigating the covariate-shift problem for time-split test set is out of scope for this study.

Compared to the ChEMBL collection with all IC_{50} data sets, the Kaggle collection contains a mixture of tasks (including on-target potency or off-target absorption, distribution, metabolism, and excretion activities), larger data sets with higher dimension of molecular features, and examples of activity distributions far from Gaussian.

Due to the complex nature of QSAR data sets, the distribution of prediction errors is often unknown. Some data sets may have near-constant prediction error variance that is unrelated to molecular features. In this case, applying constant width PIs for all molecules (i.e., homoscedastic) is preferred. Other data sets may have varying prediction error variance for different molecules, for example, variability of experimental measurements increases with their magnitude. In this case, PIs with varying widths correlated with prediction errors (i.e., heteroscedastic) are desirable. A good algorithm should provide adaptive PIs suitable for both cases. The data sets we use here are likely to contain both situations.

METHODS

Conformal Prediction. CP is a model-agnostic framework for measuring prediction uncertainty.[36,38] In regression tasks, the PIs constructed by CP achieve valid marginal coverage with the general i.i.d. (independent and identically distributed) data assumption.[37,51] The original CP algorithm, which is called transductive CP, is computationally expensive since it requires training a new predictive model for every additional molecule. For large data sets in QSAR problems, a modified version of the CP algorithm called inductive CP (ICP) or split CP has been used.[18,19,33,34,52] The steps involved in ICP are as follows:

(1) Specify an ML method to generate a model that provides point predictions.
(2) Define a nonconformity measure to quantify how unusual the prediction error of a molecule is compared to the others. The nonconformity measure may depend on the prediction $\hat{y}$ or the descriptor vector $\mathbf{X}$ of a molecule.
   - For constructing homoscedastic PIs, which have a constant width for all molecules, the nonconformity measure is defined as the absolute value of residuals
   $$\alpha = |y - \hat{y}|$$
   - For constructing heteroscedastic PIs that apply to individual molecules, the nonconformity measure can be defined as the normalized absolute value of residuals
   $$\alpha = \frac{|y - \hat{y}|}{\sigma(\mathbf{X})} \quad (1)$$
   where $\sigma(\mathbf{X})$ is an estimate of the precision of the point predictor.[53] The homoscedastic nonconformity measure is corresponding to a special case when $\sigma(\mathbf{X}) \equiv 1$.
(3) Divide the available training data randomly into a "proper training set" and a "calibration set". For QSAR tasks, the relationship between molecular structure descriptors and activities is usually highly complex. It is preferable to use a large fraction of data, e.g., around 80%, as the "proper training set", in order to train an accurate point predictor model.
(4) Use the proper training set to construct a model, and use the model to predict the activity of molecules in the calibration set. Calculate the nonconformity score α_j for each calibration set molecule x_j $(1 \le j \le n)$.
(5) Specify a desired nominal coverage probability θ, and calculate the θ^{th} percentile of nonconformity scores in the calibration set as α_θ.
(6) The PI for a new molecule $\mathbf{x}_{\text{new}}$ in the test set would be
$$\hat{y}(\boldsymbol{x}_{\text{new}}) \pm \alpha_\theta \sigma(\boldsymbol{x}_{\text{new}})$$

The choice of the scaling factor $\sigma(\mathbf{X})$ in nonconformity measure affects the efficiency of CP (i.e., the width of PIs).[38] There are two typical approaches to define the scaling factor used in previous studies: model-based[33,43,52,53] and ensemble-based approach.[18,19,34] In this paper, we will use the ensemble-based approach. Here, $\sigma(\mathbf{X})$ is a function of the ensemble prediction uncertainty $s(\mathbf{X})$, which is the standard deviation of

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

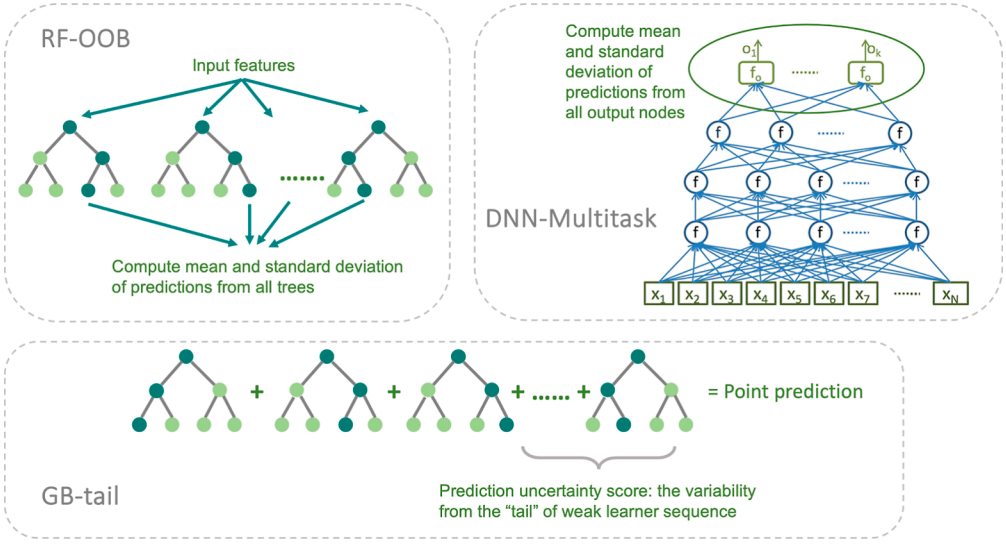


Figure 1. Illustration of generating point predictions and prediction uncertainty scores using various ML algorithms.

predictions from an ensemble of predictions. Below, we abbreviate $\sigma(X)$ and $s(X)$ as σ and s , respectively, when they are clearly dependent on molecular descriptors X. Ensemble-based scaling factors performed slightly better than the model-based scaling factors in the previous work on QSAR data sets.[34] Some ML methods like RFs natively produce an ensemble of predictions. Other popular ML methods such as DNN or boosting need modification to produce an ensemble of predictions, preferably without being too computationally expensive. This is discussed in the ML Algorithms section.

ACE Conformal Predictor. We need to transform s, the ensemble prediction uncertainty, to the scaling factor σ in eq 1. The exponential transformation of s, i.e., $\sigma = e^s$, has been a popular choice of the scaling factor in previous studies.[18,19,34,54] We call this scaling method "expSD". Svensson et al.[34] explored different transformations $\sigma = e^{\gamma \times s}$ with various weight γ ranging from 0.05 to 1.25 and concluded that the best average performance was obtained when $\gamma = 1$. However, in our experience, this scaling method is not always optimal for different data sets or ensemble approaches. The range of scaling factor σ affects the range of PI widths in a data set. For ensemble methods that produce smaller s, the exponential transformation will lead to near-constant scaling factor σ and thus almost constant-width PIs, which may not be desirable if the data set is intrinsically heteroscedastic. On the other hand, it may also lead to excessively wide and noninformative PIs for some molecules.

Therefore, we proposed a flexible calibration algorithm, ACE, which will generate transformation of any raw prediction uncertainty score s as a scaling factor for nonconformity scores adaptive to different data sets and/or ensemble models. The score s can represent the variability of predictions in an ensemble of models, or any relative uncertainty scores that correlate with the prediction errors. The ACE algorithm finds the optimal transformation by minimizing the coverage error conditional on the PI width via repeated cross-validation on the calibration set. The major steps in ACE are as follows:

(1) Calculate the mean μ_s and standard deviation sd_s of the s values on the calibration set, and calculate the normalized s as $\tilde{s} = \frac{s-\mu_s}{sd_s}$.

(2) Define b as the average of the absolute error in the calibration set. And define the scaling factor σ as a function of the parameter a

$$\sigma(a; \tilde{s}) := a^* \tanh(\tilde{s}) + b$$

where $0 \le a \le b$ to ensure that $\sigma(a)$ is always a positive and nondecreasing function of $\tilde{s}$.

(3) Perform repeated twofold cross-validation on the calibration set, and use grid-search to find the optimum value of a to achieve the lowest average coverage error over four equal size subgroups defined by PI widths.

More details are provided in the ML Algorithms section below.

