Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction

As Published: 10.1021/acs.jcim.9b00975

Publisher: American Chemical Society (ACS)

Persistent URL: https://hdl.handle.net/1721.1/134634

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

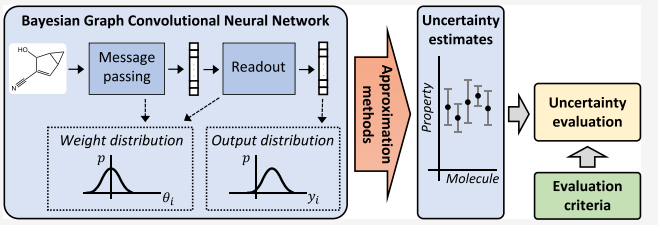
Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction

Gabriele Scalia, Colin A. Grambow, Barbara Pernici, Yi-Pei Li,* and William H. Green*

Cite This: https://dx.doi.org/10.1021/acs.jcim.9b00975 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Advances in deep neural network (DNN)-based molecular property prediction have recently led to the development of models of remarkable accuracy and generalization ability, with graph convolutional neural networks (GCNNs) reporting state-of-the-art performance for this task. However, some challenges remain, and one of the most important that needs to be fully addressed concerns uncertainty quantification. DNN performance is affected by the volume and the quality of the training samples. Therefore, establishing when and to what extent a prediction can be considered reliable is just as important as outputting accurate predictions, especially when out-of-domain molecules are targeted. Recently, several methods to account for uncertainty in DNNs have been proposed, most of which are based on approximate Bayesian inference. Among these, only a few scale to the large data sets required in applications. Evaluating and comparing these methods has recently attracted great interest, but results are generally fragmented and absent for molecular property prediction. In this paper, we quantitatively compare scalable techniques for uncertainty estimation in GCNNs. We introduce a set of quantitative criteria to capture different uncertainty aspects and then use these criteria to compare MC-dropout, Deep Ensembles, and bootstrapping, both theoretically in a unified framework that separates aleatoric/epistemic uncertainty and experimentally on public data sets. Our experiments quantify the performance of the different uncertainty estimation methods and their impact on uncertainty-related error reduction. Our findings indicate that Deep Ensembles and bootstrapping consistently outperform MC-dropout, with different context-specific pros and cons. Our analysis leads to a better understanding of the role of aleatoric/epistemic uncertainty, also in relation to the target data set features, and highlights the challenge posed by out-of-domain uncertainty.



INTRODUCTION

Deep neural network (DNN)-based molecular property prediction has received new attention recently with the development of models capable of promising performance on large and heterogeneous data sets.[1−3] In particular, recent progress in graph convolutional neural networks[4] (GCNNs)—also known as *message passing neural networks*—have led to state-of-the-art performance for property prediction across a range of public and proprietary data sets,[1] demonstrating both accuracy and generalization gains. However, some limitations still hold, and uncertainty quantification has recently been highlighted as an important direction to be investigated.[1]

The need for an effective uncertainty quantification is driven by both intrinsic characteristics of DNN models and by peculiar features of chemical space. In general, standard DNN models do not output confidence estimates, since regression models only output a mean, while classification outputs cannot be reliably interpreted as confidence scores.[5]

DNN performance strongly depends on the volume and the quality of training data, hence there is a need to assess when and to what extent a prediction can be considered reliable. While this has emerged in the context of DNNs in several heterogeneous applications, most of which are based on computer vision,[6] DNN for chemistry is characterized by additional challenges. First of all, chemical training data are intrinsically biased[7] because the chemical space has an extremely large variability, and therefore, a training data set cannot represent the whole space. Moreover, chemical training data are often limited in volume and quality. Additionally, doing predictions on molecules rather different to those seen during training is often the actual goal in the field, for example, in drug discovery applications. This demands good generalization performance on one side but also being able to identify the model's *knowledge boundary*, that is, assessing to what extent the model knows what it knows.

While uncertainty estimation in this domain has been investigated in the context of shallow models in the last few years,[8] less is known about uncertainty in DNN and GCNN models for molecular property prediction.