Algorithm 1 Adaptive Calibrated Ensemble (ACE)

1: Calculate $\tilde{s} = \frac{s-\mu_s}{sd_s}$, where μ_s and sd_s are the mean and standard deviation of the s values on calibration set.
2: Define b as the average of absolute error in calibration set.
3: Define the scaling factor σ as a function of parameter a:

$$\sigma(a; \tilde{s}) := a * \tanh(\tilde{s}) + b, \quad 0 \le a \le b$$

4: Generate a list of candidate values for parameter a, denoted A, e.g. $A = \{0, b/100, 2 * b/100, \ldots, b\}$.
5: **for** each $a \in A$ **do**
6: **for** $repeat = 1, 2, \ldots, R$ **do**
7: Randomly split the calibration into two halves, denoted as set C_1, C_2, which will be used as calibration set and test set in repeated cross-validation
8: Use C_1 and scaling factor $\sigma(a)$ to construct conformal prediction intervals for C_2 under a specified normal level
9: Compute the average PI width $w(a; repeat)$
10: Divide set C_2 into four equal-sized subsets according to the prediction interval widths
11: Compute the average of absolute coverage error on the four subsets in C_2, denoted as $\eta(a; repeat)$
12: **end for**
13: Compute $\eta_m(a)$ and $\eta_s(a)$ as the average and standard deviation of $\eta(a; repeat)$ over R repeats
14: Compute $w_m(a)$ and $w_s(a)$ as the average and standard deviation of $w(a; repeat)$ over R repeats
15: **end for**
16: Let $S_1 := \{a \in A : \eta_m(a) \le min(\eta_m(a)) + \eta_s(\arg\min_a(\eta_m(a)))/\sqrt{R}\}$
17: Let $S_2 := \{a \in S_1 : w_m(a) \le min(w_m(a)) + w_s(\arg\min_a(w_m(a)))/\sqrt{R}\}$
18: Compute $a_{opt} = median(\{a : a \in S_2\})$
19: Return the optimized scaling factor $\sigma(\tilde{s}) := a_{opt} * \tanh(\tilde{s}) + b$

ML Algorithms. In this section, we describe the supervised learning algorithms commonly used in QSAR applications and how to generate computationally efficient prediction uncertainty score s, as illustrated in Figure 1. Additional details of implementation and hyperparameter settings of each algorithm are provided in the Supporting Information.

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

Random Forests. RF has been a very popular QSAR method due to its high prediction accuracy and robustness to the choice of hyperparameters.[55] RF is an ensemble of independent decision trees. The average of individual tree predictions is the final prediction for a molecule; and $s(\mathbf{X})$ is the standard deviation of tree predictions of the molecule represented by $\mathbf{X}$. This has been an effective measure of prediction uncertainty in previous conformal applications.[18,19,34]

Compared to other supervised learning methods, RF has a unique appealing feature: the out-of-bag (OOB) data already represents a random split, and one does not need to make an explicit "proper training set/calibration set" split.[53] This prediction uncertainty estimation strategy is referred to as "RF-OOB" in the later sections.

Deep Neural Networks. DNN is a practical QSAR method for large data sets that usually achieve superior predictive performance,[39] although it is somewhat computationally expensive, sometimes requiring GPU computing hardware. Here, we consider only fully connected descriptor-based DNNs.

We considered two approaches for generating ensemble predictions that require training only one DNN prediction model: the "DNN-dropout" as it is called in previous reports[19,42] and a novel "DNN-multitask" method.

DNN-dropout: Cortes-Ciriano and Bender[19] built conformal predictors for the ChEMBL data sets using test-time dropout[42] ensembles, where the ensemble prediction was created by allowing random dropout of DNN hidden layer nodes and repeating the prediction process 100 times. The average and standard deviation across the 100 repeated predictions for a molecule were used as its predicted value and a measurement of the prediction uncertainty, respectively. We adopted the same DNN structure and optimization algorithm described by Cortes-Ciriano and Bender[19] for the ChEMBL data sets. In the study by Cortes-Ciriano and Bender,[19] multiple dropout rates, 0.1, 0.25, and 0.5, were explored, and the results demonstrated that the performance for different dropout rates was comparable. We adopted the median value of the dropout rate, i.e., using a constant dropout rate of 0.25 in all hidden layers. For Kaggle data sets, we used the recommended parameter setting from Ma et al.[39]

DNN-multitask: We created an artificial sparse multitask DNN model from a single-task QSAR data set, by replicating the molecular activities K times with random omissions of molecules (with the probability of omission p), for example

$$\tilde{y}_i = (y_i, y_i, \text{NA}, y_i, \text{NA}, \ldots), \forall\, i$$

where the artificial vector outcome $\tilde{y}_i$ for each molecule i is a K-dimensional vector with elements $\tilde{y}_{ij}$, $1 \le j \le K$

$$\Pr(\tilde{y}_{ij} = y_i) = 1 - p, \qquad \Pr(\tilde{y}_{ij} = \text{NA}) = p$$

"NA" represents an omitted molecular activity. The average of multitask outputs is taken as the point prediction. The standard deviation of multitask predictions for each molecule is used as the prediction uncertainty metric $s(X)$. The intuition is that molecules that are "easier" to predict would get more consistent predictions across this pseudoensemble model. To explore whether the multitask standard deviation is sensitive to the choice of hyperparameters K and p, we evaluated multiple combinations of (K, p) pairs via simulations (see the Supporting Information) and recommend using a higher omission probability ($p = 0.6$) and a moderate number of the output nodes ($K = 20$ or 50).

For Kaggle data sets, the neural network structure and optimization algorithm are the same as Ma et al.,[39] except for the output layer size. For the ChEMBL data sets, since the dimensions of input molecular features are smaller and the molecules are "easier" to predict, we used a smaller neural network structure, which achieved similar prediction accuracy with less computational cost compared to the DNN-dropout setting in the study by Cortes-Ciriano and Bender.[19]

Gradient Boosting. GB is a widely used QSAR method due to its computational efficiency and accuracy.[40,41] A GB model consists of a series of many (typically 1000 or more) shallow decision trees. The prediction of the model on a molecule is the sum of predictions of the decision trees on that molecule. The trees are added to the model in iterations such that the errors from the current model are used to grow a new tree, so that the new tree is forced to learn information that the current model did not yet learn from the data. Thus, in general, we expect the trees learned earlier in the sequence to have larger contributions to the overall prediction than those later in the sequence. For prediction of a new molecule, if the quality of the prediction is high, we would expect that the absolute magnitude of the contribution would diminish for later trees. If the absolute contribution falls to nearly zero after the first few trees, it indicates that the prediction for that molecule is reliable. On the other hand, if the absolute contributions from the last few trees are still large, then the prediction is less likely to be reliable. Therefore, we propose the "GB-tail" method: for each molecule, let s be the mean absolute value of the contributions from the final w fraction of trees (for instance, for $w = 0.2$, the last 200 out of 1000 trees). This approach requires only postprocessing of tree predictions and has a minimal computational cost.

To check whether the estimate of the prediction uncertainty by the GB-tail method is sensitive to w, we investigated the impact of varying the hyperparameter w (Supporting Information) and recommend $w = 0.2$.

Evaluation Metrics. We compared three types of conformal PIs: Homoscedastic PIs (abbreviated as "homo" in the figures), heteroscedastic PIs with the "expSD" scaling factor (expSD), and heteroscedastic PIs with the scaling factor computed from the proposed ACE algorithm (ACE). They are applied in combination with four supervised learning algorithms/ensemble schemes introduced in the ML Algorithms section.

We evaluated several important aspects of the prediction models, including the accuracy of point predictions, computational costs, and the informativeness of PIs. All the evaluation metrics are calculated on the test set and averaged across multiple random repeats and/or data sets.

The prediction accuracy is measured by the squared Pearson correlation coefficient (R-squared) and the normalized root-mean-squared error (RMSE). These metrics serve the purpose of assessing the quality of predictions generated by different ML methods. It is essential to ensure satisfactory prediction accuracy before evaluating the performance of conformal PIs.

The performance of a conformal predictor is usually evaluated in terms of validity and efficiency. A valid PI has a coverage probability no less than the prespecified nominal coverage level, i.e., if the probability is 90%, then at least 90% of the true activity measurements should fall within the intervals. The efficiency is measured by the widths of the PIs. In regression analysis, a conformal predictor is considered favorable if it produces narrow PIs around the point predictions while maintaining the desired nominal coverage. ICP algorithms provide PIs that guarantee marginal validity regardless of the underlying data

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

Table 1. Summary of Test Set Predictive Performance and Run Time (in Seconds) of Each ML Method for ChEMBL and Kaggle Data Sets[a]

	ChEMBL			Kaggle		
ML prediction algorithm	*R*-squared	RMSE	runtime (s)	*R*-squared	RMSE	runtime (s)
RF	0.633 (0.115)	0.710 (0.086)	41.0	0.683 (0.084)	0.565 (0.073)	1314.3
DNN-dropout	0.652 (0.112)	0.688 (0.085)	127.8	0.703 (0.106)	0.541 (0.098)	893.4
DNN-multitask	0.645 (0.113)	0.697 (0.088)	36.2	0.700 (0.108)	0.547 (0.102)	781.3
GB	0.644 (0.114)	0.694 (0.087)	8.4	0.706 (0.098)	0.536 (0.089)	143.8

[a]These numbers represent an average over the data sets and repeats. The standard deviation for the *R*-squared and RMSE is provided in brackets. The RMSE of Kaggle data sets is scaled by the standard deviation of molecular activities.

distribution. However, the efficiency of these PIs can be influenced by the choice of nonconformity scores.[38,51] The selection of an appropriate nonconformity score is crucial in achieving optimal efficiency.

Coverage error: For each nominal coverage probability, we calculated the coverage error as the difference between the actual coverage of test set compounds and the nominal coverage. For example, if the nominal coverage of the interval is 95% and the actual coverage of the interval is 85%, the difference is 10%. Although the conformal approach theoretically guarantees that the coverage is correct, in practice, coverages of finite samples will fluctuate around the nominal value.[36] We also calculated the mean absolute error (MAE) of the test set coverage across different data sets and repeats.

Efficiency: To compare the efficiency across algorithms and data sets, one needs to normalize the raw PIs. One way to do this is to find the ratio of the width of the raw interval to the width of the interval in the corresponding "no-model" case. The no-model case uses the distribution of observed training set activities rather than predictions. For example, for a nominal coverage of 90%, the interval is between the fifth-percentile and 95th-percentile values of the observed activities in the training set. The "no-model" PI should be wider than the PIs obtained by the ICP analysis where the model-based point predictions are used. When comparing the efficiency between heteroscedastic and homoscedastic PIs, we scaled the heteroscedastic PI widths by the corresponding homoscedastic PI width for the same test set and the same point prediction model.

Adaptability: The ability to differentiate easy-to-predict and difficult-to-predict molecules, i.e., "adaptability", is another top consideration for constructing informative PIs.[56−60] The expectation is that molecules that can be more accurately predicted have narrower PIs. Ideally, one would like the PI to cover the true activity with the prespecified nominal coverage probability for every molecule, i.e., to achieve conditional validity. However, this cannot be achieved without imposing assumptions on the data distribution.[58] Although there is no theoretical guarantee for ICP algorithms to satisfy the conditional validity, one may approximate this property with heteroscedastic PIs constructed from well-designed nonconformity scores. We used a size-stratified coverage metric to evaluate the adaptivity of PIs in the following way. First, we sorted the test set molecules by their PI widths, and divided them into five equal-size subgroups. We then calculated the coverage error within each subgroup, as well as the average absolute error of coverages across five subgroups for each data set. If the coverage in these subgroups significantly deviates from the nominal coverage, it indicates a lower quality of the conditional coverage.[36,56,57,61]

In the Supporting Information section, we use simulated data with Gaussian-distributed noise that have either constant variance or changing variance to demonstrate how the state-of-the art expSD and the proposed ACE CP algorithms performed in these cases. Based on the simulation study, the expSD method cannot provide PIs adaptive to the magnitude of errors and does not perform well for all ML algorithms. When applying expSD with the DNN-multitask and GB-tail ML method, the PIs have near-constant widths and are not adaptive to the errors from different molecules. The reason is that the raw prediction uncertainty scores from DNN-multitask and GB-tail span a much narrower range and it leads to close to constant scaling factor σ in the expSD method. And the results from expSD with RF-OOB or DNN-dropout methods are also unsatisfactory: For data simulated with constant variance, they have larger coverage errors in size-stratified subgroups than those from either homo or ACE and are very likely to produce some PIs that are too wide to be useful. Thus, we decided to exclude the expSD CP algorithm from any further comparisons in the application section.

RESULTS: APPLICATION TO CHEMBL AND KAGGLE DATA SETS

In all the following numerical experiments, each data set is randomly split into a proper training set (70% of the data) for training a prediction model for molecular activity and generating raw uncertainty scores; a calibration set (15%) for CP; and a test set (15%) for evaluation. The data split is repeated 20 times, and the same splits are used for all algorithms. For RF models, since we use the OOB data for CP instead of a separate calibration set, the union of the proper training set and calibration (85% of the data) is used for RF model training, and the same 15% test set is used for evaluation.

The average predictive performance and computational time over repeats for two groups of data sets are listed in Table 1. All models achieved satisfactory prediction accuracy. The two DNN models and the GB model perform slightly better than RF, which is usually used as the baseline method in QSAR modeling, despite that the RF model uses more training data (85% of an entire data set) compared to others (70% of an entire data set). Also, the DNN-multitask and GB models took less time compared to DNN-dropout and RF. While the accuracy and run-time of point predictors may not directly affect the performance of conformal predictors, these comparisons demonstrated the necessity of developing suitable uncertainty quantification methods tailored for various practical QSAR models, in addition to the widely used RF-based conformal predictors.

We created conformal predictors using two CP algorithms (homo and ACE) and four ML methods (RF-OOB, DNN-dropout, DNN-multitask, and GB) under eight nominal coverage levels ranging from 60 to 95% with increments of 5%. Figure 2 shows the boxplots of the marginal coverage error