Received: October 20, 2019
Published: April 3, 2020

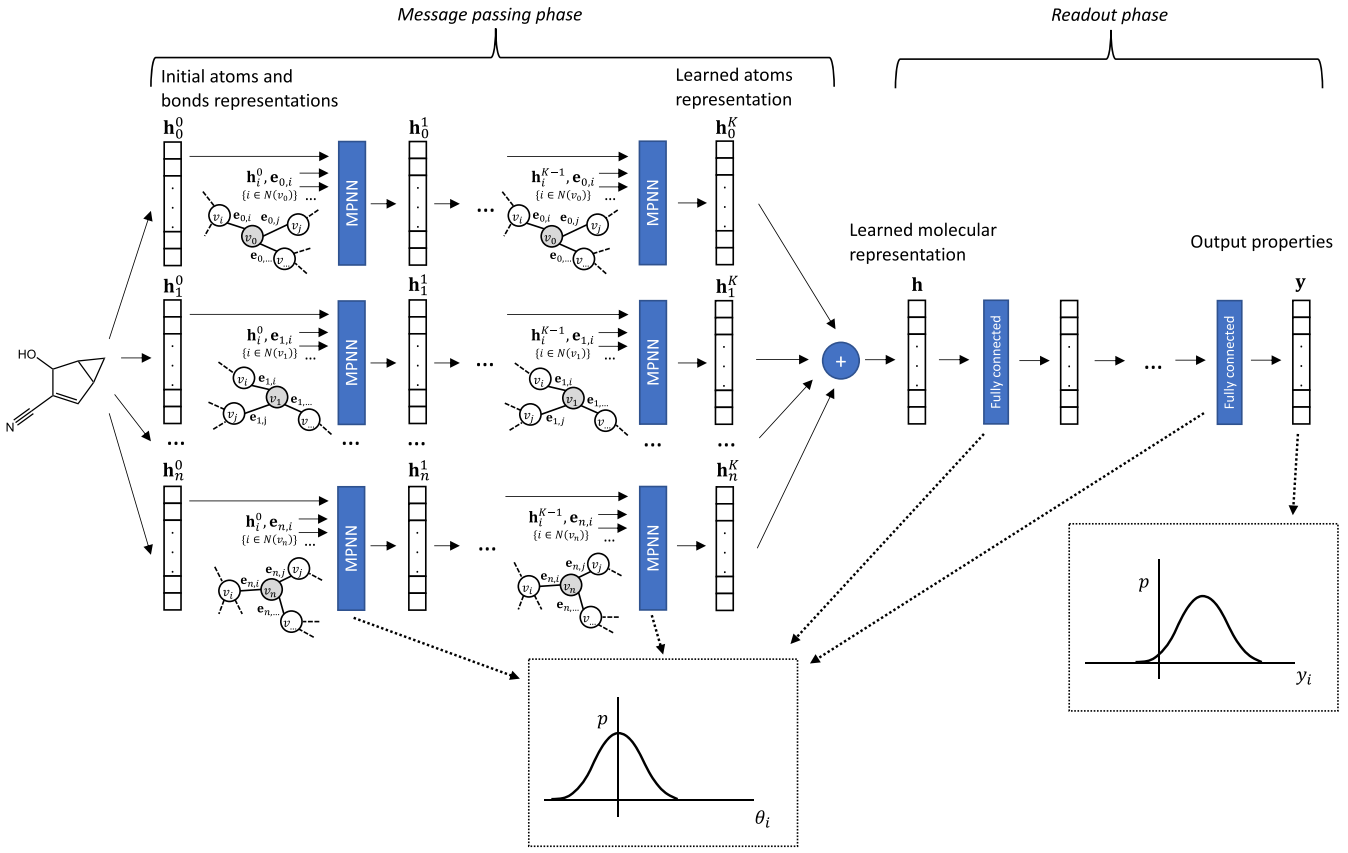


Figure 1. Illustrative overview of a GCNN for molecular property prediction extended as a BNN. *Message passing phase.* Starting from a molecular graph, the model extracts an initial representation $\mathbf{h}_i^0$ for each atom v_i and a bond representation $\mathbf{e}_{i,j}$ for each bond between v_i and v_j. At each step, the atom representation is updated based on the representation of the atom's neighbors and the related bonds. $\mathbf{h}_i^j$ refers to the representation for the i-th atom at the j-th update step and $N(v_i)$ are the atom's neighbors. At the end (K update steps), the molecule representation $\mathbf{h}$ results from the sum of the learned atoms features. *Readout phase.* The molecular representation is updated through a series of fully connected layers obtaining the output vector $\mathbf{y}$. The peculiarity of a BNN is to model network weights and outputs as probability distributions (e.g., Gaussians), instead of point estimates. This allows taking into account uncertainty—epistemic and aleatoric—in the model.

Bayesian neural networks (BNNs) have long been studied as an effective and principled way to take into account model uncertainty in the predictions of a DNN,[9] but the intractability of exact Bayesian inference together with the limited practicality of the approaches proposed has prevented the widespread diffusion of these solutions in applications until recently.[5] The recent work from Gal and Ghahramani[10] gave a decisive contribution to the spread of approximate BNNs in applications, proposing Monte Carlo dropout (MC-dropout), a practical method based on the widely used dropout regularization technique, to account for model uncertainty. Moreover, Kendall and Gal[6] proposed a framework to separate *epistemic uncertainty*, which refers to uncertainty in the model predictions, from the *aleatoric uncertainty*, which captures noise inherent in the data. MC-dropout has been used in various applications, including molecular property prediction.[7,11]

Other techniques to efficiently approximate BNNs have been proposed. Finding a good trade-off between effective approximation and scalability remains an important open challenge. Notably, the ensemble-based approach proposed by Lakshminarayanan et al.[12] constitutes a simple and scalable technique to obtain well-calibrated uncertainty estimates and has already been used in several applications.[13,14] Moreover, even if originally proposed as a non-Bayesian alternative to estimate uncertainty in DNNs,[12] recent work highlighted how ensembling in DNNs can be traced back to Bayesian inference.[15−17]

In parallel to the development of methods to efficiently approximate BNNs, their evaluation and in particular their comparative assessment have recently attracted great interest given the challenges it poses.[17−20] Indeed, we usually do not have "ground truth uncertainties", which prevents using traditional benchmarks. Furthermore, evaluating uncertainty involves measuring the model's unknowns and taking into account domain-specific features. Comparative assessments have been conducted for computer vision tasks.[17,19,20] However, no comparisons have been carried out for GCNNs in the chemistry domain. Moreover, many metrics traditionally used to evaluate uncertain forecasts, such as *calibration*, have been defined in a classification setting, while their extension for *regression*—needed for scalar molecular properties—has been discussed only recently.[21,22]

Comparative analysis of different methods calls for *multiple metrics* and *quantitative indices*. In contrast, recent works targeting uncertainty estimation for DNN-based molecular property prediction only employ a single technique, such as confidence-error diagrams and qualitative evaluations.[7,11]

In the cheminformatics field, defining the set of molecules for which a model can reliably generate predictions (its *domain of applicability*, DA) is still a very active area of research, and different definitions and methods have been introduced for this purpose.[23,24]

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

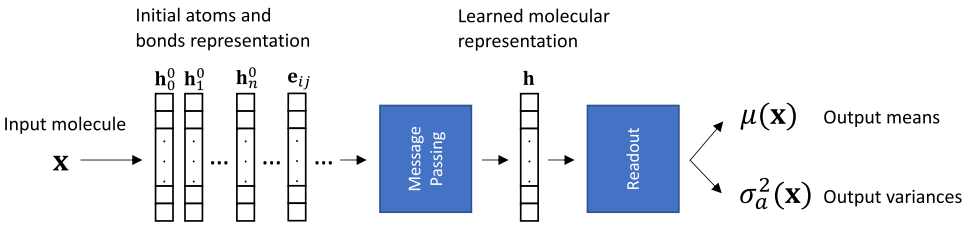


Figure 2. Aleatoric uncertainty estimation assuming an underlying Gaussian error. Last layer of the DNN is split to predict both the mean μ and the variance σ for each output property, and the network is trained minimizing the loss (eq 2). Predicted means correspond to the output properties and the predicted variances correspond to the aleatoric uncertainties (one for each property).