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

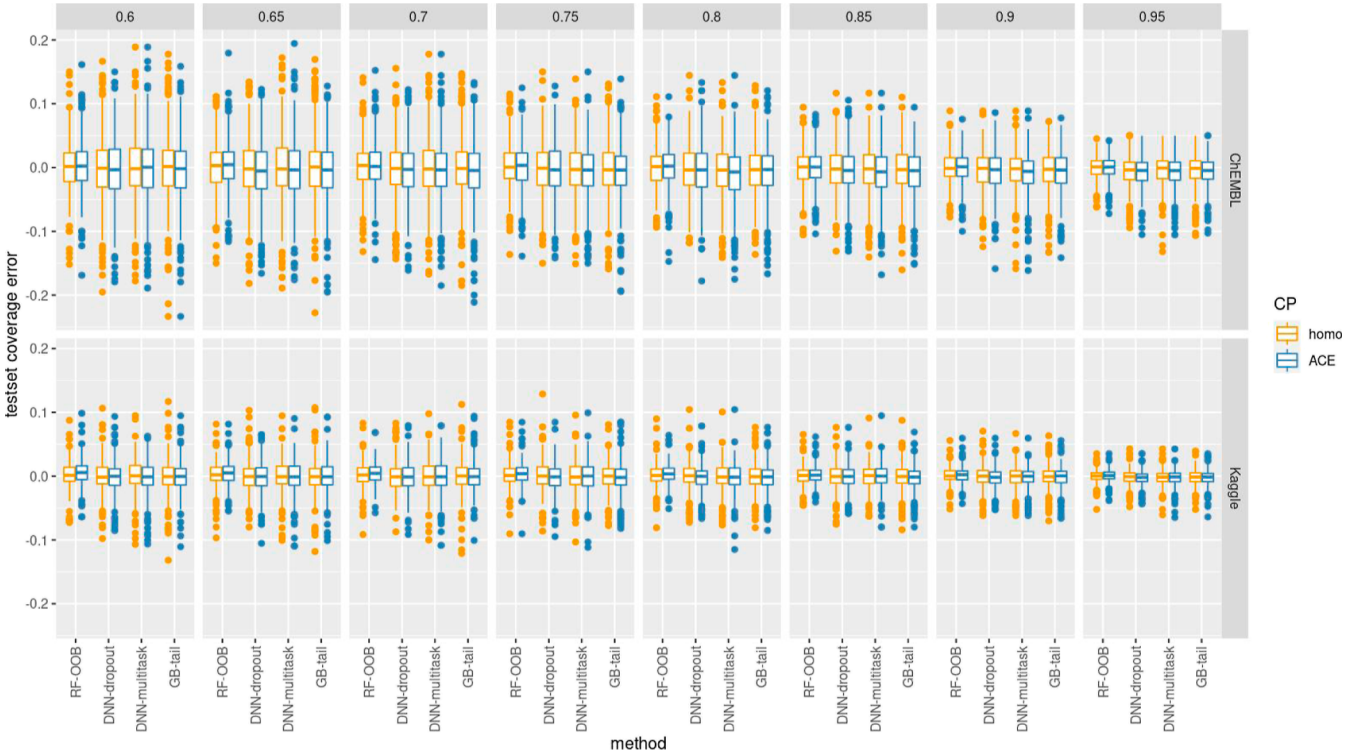


Figure 2. Boxplots of marginal coverage errors in two groups of data sets for eight nominal coverages ranging from 60 to 95%. Top row: ChEMBL data sets; bottom row: Kaggle data sets. In all the boxplots, the horizontal line drawn in the middle of each box denotes the median, and the range of box is from 25th percentile (Q_1) to 75th percentile (Q_3). The dots represent data beyond 1.5 times the interquartile range ($Q_3 - Q_1$) above Q_3 or below Q_1.

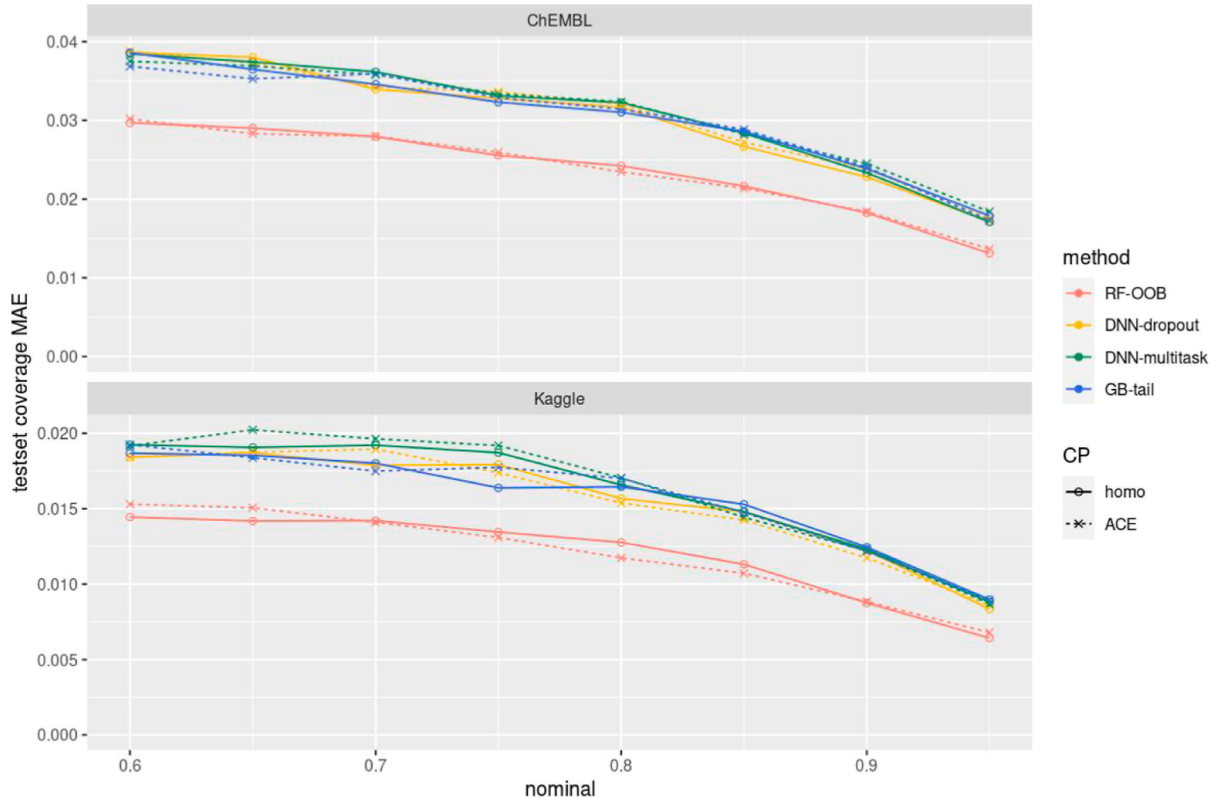


Figure 3. Absolute marginal coverage errors averaged across data sets and repeats, for eight nominal coverages ranging from 60 to 95% and two groups of data sets. Lower values indicate better marginal validity. Top row: ChEMBL data sets; bottom row: Kaggle data sets.

on test sets under different nominal coverage levels for each method and CP algorithm. In all cases, the coverage errors are centered around zero, which indicates that they all achieved marginal validity. Since the Kaggle data sets are larger than