Here, we first review existing methods for uncertainty estimation in DNNs, focusing on scalable and practical techniques that can be used in applications. We contextualize them in a unique framework to estimate aleatoric and epistemic uncertainty, and we draw a theoretical comparison. Second, we introduce a set of uncertainty evaluation criteria, which are based on the extension of existing benchmarks used in other fields and on chemistry-specific features, also including *out-of-domain* performance evaluation. Third, we extend a recently published state-of-the-art GCNN for molecular property prediction (chemprop[1]) with the presented uncertainty estimation methods and we compare them on multiple public data sets/properties (QM9, Alchemy, PDBbind and Lipophilicity) for the regression task. We quantify the positive impact of modeling uncertainty in the network on the prediction error and discuss the behavior of aleatoric and epistemic uncertainty in different contexts, including to what extent they are correlated and related to ground truth errors, how they are affected by data set features, and their performance with respect to test set domain shifts.

METHODS

This section is organized as follows. We first summarize GCNNs, which constitute the state-of-the-art for DNN-based molecular property prediction. We then review Bayesian Uncertainty Estimation in DNNs, detailing the methods that will be tested. Finally, we discuss uncertainty evaluation and related metrics. An overview of a GCNN extended as a BNN is shown in Figure 1.

Graph Convolutional Neural Networks. A GCNN used for property prediction takes as input a molecular graph G, where the nodes are atoms and the edges are bonds, with each atom v_i initialized with the feature vector $\mathbf{h}_i^0$ and each bond $v_i - v_j$ with the feature vector $\mathbf{e}_{i,j}$ and then operates in two phases (see Figure 1). During the first phase—*message passing*—each atom's feature vector is updated based on the neighbors' features and related bond representations. This phase is executed K times, iteratively, so that in the steps following the first one each atom's feature $\mathbf{h}_v^t$ is updated based on already updated neighbors features. This allows the interaction of distant atoms in the resulting representations. At the end, the molecule representation is given by the sum of its atoms representations. The second phase—*readout*—is based on a feedforward neural network that uses the final representation of the molecule to predict some properties of interest. The message passing phase allows the model to learn its own feature representations, while the readout phase allows learning the relationship between such representations and output properties.

Starting from this general description, several specific improvements have been recently proposed.[1−3,11,25] Here, we start from a well-tested software package, chemprop,[1] that recently reported state-of-the-art performance on multiple data sets. One of the peculiar features of chemprop is the usage of messages associated with directed edges (bonds) instead of vertices (atoms). Interested readers can refer to the original work by reported by Yang et al.[1] for the details.

We extended the chemprop model to include the uncertainty estimation and evaluation methods presented next. The new version of the software is available at https://github.com/gscalia/chemprop/tree/uncertainty.

Bayesian Uncertainty Estimation. Uncertainty can be the result of inherent data noise or could be related to what the model does not yet know. These two kinds of uncertainties—*aleatoric* and *epistemic*—are reviewed in the next two sections, together with scalable techniques which have been proposed for their approximate computation. At the end, we describe how these two kinds of uncertainty can be combined to obtain the *total uncertainty* of a prediction.

Aleatoric Uncertainty. When not explicitly modeled, the inherent observation noise is assumed to be the same for every molecule. This defines a *homoscedastic* aleatoric uncertainty, that is, an uncertainty which does not vary over the data distribution and is essentially only task-dependent.[26] However, this assumption does not hold in many realistic settings. For chemistry applications, it is usually difficult to obtain high-quality data on a large set of molecules; therefore, one often needs to use multiple data sources of various accuracy to compose a large enough data set to train a model. Data-dependent aleatoric uncertainty is referred to as *heteroscedastic*,[27] and its importance for DNNs has been recently highlighted.[6,11]

Because aleatoric uncertainty is a property of data, it can be learned directly from data adapting the model and the loss function. Assuming an underlying Gaussian error, the model (with weights θ) can estimate both the mean μ and the variance σ^2 of the output distribution $\mathbf{y}$ given an input $\mathbf{x}$

$$p(\mathbf{y}|\mathbf{x}, \theta) = \mathcal{N}(\mu_\theta(\mathbf{x}), \sigma_\theta{}^2(\mathbf{x})) \tag{1}$$

This does not require "noise labels" but only changing the loss function. Indeed, by performing maximum a posteriori estimation (MAP) inference, we obtain[28]

$$\mathcal{L}(\theta) \propto \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2\sigma_\theta{}^2(\mathbf{x}_i)} \|\mathbf{y}_i - \mu_\theta(\mathbf{x}_i)\|_2{}^2 + \frac{1}{2}\log \sigma_\theta{}^2(\mathbf{x}_i) \tag{2}$$

with an additional *weight decay* term. Notice that, assuming a homoscedastic uncertainty, minimizing eq 2 coincides with the usual MSE. In practice, the last layer of the DNN is split to predict both μ_θ and $\sigma_\theta{}^2$, and the network is trained using eq 2, with $\sigma_\theta{}^2$ implicitly learned. The output $\sigma_\theta{}^2$ corresponds to the heteroscedastic aleatoric uncertainty: $\sigma_a{}^2 = \sigma_\theta{}^2$. This is shown in Figure 2.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Interestingly, σ_θ^2 in eq 2 can be interpreted as *learned loss attenuation*.[6] Intuitively, the network can learn to increase σ_θ^2 to reduce the impact of uncertain predictions on the overall loss. The second term prevents outputting an infinite uncertainty for every point.