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

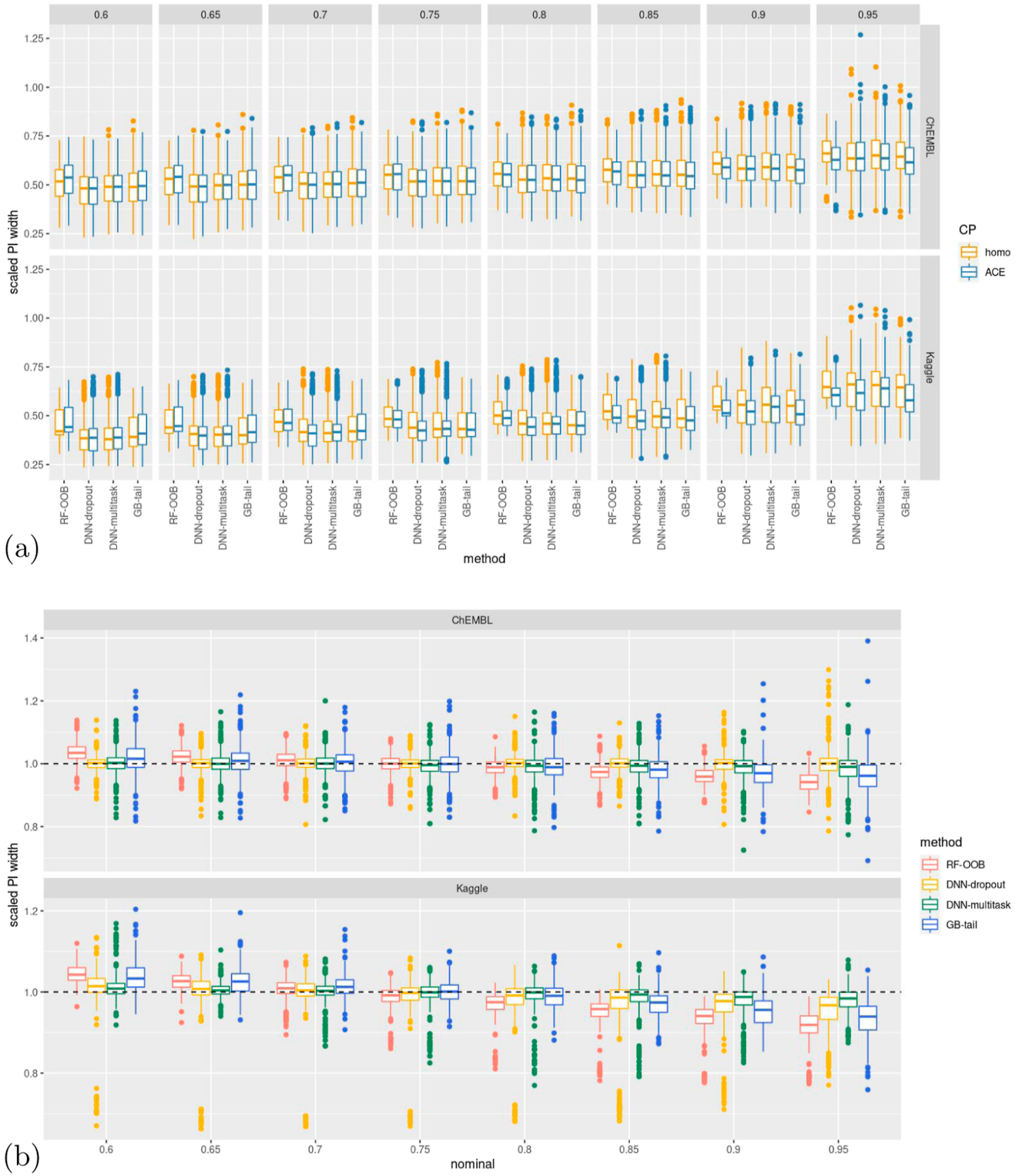


Figure 4. Boxplots of the scaled average PI widths for eight nominal coverages ranging from 60 to 95%. For each ML algorithm, CP method, and each of the 20 repeats, the PIs for each nominal coverage are averaged across the test set. In order to facilitate the comparison of average prediction widths across different data sets in the boxplots, it is necessary to scale the PIs. The two subfigures below illustrate two different scaling approaches. (a) The PIs are scaled by the "no-model" interval width, which is calculated from the training set activities rather than model predictions. For a nominal coverage of $(1-\alpha)$, the "no-model" interval is between the $\alpha/2$ quantile and $(1-\alpha/2)$ quantile values of the observed activities in the training set. Top row: ChEMBL data sets; bottom row: Kaggle data sets. Each column of subplots corresponds to one nominal coverage level. (b) The PIs are scaled by the corresponding homoscedastic PI width for the same test set and the same point prediction model. Top row: ChEMBL data sets; bottom row: Kaggle data sets. The x-axis is the nominal coverage level.

ChEMBL data sets, the distributions of errors have less variability. The comparison of variability in marginal coverage errors between the ML method and CP algorithm is shown in Figure 3. For each ML method and CP algorithm, we averaged

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

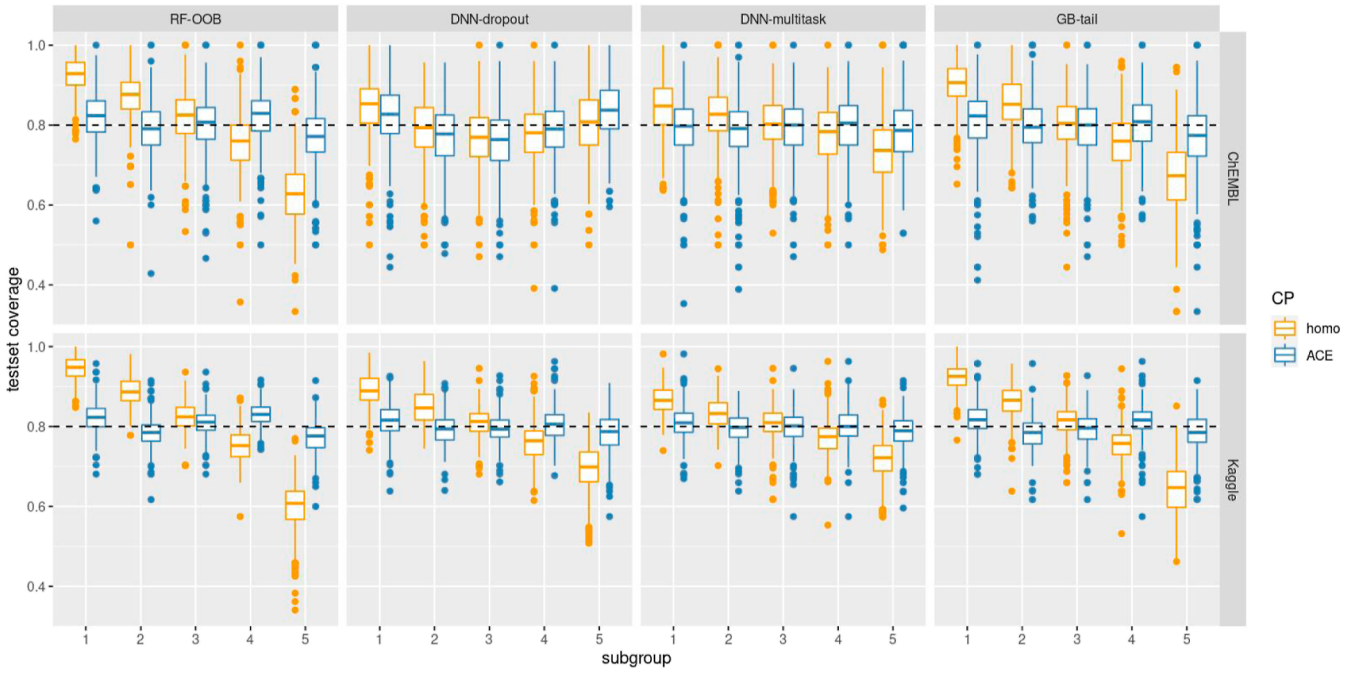


Figure 5. Test set coverage of subgroups defined by PI widths, at a nominal level of 80%. For each ML algorithm, data set, and repeat, the test set molecules are sorted by the PI widths in increasing order and split into five equal-size subgroups numbered 1 to 5. And the test set coverage for each subgroup using different CP methods is calculated as the proportion of true activity that falls within the PI at a nominal level of 80% (horizontal dashed line).

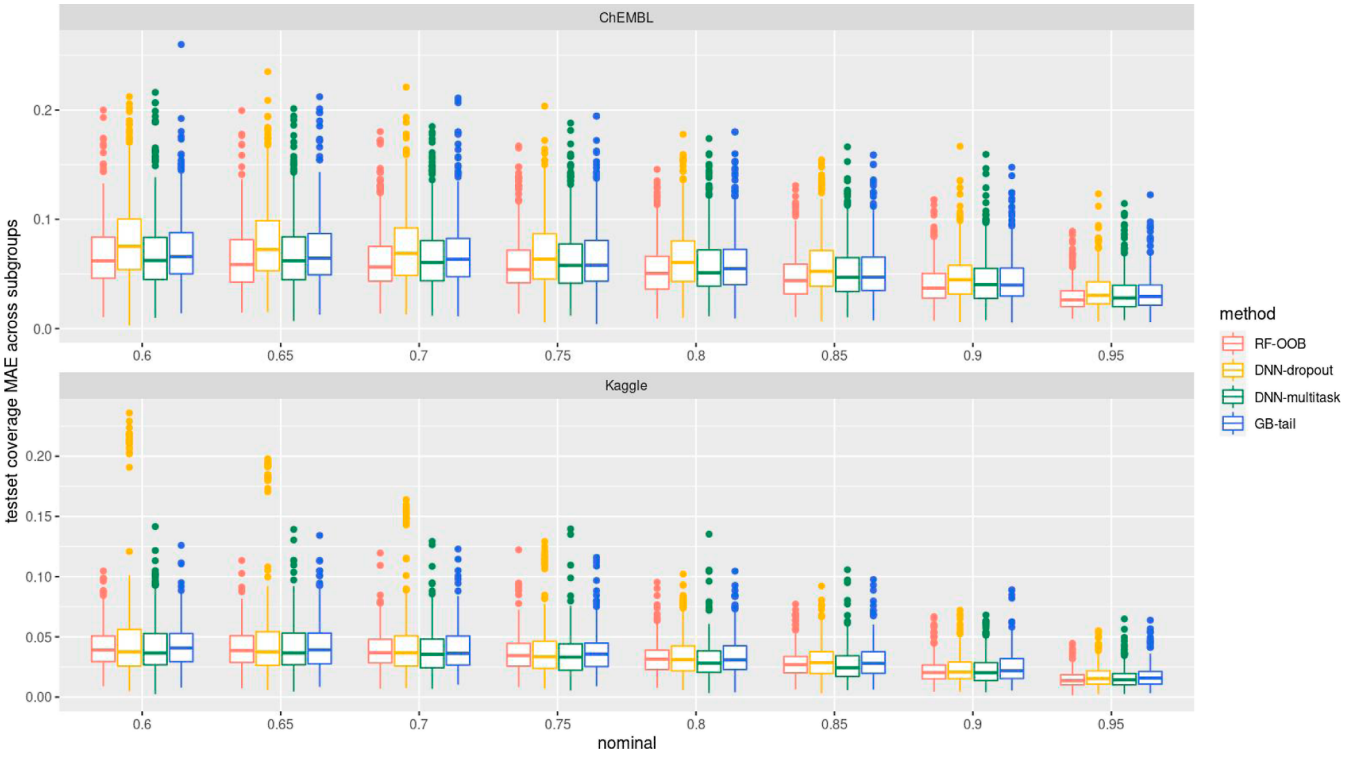


Figure 6. Comparison of the conditional coverage of ACE PIs generated by different methods. For each ML algorithm, data set, and repeat, the test set molecules are sorted by the PI widths in increasing order and split into five equal-size subgroups. The test set coverage for each subgroup using the ACE CP method is calculated as the proportion of true activity that falls within the PI at various nominal levels. The *y*-axis is the MAE of test set coverage across subgroups, i.e., the absolute errors of test set coverage averaged across five subgroups in each test set, with lower values indicating better conditional coverage.

the absolute value of coverage errors across data sets and repeats at multiple nominal coverages. The RF-OOB method has lower MAEs of marginal coverage at all nominal levels. The RF model is trained on all available data without splitting into proper training sets and calibration sets, and it used the out-of-bag data for calibration, which is effectively larger than the stand-alone

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

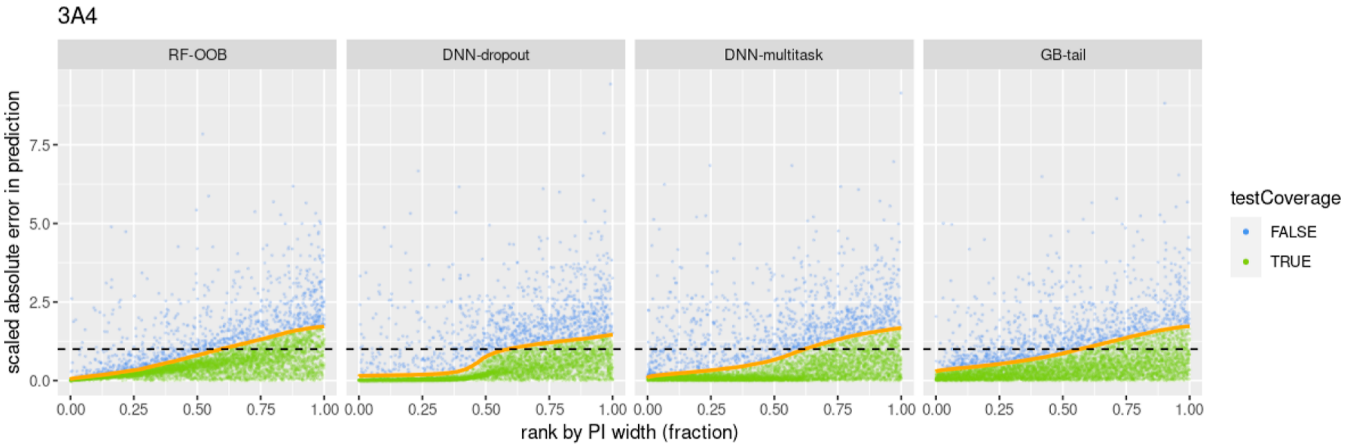


Figure 7. Association of the ACE PI widths and absolute prediction errors (nominal level of 80% for the Kaggle data set 3A4, and shown results are for one test set out of 20 repeated runs). The orange line is the PI width from the ACE algorithm. Each dot in the scatter plot is a test set molecule, colored by whether it is successfully covered by the PI (with the absolute error below the orange line). Both absolute errors and the PI width are scaled by the corresponding homoscedastic PI width. The horizontal black dash line with y-intercept at 1.0 represents the scaled homoscedastic PI size (unit length).

calibration sets in other methods. Due to this unique feature of RF, it achieved more robust performance in the marginal coverage. There was no difference between ACE and homoscedastic, which is expected.

Figure 4 compares the efficiency of conformal predictors. In Figure 4a, the DNN-dropout, DNN-multitask, and GB-tail methods have slightly narrower average PI widths than the RF-OOB method in the lower nominal range (60 to 85%), but the distributions of their average PI widths have a wider spread. Also, there are a few outliers at the higher nominal level of 95%. Figure 4b compares the efficiency between ACE and homoscedastic conformal algorithms. For each ML method, the average PI width of the ACE algorithm of each test set is scaled by the corresponding homoscedastic PI width. As the nominal level increases, the ACE algorithm is more likely to produce narrower PIs, on average, compared to homoscedastic.

Figure 5 demonstrates the conditional coverage property of ACE in contrast to homoscedastic using the nominal level of 80% as an example. Again, each test set was sorted by the PI widths for each method in increasing order and split into five equal-size subgroups. Similar to what we observed in the simulation study, the coverage of the ACE algorithm is closer to the expected nominal coverage (dashed line) in all subgroups compared to homoscedastic. In most situations, except the DNN-dropout model for ChEMBL data sets, there is a clear decreasing trend of homoscedastic PI coverage from the first subgroup to the fifth subgroup. It indicates that most of the molecules with smaller PIs have smaller prediction errors, which leads to overcoverage of homoscedastic PIs in the first two subgroups. Also, the molecules in the last two subgroups have larger prediction errors than the PI width and hence this results in the under-coverage of homoscedastic PIs. In other words, the decreasing trend of homoscedastic PI coverage demonstrates that the PI effectively differentiates between the molecules with small prediction errors from those with large prediction errors. Thus, the PIs from DNN-dropout ensembles in some cases are not as informative as the other methods. Figure 6 compares the conditional coverage of ACE PIs generated by different methods. For ChEMBL data sets, the DNN-dropout model shows a higher MAE of coverage by subgroups at multiple nominal levels. For some Kaggle data sets, the DNN-dropout model also produced noticeably higher errors at lower nominal levels.

Figure 7 shows the ACE PIs using the Kaggle 3A4 data set as an example. The molecular activities in the 3A4 data set have a truncated distribution causing heteroscedastic prediction errors. In each subplot, on the horizontal axis, the test set molecules are ordered by the PI widths from the corresponding ML method. The orange line shows that the spread of the absolute errors is increasing with the PI width for all four methods, indicating a strong association between the magnitude of the true prediction error and PI width. Thus, with the ACE algorithm, we obtained well-calibrated heteroscedastic PIs with the widths adaptive to the true absolute prediction errors. Empirical coverage of the ACE PIs reflected by the proportion of points below the regression line (numbers are not shown) is close to the nominal levels at each interval width. Thus, the ACE method provides meaningful estimates of the prediction errors.

DISCUSSION

In this work, we developed the ACE algorithm, an inductive conformal predictor whose nonconformity scores are calculated using estimates of the prediction uncertainty generated by the ensemble of point prediction models. We evaluated this method using both simulated and large collections of real QSAR data. The ACE algorithm produced PIs that are well adapted to the data: for homoscedastic data, the interval width is nearly constant; and for heteroscedastic data, the width varies with the magnitude of prediction errors and achieves close to nominal coverage for each molecule. This adaptiveness feature of ACE PIs makes it highly informative in QSAR applications. One can use the interval size to differentiate the molecules which are accurately predicted vs those that are not.

The ACE algorithm is applicable to any prediction model, as long as a raw prediction uncertainty score can be computed, and that uncertainty correlates with the variance of actual prediction errors. Since the PI widths produced by the ACE algorithm as well as other ensemble-based CP algorithms are positively correlated with the raw prediction uncertainty scores, it is critical to develop ML methods that produce raw uncertainty scores that effectively track the errors to achieve adaptivity. Some of the widely used QSAR methods, such as DNNs, provide highly accurate point predictions but do not generate raw uncertainty

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

scores because they do not naturally produce ensembles. The GB method, although an ensemble method, also does not natively generate a prediction uncertainty estimate. To address this, we proposed the novel ensemble DNN-multitask and GB-tail methods, to compute the raw prediction uncertainty score from DNN and GB point prediction models with a small computational cost. The ACE PIs with raw prediction uncertainty scores obtained by both these methods showed competitive performance in a diverse collection of QSAR data sets. Notably, in the case of neural net models, DNN-multitask has shown to produce more useful ensembles compared to DNN-dropout. Additionally, implementing the DNN-multitask method involves only a simple reformatting of the training molecular activities without changes in the neural net structure. This method could be also applied to the convolutional neural network[62−64] or graph convolutional networks.[65−68] By tailoring uncertainty quantification techniques to different ML methods, we can enhance the applicability of conformal predictors in real-world scenarios. While a comparison across different ML methods could provide valuable insights, it falls beyond the scope of this paper.