This approach is very practical, requiring minimal modifications to the original network, and can be used in conjunction with MC-dropout,[6] ensembling,[12] or other estimates of epistemic uncertainty.

The output distribution does not need to be Gaussian (see Figure 1); more complex models could be used, such as mixture density networks (MDN)[29,30] or compound density networks.[31]

Being predicted as a data variance, aleatoric uncertainty cannot account for uncertainty in the model's parameters θ. Moreover, the MAP estimate does not take into account multiple plausible values for θ but only the most probable one. This can be overcome by performing Bayesian inference, as discussed next.

Epistemic Uncertainty. In a BNN, the parameters θ are modeled as *distributions* learned from training data $\mathcal{D}$, instead of point estimates, and therefore, it is possible to predict the output distribution $\mathbf{y}$ of some new input $\mathbf{x}$ through the *predictive posterior distribution*, eq 3.

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \int p(\mathbf{y}|\mathbf{x}, \theta)p(\theta|\mathcal{D})\mathrm{d}\theta \tag{3}$$

Equation 3 accounts for epistemic uncertainty because a prediction is the "weighted sum" of each outcome for each possible θ configuration of the model, with more probable θ having higher weight. The probability of θ depends on $\mathcal{D}$.

Monte Carlo integration over M samples $\theta^{(i)}$ of the posterior distribution $p(\theta|\mathcal{D})$ can approximate the integral, eq 4.

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) \approx \frac{1}{M}\sum_{i=1}^{M} p(\mathbf{y}|\mathbf{x}, \theta^{(i)}), \qquad \theta^{(i)} \sim p(\theta|\mathcal{D}) \tag{4}$$

However, obtaining samples $\theta^{(i)}$ directly from the true posterior distribution is virtually impossible for neural networks. Therefore, an approximate posterior distribution $q(\theta) \approx p(\theta|\mathcal{D})$ is introduced, and approximate samples $\theta^{(i)}_{\text{approx}} \sim q(\theta)$ are used instead.

Several methods to sample from $q(\theta)$ have been introduced. The pioneering work by Neal,[9] employing the MCMC variant Hamiltonian Monte Carlo, is currently considered the *gold standard*, but its applicability is limited to small networks and data sets. Stochastic and optimized variations enhance scalability at the expense of approximation performance.[32,33]

Variational inference (VI) is an alternative paradigm to derive $q(\theta)$. In this case, a class of approximating distributions $q_\phi(\theta)$ parameterized by ϕ is explicitly chosen so that posterior approximation becomes an optimization problem of finding ϕ minimizing the Kullback–Leibler divergence with respect to $p(\theta|\mathcal{D})$.

VI methods constitute a standard technique in Bayesian modeling. However, scalability requirements and NN-specific features have led to the design of new methods.[10,15,34−36] Nonetheless, some of these approaches—such as Stein variational gradient descent[36]—do not actually scale up to training-intensive applications such as active learning-based molecular property prediction.[7] In this work, we target *scalable* and *practical* methods, which could easily be used on large data sets/networks and in applications such as active learning.

MC-dropout and ensembling-based methods are currently the most popular approaches for large-scale uncertainty estimation in NNs[17] and, within chemistry, both have been very recently introduced.[7,11,37−39] In addition to their scalability, these methods owe their popularity to the relative ease of implementation. For this reason, in the following, we will focus on MC-dropout and ensembling, describing both the original methods, main variations (in particular, bootstrapping), recent improvements, and interpretations.

Monte Carlo Dropout. MC-dropout[6,10] is a simple and scalable VI approach. The algorithm consists in training a network with dropout before every layer and then, at testing time, keeping dropout to sample M outputs $\mathbf{y}^{(i)}$ with different random masks. Each different random dropout mask corresponds to a sample from the approximate posterior $q_\phi(\theta)$. The model prediction $\tilde{\mathbf{y}}$ is the mean of the different outputs, while the epistemic uncertainty σ_e^2 can be captured by the variance of the different outputs. If the aleatoric uncertainty is also computed (as in Figure 2), the output aleatoric uncertainty is the mean of the different aleatoric uncertainty estimates (and, in this case, the $\mathbf{y}^{(i)}$ are substituted by the $\boldsymbol{\mu}^{(i)}$)

$$\tilde{\mathbf{y}} = \frac{1}{M}\sum \mathbf{y}^{(i)} \quad \sigma_e^2 = \mathrm{var}(\mathbf{y}^{(i)}) \quad \sigma_a^2 = \frac{1}{M}\sum \sigma_a^{(i)} \tag{5}$$

Formally, the MC-dropout algorithm approximates the posterior with a product of Bernoulli distributions. Indeed, given a dropout probability p, each unit of the network with parameters θ_i has probability p of being dropped and set to zero. Equivalently, the approximation distribution can be seen as a mixture of two Gaussians with small variances.[6,10]

A drawback of the MC-dropout approach is the introduction of the dropout rate p as hyperparameter. Such a choice has an important impact both on the model's accuracy and uncertainty estimation. Indeed, p contributes to determine the magnitude of the epistemic uncertainty. Moreover, this hinders model hyperparametrization, especially if p is chosen to be layer-dependent.