The topic of uncertainty quantification has also been explored in the applicability domain (AD) research, a subfield in QSAR.[69−71] However, there are many definitions of the concept of AD and there is no consensus on the best approach.[2,32] To improve the value of AD model, Hanser et al.[72] clarified, formalized, and extended the definition of AD. In this work, the evaluation of AD was split into three well-defined subtopics: the chemical structure space boundaries of model applicability, the reliability of predictions, and whether the predictions can help to make a clear decision. The well-defined mathematical framework of CP produces PIs with straightforward interpretation, which is a valuable addition to the prediction reliability assessment methodology in AD.

Although this work focuses on developing conformal PIs for QSAR regression tasks, there are CP algorithms for classification modeling and have been applied in QSAR before.[24,32,33,73−75] In classification, the nonconformity score is usually defined based on the predicted probabilities associated with each class. The proposed DNN-multitask and GB-tail methods can also be used to compute the raw prediction uncertainty scores for classification models, which could be helpful for constructing novel nonconformity measures. Another interesting direction to explore in future study is comparing the performance of uncertainty estimation methods on various tasks, either regression or classification, categorized by the underlying data distribution.

In conclusion, the conformal predictors computed by the proposed ACE algorithm jointly with highly accurate and commonly used ML models may serve as practical uncertainty quantification tools for QSAR modeling, producing accurate and informative PIs without compromising the point prediction quality or computational efficiency.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.4c02017.

Visualization of data set distribution, download links for two collections of QSAR data sets, ML algorithm implementation and parameters, hyperparameter tuning for prediction uncertainty score methods, simulation settings, results for data with simulated noise, and companion summary statistic tables for boxplot figures in section "Results: Application to ChEMBL and Kaggle data sets" (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Yuting Xu − *Early Development Statistics, Merck & Co., Inc., Rahway, New Jersey 07065, United States;* orcid.org/0000-0003-2091-3854; Email: yuting.xu@merck.com

Authors

Andy Liaw − *Early Development Statistics, Merck & Co., Inc., Rahway, New Jersey 07065, United States*

Robert P. Sheridan − *Modeling and Informatics, Merck & Co., Inc., Rahway, New Jersey 07033, United States*

Vladimir Svetnik − *Early Development Statistics, Merck & Co., Inc., Rahway, New Jersey 07065, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.4c02017

Notes

The authors declare the following competing financial interest(s): All authors are employed by Merck Sharp & Dohme LLC, a subsidiary of Merck & Co., Inc., Rahway, NJ, USA.

■ ACKNOWLEDGMENTS

The authors wish to thank the EDS-BARDS department and the M&I-Chemistry department in MRL for support in this work.

■ REFERENCES

(1) Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R.; et al. QSAR modeling: where have you been? Where are you going to? *J. Med. Chem.* **2014**, *57*, 4977−5010.

(2) Muratov, E. N.; Bajorath, J.; Sheridan, R. P.; Tetko, I. V.; Filimonov, D.; Poroikov, V.; Oprea, T. I.; Baskin, I. I.; Varnek, A.; Roitberg, A.; et al. QSAR without borders. *Chem. Soc. Rev.* **2020**, *49*, 3525−3564.

(3) Xu, Y. *Artificial Intelligence in Drug Design*; Springer, 2022; pp 233−260.

(4) Sahlin, U. Uncertainty in QSAR predictions. *ATLA, Altern. Lab. Anim.* **2013**, *41*, 111−125.

(5) Sheridan, R. P. The relative importance of domain applicability metrics for estimating prediction errors in QSAR varies with training set diversity. *J. Chem. Inf. Model.* **2015**, *55*, 1098−1107.

(6) Hirschfeld, L.; Swanson, K.; Yang, K.; Barzilay, R.; Coley, C. W. Uncertainty quantification using neural networks for molecular property prediction. *J. Chem. Inf. Model.* **2020**, *60*, 3770−3780.

(7) Mervin, L. H.; Johansson, S.; Semenova, E.; Giblin, K. A.; Engkvist, O. Uncertainty quantification in drug design. *Drug Discovery Today* **2021**, *26*, 474−489.

(8) Yu, J.; Wang, D.; Zheng, M. Uncertainty quantification: Can we trust artificial intelligence in drug discovery? *Iscience* **2022**, *25*, 104814.

(9) Koenker, R.; Hallock, K. F. Quantile regression. *J. Econ. Perspect.* **2001**, *15*, 143−156.

(10) Hao, L.; Naiman, D. Q.; Naiman, D. Q. *Quantile Regression*; Sage, 2007.

(11) Sesia, M.; Candès, E. J. A comparison of some conformal quantile regression methods. *Stat.* **2020**, *9*, No. e261.

(12) El-Telbany, M. E. What quantile regression neural networks tell us about prediction of drug activities. *10th International Computer Engineering Conference (ICENCO)*; IEEE, 2014; pp 76−80.