Concrete dropout[40] represents a practical gradient-based solution to automatically tune p. This approach has comparable performance as grid-searched p[40] and better model calibration than standard MC-dropout.[20] Therefore, we will compare this nonparametric version of MC-dropout to the intrinsically nonparametric ensembling approach.

Ensembling. Ensembling has been introduced as a practical non-Bayesian alternative to estimate uncertainty with the name *Deep Ensembles*.[12] The algorithm consists in training the same network multiple times with random initialization, minimizing the MLE objective $-\log p(\mathbf{y}|\mathbf{x},\theta)$ each time. The output of the ensemble is given by the mean of the predictions, while the variance corresponds to the ensemble uncertainty, as shown in eq 5 for MC-dropout.

It is possible to draw a parallel between ensembling and MC-dropout because the latter can also be interpreted as a form of ensembling.[12,41] Even if ensembling has been originally proposed as a non-Bayesian solution,[12] recent literature has proved how, with minor modifications to the original ensembling methodology, it is possible to interpret it as a Bayesian inference technique.[15,16] Nonetheless, even without the modifications, ensembling can be interpreted as Bayesian approximation with an implicit distribution $q(\theta)$.[17]

Ensemble methods have long been recognized as very effective to improve predictive performance of machine learning[42] and deep learning models,[43] and also for chemistry

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

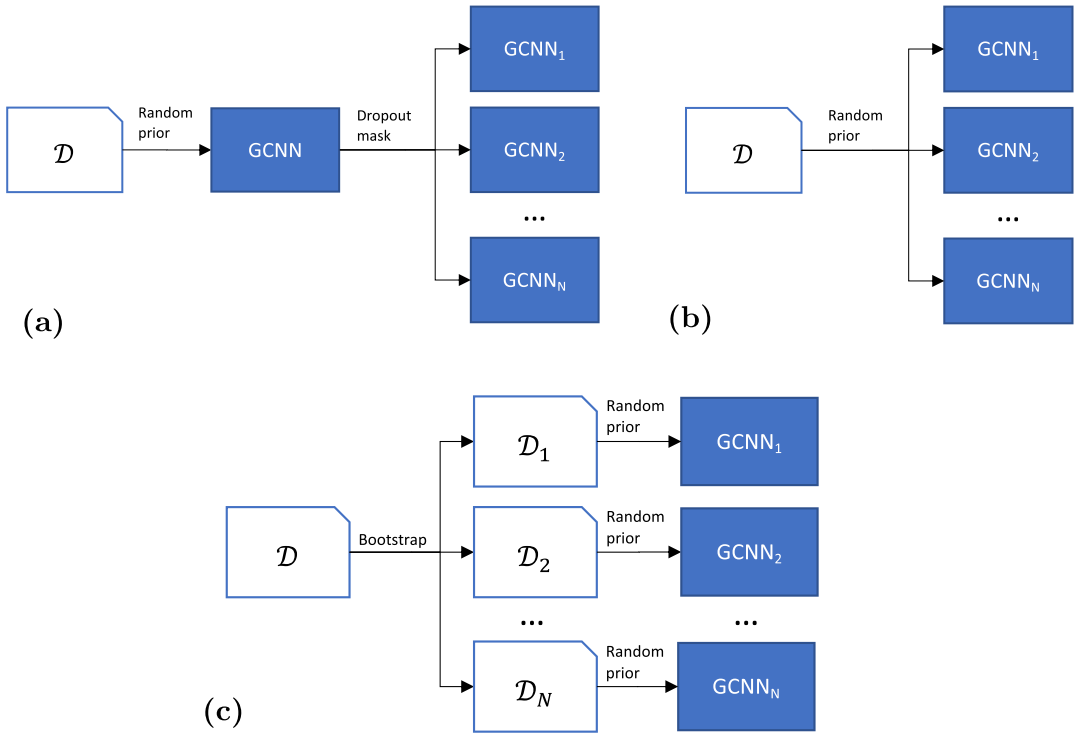


Figure 3. Overview and comparison of MC-dropout, ensembling, and bootstrapping. $\mathcal{D}$ is the training data set. (a) MC-dropout: only a network is trained to minimize the loss on the training data set. Then, at testing time, multiple models are "generated" applying a stochastic dropout mask to the initial network. All the models $GCNN_1$, ...$GCNN_N$ share (part of) the same weights. Diversity in the models is the result of dropout masks. (b) Ensembling: different models are trained to minimize the loss on the same training data set. Diversity in the models results from different initial configurations (random priors). (c) Bootstrapping: each model is trained to minimize the loss on a bootstrap sample of the training data set. This, together with different initial configurations (random priors), ensures diversity in the models.

in QSPR.[1] The reason why ensembling allows reducing the overall error with respect to each of components resides in the diversity of their errors. Indeed, perfectly correlated errors do not bring any advantage to the ensemble error, while perfectly uncorrelated errors reduce the expected ensemble error proportionally to the number of employed instances.[43] Different solutions can be easily reached by deep models given their nonconvexity and the suboptimal optimization strategies employed.

The intuition behind the interpretation of the ensemble variance as model uncertainty is simple. Different instances of the ensemble of models will tend to output similar values when the inputs are similar to the observed training data because each instance'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 instance tend to be more affected by the specificities of the suboptimal solution reached, thus the higher variance. Given this, it seems clear that diversity in the ensembled models should be promoted both for error reduction and uncertainty improvement.

Traditional regularization techniques, such as weight decay and early stopping, affect the solutions reached by NNs. Recently, the usage of these techniques has been proposed not only as a practical strategy to increase ensemble diversity but also as a formal evidence for a Bayesian interpretation of ensembling.[15,16] This is discussed in detail in Supporting Information (see section Anchored Ensembles and Early Stopping).

Bootstrapping. Also referred to as bagging, bootstrapping is a popular technique where ensemble members, instead of being trained on the whole data set, are trained on different *bootstrap samples* of the original training set. Each bootstrap sample $\mathcal{D}_i$ is obtained by sampling K samples with replacement from the data set $\mathcal{D}$ and therefore will include a fraction of the elements in $\mathcal{D}$ and duplicates. If the original data set is a good approximator of the underlying distribution, each $\mathcal{D}_i$ will also be.

Bootstrapping allows increasing the diversity in the trained instances, which, as previously discussed, is a key factor for ensembling performance. However, instead of relying on diversity in the models, bootstrapping relies on diversity in the data sets. This approach has been successfully employed to increase the diversity in shallow ensembles, but its use within NNs might be less beneficial because given the dependence on a large amount of training data, each individual instance will be less powerful.[12] Moreover, NNs are characterized by an extremely large amount of equivalent local minima[43] and this, together with stochastic gradient descent optimization, should already provide some degree of diversity even when trained on the same data set.

Nonetheless, because bootstrapping has been recently described in the literature as an effective approach for NNs,[37,39] below we compare it to the full-data set ensemble method.

A comparative overview of MC-dropout, ensembling, and bootstrapping is presented in Figure 3. As shown, each method relies on a set of predictions (explicit or implicit models). The different predictions are used to estimate epistemic uncertainty, as shown in Figure 4.

Total Uncertainty. Aleatoric and epistemic uncertainty can be added to approximate the *total uncertainty* of a prediction,[6,12]

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

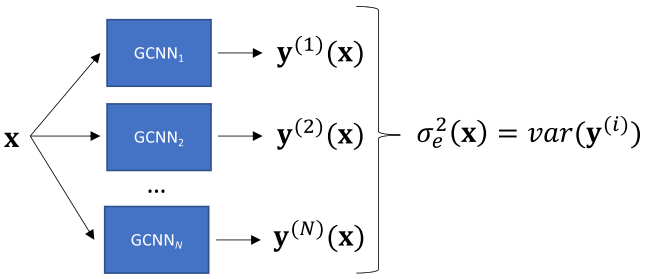


Figure 4. Epistemic uncertainty computation. Independently from the method used to obtain multiple models (see Figure 3), the epistemic uncertainty is estimated as the variance of the different outputs (ref. eq 5).

including both the uncertainty in the model's prediction of μ and model-data deviations coming from noise in the data.

For this work, we focus on three scalable uncertainty frameworks for predicting total uncertainty: (1) MC-dropout using concrete dropout (2) ensembling, and (3) bootstrapping. All the different methods use the same aleatoric approximation scheme previously described, but the different ways epistemic uncertainty is modeled also affects the aleatoric uncertainty results, thus resulting in different outputs (ref. eq 5).

Hyperparameters. Uncertainty estimates obtained through DNN-based methods are affected by the choice of training hyperparameters. Some uncertainty estimation methods introduce additional hyperparameters, but the methods used here do not, with the exception of the sampling size M.

Hyperparameter optimization for the base network is performed using the hyperopt package (https://github.com/hyperopt/hyperopt). The impact of hyperparameters on uncertainty estimates is further discussed in Supporting Information along with more details about how we tuned them (see section Hyperparameters).

Evaluating the Quality of Uncertainty Estimates. Evaluating the quality of uncertainty estimates is tricky because (1) users have different objectives, and (2) usually, the true uncertainties are unknown. Here, we use several evaluation methods.

Ranking-Based Methods. A first class of evaluation indices is based on the ranking defined by uncertainty estimates. This allows defining a *confidence curve*, which, in turn, allows defining several quantitative indices.

Confidence Curve. One way to evaluate the uncertainty is by considering how the error varies as we remove molecules with the highest uncertainty in the test data set. Indeed, a meaningful uncertainty should lead to a lower error on a subset of high-confident predictions. This concept is captured by the *confidence curve*, that highlights how the error varies [with respect to a given metric, e.g., mean absolute error (MAE) or root mean squared error (RMSE)] as a function of confidence percentile (or, in general, confidence q-quantile), that is, the error on the subset of n % molecules (or n-th q-quantile) with the lower uncertainty.

Ideally, we would expect a decreasing confidence curve. The error corresponding to the left-most point is simply the error on the complete test data set; the following points correspond to the error on the subset of testing molecules belonging to the n-th q-quantile. A better uncertainty estimate gives a confidence curve with a higher slope because it allows decreasing the error faster for the same amount of removed molecules. For comparison, randomly sampling the molecules to be removed should lead to a more or less constant function.

What this kind of evaluation really assesses is the *ordering* of the predictions by their confidence. From this perspective, the best possible ordering is the one imposed by the true error, giving the *oracle confidence curve*.[19]

Confidence-Oracle Error and AUCO. Because the oracle ordering corresponds to the lower bound, we can define the confidence-oracle error as the difference between the confidence curve for a given uncertainty estimation, $h^{(i)} = (h_1^{(i)}, h_2^{(i)}, ...h_{q-1}^{(i)})$, and oracle confidence curve, $h^{(o)} = (h_1^{(o)}, h_2^{(o)}, ...h_{q-1}^{(o)})$. In general, we want this difference to be as small as possible, and therefore, we introduce the *area under the confidence-oracle error*, AUCO, to quantify it in a single number[a]

$$\mathrm{AUCO}(h^{(i)}) = \sum_{j=1}^{q-1} (h_j^{(i)} - h_j^{(o)}) \tag{6}$$

This value allows an easy comparison between two uncertainty estimations $h^{(i)}$ and $h^{(j)}$ with respect to the oracle, where smaller is better.