(13) Feng, D.; Svetnik, V.; Liaw, A.; Pratola, M.; Sheridan, R. P. Building quantitative structure−activity relationship models using Bayesian additive regression trees. *J. Chem. Inf. Model.* **2019**, *59*, 2642−2655.
(14) DiFranzo, A.; Sheridan, R. P.; Liaw, A.; Tudor, M. Nearest neighbor gaussian process for quantitative structure−activity relationships. *J. Chem. Inf. Model.* **2020**, *60*, 4653−4663.
(15) Kolmar, S. S.; Grulke, C. M. The effect of noise on the predictive limit of QSAR models. *J. Cheminf.* **2021**, *13*, 92.
(16) Damme, S. V.; Bultinck, P. A new computer program for QSAR-analysis: ARTE-QSAR. *J. Comput. Chem.* **2007**, *28*, 1924−1928.
(17) Wood, D. J.; Carlsson, L.; Eklund, M.; Norinder, U.; Stålring, J. QSAR with experimental and predictive distributions: an information theoretic approach for assessing model quality. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 203−219.
(18) Cortés-Ciriano, I.; Bender, A. Deep confidence: a computationally efficient framework for calculating reliable prediction errors for deep neural networks. *J. Chem. Inf. Model.* **2019**, *59*, 1269−1281.
(19) Cortes-Ciriano, I.; Bender, A. Reliable prediction errors for deep neural networks using test-time dropout. *J. Chem. Inf. Model.* **2019**, *59*, 3330−3339.
(20) Nix, D. A.; Weigend, A. S. Estimating the mean and variance of the target probability distribution. *Proceedings of 1994 IEEE International Conference On Neural Networks (ICNN'94)*; IEEE, 1994; pp 55−60.
(21) Huang, W.; Zhao, D.; Sun, F.; Liu, H.; Chang, E. Scalable Gaussian process regression using deep neural networks. *Twenty-Fourth International Joint Conference On Artificial Intelligence*; Citeseer, 2015.
(22) Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U. R.; et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion* **2021**, *76*, 243−297.
(23) Tian, Q.; Nordman, D. J.; Meeker, W. Q. Methods to compute prediction intervals: A review and new results. *Stat. Sci.* **2022**, *37*, 580−597.
(24) Alvarsson, J.; Arvidsson McShane, S.; Norinder, U.; Spjuth, O. Predicting with confidence: using conformal prediction in drug discovery. *J. Pharm. Sci.* **2021**, *110*, 42−49.
(25) Boström, H.; Johansson, U. Mondrian conformal regressors. *Conformal and Probabilistic Prediction and Applications*; PMLR, 2020; pp 114−133.
(26) Boström, H. crepes: a Python package for generating conformal regressors and predictive systems. *Conformal and Probabilistic Prediction with Applications*; PMLR, 2022; pp 24−41.
(27) McShane, S. A.; Norinder, U.; Alvarsson, J.; Ahlberg, E.; Carlsson, L.; Spjuth, O. CPSign-Conformal Prediction for Cheminformatics Modeling. *bioRxiv* **2023**, 2023−2111.
(28) Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A. F. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Trans. Neural Netw. Learn. Syst.* **2011**, *22*, 1341−1356.
(29) Scalia, G.; Grambow, C. A.; Pernici, B.; Li, Y.-P.; Green, W. H. Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction. *J. Chem. Inf. Model.* **2020**, *60*, 2697−2717.
(30) Eklund, M.; Norinder, U.; Boyer, S.; Carlsson, L. Application of conformal prediction in QSAR. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*; Springer, 2012; pp 166−175.
(31) Carlsson, L.; Eklund, M.; Norinder, U. Aggregated conformal prediction. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*; Springer, 2014; pp 231−240.
(32) Norinder, U.; Carlsson, L.; Boyer, S.; Eklund, M. Introducing conformal prediction in predictive modeling. A transparent and flexible alternative to applicability domain determination. *J. Chem. Inf. Model.* **2014**, *54*, 1596−1603.
(33) Eklund, M.; Norinder, U.; Boyer, S.; Carlsson, L. The application of conformal prediction to the drug discovery process. *Ann. Math. Artif. Intell.* **2015**, *74*, 117−132.
(34) Svensson, F.; Aniceto, N.; Norinder, U.; Cortes-Ciriano, I.; Spjuth, O.; Carlsson, L.; Bender, A. Conformal regression for quantitative structure−activity relationship modeling—quantifying prediction uncertainty. *J. Chem. Inf. Model.* **2018**, *58*, 1132−1140.
(35) Cortés-Ciriano, I.; Bender, A. Concepts and applications of conformal prediction in computational drug discovery. 2022, arXiv:1908.03569. arXiv preprint. https://arxiv.org/abs/1908.03569.
(36) Vovk, V.; Gammerman, A.; Shafer, G. *Algorithmic Learning in a Random World*; Springer Science & Business Media, 2005.
(37) Lei, J.; G'Sell, M.; Rinaldo, A.; Tibshirani, R. J.; Wasserman, L. Distribution-free predictive inference for regression. *J. Am. Stat. Assoc.* **2018**, *113*, 1094−1111.
(38) Shafer, G.; Vovk, V. A Tutorial on Conformal Prediction. *J. Mach. Learn. Res.* **2008**, *9*, 371.
(39) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep neural nets as a method for quantitative structure−activity relationships. *J. Chem. Inf. Model.* **2015**, *55*, 263−274.
(40) Svetnik, V.; Wang, T.; Tong, C.; Liaw, A.; Sheridan, R. P.; Song, Q. Boosting: An ensemble learning tool for compound classification and QSAR modeling. *J. Chem. Inf. Model.* **2005**, *45*, 786−799.
(41) Sheridan, R. P.; Wang, W. M.; Liaw, A.; Ma, J.; Gifford, E. M. Extreme gradient boosting as a method for quantitative structure−activity relationships. *J. Chem. Inf. Model.* **2016**, *56*, 2353−2360.
(42) Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference On Machine Learning*; PMLR, 2016; pp 1050−1059.
(43) Romano, Y.; Patterson, E.; Candes, E. Conformalized quantile regression. *33rd Annual Conference on Neural Information Processing Systems*; NeurIPS, 2019; Vol. *32*.
(44) Sesia, M.; Romano, Y. Conformal prediction using conditional histograms. *Advances in Neural Information Processing Systems*; NeurIPS, 2021; Vol. *34*, pp 6304−6315.
(45) Chernozhukov, V.; Wüthrich, K.; Zhu, Y. Distributional conformal prediction. *Proc. Natl. Acad. Sci.* **2021**, *118*, No. e2107794118.
(46) Vovk, V.; Shen, J.; Manokhin, V.; Xie, M.-g. Nonparametric predictive distributions based on conformal prediction. In *Proceedings of Machine Learning Research*; Conformal and Probabilistic Prediction and Applications; PMLR, 2017; pp 82−102.
(47) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.
(48) Landrum, G.; et al. *RDKit: Open-source cheminformatics*, 2006.
(49) Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom pairs as molecular features in structure-activity studies: definition and applications. *J. Chem. Inf. Model.* **1985**, *25*, 64−73.
(50) Kearsley, S. K.; Sallamack, S.; Fluder, E. M.; Andose, J. D.; Mosley, R. T.; Sheridan, R. P. Chemical similarity using physiochemical property descriptors. *J. Chem. Inf. Model.* **1996**, *36*, 118−127.
(51) Papadopoulos, H. *Tools in Artificial Intelligence*; Citeseer, 2008.
(52) Papadopoulos, H.; Proedrou, K.; Vovk, V.; Gammerman, A. Inductive confidence machines for regression. *European Conference on Machine Learning*; Springer, 2002; pp 345−356.
(53) Johansson, U.; Boström, H.; Löfström, T.; Linusson, H. Regression conformal prediction with random forests. *Mach. Learn.* **2014**, *97*, 155−176.
(54) Papadopoulos, H.; Vovk, V.; Gammerman, A. Regression conformal prediction with nearest neighbours. *J. Artif. Intell. Res.* **2011**, *40*, 815−840.
(55) Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J. C.; Sheridan, R. P.; Feuston, B. P. Random forest: a classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Model.* **2003**, *43*, 1947−1958.
(56) Vovk, V. Conditional validity of inductive conformal predictors. *Mach. Learn.* **2013**, *92*, 349−376.
(57) Feldman, S.; Bates, S.; Romano, Y. Improving conditional coverage via orthogonal quantile regression. *Advances in Neural*

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490

Information Processing Systems; Curran Associates, Inc., 2021; Vol. *34*, pp 2060−2071.
(58) Foygel Barber, R.; Candes, E. J.; Ramdas, A.; Tibshirani, R. J. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA* **2021**, *10*, 455−482.
(59) Bastani, O.; Gupta, V.; Jung, C.; Noarov, G.; Ramalingam, R.; Roth, A. Practical adversarial multivalid conformal prediction. *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2022; Vol. *35*, pp 29362−29373.
(60) Jung, C.; Noarov, G.; Ramalingam, R.; Roth, A. Batch multivalid conformal prediction. 2022, arXiv:2209.15145. arXiv preprint. https://arxiv.org/abs/2209.15145.
(61) Cauchois, M.; Gupta, S.; Duchi, J. C. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *J. Mach. Learn. Res.* **2021**, *22*, 1−42.
(62) LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278−2324.
(63) Meyer, J. G.; Liu, S.; Miller, I. J.; Coon, J. J.; Gitter, A. Learning drug functions from chemical structures with convolutional neural networks and random forests. *J. Chem. Inf. Model.* **2019**, *59*, 4438−4449.
(64) Shi, T.; Yang, Y.; Huang, S.; Chen, L.; Kuang, Z.; Heng, Y.; Mei, H. Molecular image-based convolutional neural network for the prediction of ADMET properties. *Chemom. Intell. Lab. Syst.* **2019**, *194*, 103853.
(65) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2015; pp 2224−2232.
(66) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J. Comput.-Aided Mol. Des.* **2016**, *30*, 595−608.
(67) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513−530.
(68) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370−3388.
(69) Jaworska, J.; Nikolova-Jeliazkova, N.; Aldenberg, T. QSAR applicability domain estimation by projection of the training set in descriptor space: a review. *ATLA, Altern. Lab. Anim.* **2005**, *33*, 445−459.
(70) Weaver, S.; Gleeson, M. P. The importance of the domain of applicability in QSAR modeling. *J. Mol. Graphics Modell.* **2008**, *26*, 1315−1326.
(71) Sahigara, F.; Mansouri, K.; Ballabio, D.; Mauri, A.; Consonni, V.; Todeschini, R. Comparison of different approaches to define the applicability domain of QSAR models. *Molecules* **2012**, *17*, 4791−4810.
(72) Hanser, T.; Barber, C.; Marchaland, J.; Werner, S. Applicability domain: towards a more formal definition. *SAR QSAR Environ. Res.* **2016**, *27*, 865−881.
(73) Bosc, N.; Atkinson, F.; Felix, E.; Gaulton, A.; Hersey, A.; Leach, A. R. Large scale comparison of QSAR and conformal prediction methods and their applications in drug discovery. *J. Cheminf.* **2019**, *11*, 4−16.
(74) Norinder, U.; Boyer, S. Binary classification of imbalanced datasets using conformal prediction. *J. Mol. Graphics Modell.* **2017**, *72*, 256−265.
(75) Romano, Y.; Sesia, M.; Candes, E. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. *33*, pp 3581−3591.

https://doi.org/10.1021/acsomega.4c02017
ACS Omega 2024, 9, 29478−29490