Every confidence curve depends not only on the uncertainty estimation but also on the predictive model: the first defines the q-quantiles, and the second is used to calculate the errors. It is not possible to directly compare two confidence curves obtained through different models to establish which uncertainty estimation is better. This is particularly relevant because often the uncertainty estimation and the predictive model are strongly tied: for example, ensembling is an uncertainty technique that also affects the predictions. An added benefit of the confidence-oracle error is that because it marginalizes out the oracle, it enables a fair comparison of uncertainty estimates based on different methods.[19] Therefore, the confidence-oracle error and the AUCO will be used in the following for this purpose.

Notice that, using q-quantiles, each uncertainty-imposed ranking that does not change the specific quantile each prediction belongs to, even if it does change the relative position of the q predictions inside each quantile, is equivalent from the point of view of the confidence curve, the confidence-oracle error and the AUCO. These are all affected by the choice of q. In the following, we will use percentiles as commonly reported in the literature.

Error Drop. Error drop is defined as the error ratio between the first and last quantiles, which should correspond to the confidence curve's maximum and minimum, respectively, if the confidence curve behaves correctly

$$\text{Error drop } (h^{(i)}) = \frac{h_1^{(i)}}{h_{q-1}^{(i)}} \tag{7}$$

This index measures the relative performance improvement of the model obtainable by considering only the most confident predictions instead of the entire data set.

Decreasing Coefficient. A limitation of the AUCO and error drop indices is that they do not take into account the monotonicity of the confidence curve, which is represented by the decreasing ratio. Given a confidence curve $h^{(i)} = (h_1^{(i)}, h_2^{(i)}, ...h_{q-1}^{(i)})$

$$\text{Decr. ratio } (h^{(i)}) = \frac{\left|\{h_j^{(i)} | h_j^{(i)} \geq h_{j+1}^{(i)}\}\right|}{q-1}$$
$$\forall\, j \in 1, ..., q-2 \tag{8}$$

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

where decr. ratio = 1 corresponds to a perfectly nonincreasing curve.

This ratio captures the noise in the confidence curve.

Uncertainty Calibration Methods. One limitation of the ranking-based methods is that they do not take into consideration the *actual values* of the uncertainty estimates. Indeed, another important aspect of uncertainty is more strictly related to the actual values it expresses and referred to as *calibration*. Calibration of a model refers to the property of outputting probability distributions that are consistent with observed empirical frequencies.

Calibration evaluation of neural networks gained interest in the last two years because it has been shown that modern neural networks, while being more accurate on one side, are less calibrated on the other,[18] thus encouraging more research on the topic.[6,12] Indeed, model calibration is orthogonal with respect to model accuracy.[12] Calibrated confidence is important for model interpretability and to establish trustworthiness with the user.[18]

Uncertainty calibration is a well-studied topic in the context of classification,[44] both in its traditional domain of weather forecasting[45] and, more recently, in deep learning.[18] In this context, a model is perfectly calibrated if the confidence assigned to each class is equal to the probability of a prediction of belonging to that specific class. In practice, over a finite number of samples, calibration can be captured by a *calibration plot*,[6] also called *reliability diagram*.[18] To obtain such a plot, the model predictions for all samples and classes in the test set are split into K bins in the range $[0, 1]$ and the frequency of correctly predicted labels for each bin is plotted.[44] Perfect calibration corresponds to a *diagonal line*.

Calibration can vary within the same uncertainty estimator when considering different uncertainty intervals. This could happen, for example, if a model has well-calibrated low uncertainty but ill-calibrated high uncertainty, or vice versa. Such cases are highlighted by a calibration plot that diverges from the diagonal line in some specific confidence intervals but not in others.

Calibration in Regression. Calibration for regression appears to be less investigated, and different solutions to evaluate it have been employed and discussed only recently.[6,17,21,22] In the following, we will consider two different definitions that extend calibration in a regression setting: *confidence-interval*-based and *error*-based calibration.

- Confidence-based calibration (also called interval-based calibration)[17,21] interprets each prediction and its uncertainty as the mean and the variance of a Gaussian distribution $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$, and we are interested in evaluating the confidence intervals thus defined. To do so, we consider symmetric intervals of varying confidence around the mean and compare them to the empirical probabilities of belonging to each interval. In a well-calibrated model, x % of the predictions should fall in the x % confidence interval. In practice, we discretize the confidence intervals and calculate the fraction of predictions falling in each interval. This allows obtaining a calibration plot in the $[0, 1]$ range, as in the classification case, where perfect calibration corresponds to a diagonal line.

- Error-based calibration, originally described by Levi et al.,[22] proposes to directly compare the uncertainty to the empirical error, as in eq 9.

$$\mathbb{E}[(\mu(\mathbf{x}) - \mathbf{y})^2 | \sigma^2(\mathbf{x}) = \tilde{\sigma}] = \tilde{\sigma}, \qquad \forall\, \tilde{\sigma} \tag{9}$$

  - This defines a perfectly calibrated model as one outputting an uncertainty matching the expected error. In practice, to assess calibration, it is necessary to split the test data ordered by estimated uncertainty in K bins and average uncertainties and errors for each bin. It is then possible to define the calibration curve by plotting the RMSE of the i-th bin as a function of its root mean uncertainty, $\sqrt{\mathbb{E}(\sigma^2)}$. Notice that, unlike classification and confidence-interval calibration cases, here, the calibration plot is not bound in the $[0, 1]$ interval but ranges between 0 and the maximum uncertainty. As in the other cases, perfect calibration corresponds to a diagonal line.

These two ways of constructing the calibration plot have pros and cons. Confidence-based calibration has the advantage of considering all the predictions to compute each point of the plot, thus resulting in more robust empirical calculations. However, as recently highlighted,[22] one can recalibrate practically any output distribution using this evaluation method. The main advantage of error-based calibration is that it directly ties computed uncertainty to expected error, as a user would expect. However, because only a fraction of uncertainty estimates contributes to each computed point and the uncertainty estimates are not uniformly distributed, the subsets used to compute the different points are not homogeneous.

Calibration Error Curve and AUCE. We can quantitatively evaluate uncertainty calibration by computing the absolute difference of the calibration plot with respect to perfect calibration, thus obtaining the *calibration error curve*. This difference can be quantified by considering the area under this curve, which has been referred to as the *area under the calibration error curve*, AUCE metric.[17] This is a cumulative metric accounting for the total calibration error.

ECE, MCE, and ENCE. Rather than considering the total error, it is possible to define the *expected calibration error* (ECE) and the *maximum calibration error* (MCE).

For confidence-based calibration

$$\mathrm{ECE} = \frac{1}{K}\sum_{i=1}^{K} |\mathrm{acc}(i) - i| \quad \mathrm{MCE} = \max_{i=1}^{K}(|\mathrm{acc}(i) - i|) \tag{10}$$

where $i \in K$ is a confidence interval and $\mathrm{acc}(i)$ is the fraction of times a prediction falls into the i-th confidence interval. ECE and MCE correspond to the average and the maximum over the calibration error curve, respectively. MCE is especially important in high-risk applications because it models the *worst-case scenario*.[18]

For error-based calibration, a variation of ECE called *expected normalized calibration error* (ENCE) is defined[22]

$$\mathrm{ENCE} = \frac{1}{K}\sum_{i=1}^{K} \frac{|\mathrm{mVAR}(i) - \mathrm{RMSE}(i)|}{\mathrm{mVAR}(i)} \tag{11}$$

where $\mathrm{mVAR}(i)$ is the root of the mean uncertainty over the i-th bin and $\mathrm{RMSE}(i)$ is the root mean square error over the i-th bin. This discrepancy is further normalized by the uncertainty over the bin, $\mathrm{mVAR}(i)$, because the error is expected to be naturally higher as the uncertainty increases.[22]

Sharpness and Dispersion. Calibration is insufficient to fully evaluate an uncertainty estimator. Indeed, if the model always

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 1. Summary of the Data Sets and Split Types for the Experiments

					split	
data set	category	size	property	metric	in-domain	out-of-domain
QM9	quantum chemistry	130,828	enthalpy [kcal·mol^{-1}]	MAE	random	scaffold
Alchemy	quantum chemistry	103,657	heat capacity [cal·(mol·K)$^{-1}$]	MAE	random	size
PDBbind	biophysics	11,908	protein binding affinity [$-\log(K_d/K_i)$]	RMSE	random	time (publication date)
Lipophilicity	physical chemistry	4200	octanol/water distribution coefficients [log D]	RMSE	random	chemical element (contains F)

outputs the *same constant uncertainty* which matches the empirical accuracy over the entire distribution, we obtain a perfectly calibrated uncertainty but not a very useful one because it does not depend on the input data at all. This concept is captured by *sharpness*, an uncertainty's property orthogonal and complementary to calibration.[46]

Originally defined in the classification settings, this notion has been recently extended for regression.[21,22] Following the definition introduced in Levi et al.,[22] in the following, the *dispersion* of an uncertainty estimator is defined as the *coefficient of variation* c_v of its uncertainty estimates. A higher c_v corresponds to more heterogeneous estimates for different inputs.

Because uncertainty estimates are often characterized by very high/low values for specific molecules, in the following we use a modified version of the coefficient of variation more robust to outliers, where the standard deviation and the mean are substituted by the *interquartile range* and the *median*, respectively.

Dispersion represents a useful metric to be taken into account along with calibration when comparing different methods. In particular, we are interested in verifying that an improvement in calibration of an uncertainty estimator with respect to another one does not originate from a reduction in dispersion.

To the best of our knowledge, dispersion has not been taken into account before in comparative evaluations of deep learning uncertainty estimation frameworks[17−19,47] or in the context of deep molecular property prediction.[7,11]

Domain Shift. An important feature that should characterize a well-behaving uncertainty estimator is its ability to correctly manage *domain shifts*, when the test set is markedly different from the training set. Every data-driven model will degrade at some point on unseen samples as they become more different from those seen during training, but a well-calibrated uncertainty should be able to correctly identify this "knowledge boundary".

The need for DNN-based uncertainty estimates which are reliable over domain shifts has been highlighted in other contexts,[12] but it is even more important in the chemical domain. Indeed, generalization power is a requirement in key applications such as drug discovery, and the intrinsic high variability of chemical space makes it challenging to fulfill this requirement. Despite this prominent role, the evaluation of out-of-domain DNN-based uncertainty performance in the chemistry field appears to be absent[7] or very limited,[11] thus demanding a more extensive analysis.

This analysis is also related to an important issue in QSAR, that is, the definition of the *domain of applicability* of a model.[23,24] We are interested in evaluating if the tested uncertainty estimation methods can help define and/or extend the applicability domain of DNN-based models.

To achieve this goal, we test each target data set under two different settings. First, we use an *in-domain* test set obtained by splitting the entire data set randomly so that training and test distributions are comparable. Then, we use a non-random *out-of-domain* test set.

In general, the definition of "out-of-domain" is not unambiguous, especially in chemistry. As explained next, in this work, we use different split types to model out-of-domain test sets: *scaffold splitting*, *size splitting*, *time splitting*, and *chemical element splitting*.

We are interested in *re-evaluating* all the already introduced metrics—AUCO, AUCE, and so forth—in the out-of-domain setting. We will pay particular attention to *out-of-domain calibration* because it can measure to what extent a model knows what it does not know. We are interested in quantifying the ratio between in-domain and out-of-domain metrics, also in relation to the ratio between in-domain and out-of-domain errors.

NUMERICAL EXPERIMENTS

We first describe the target data sets, followed by a description of the experimental procedure.

Data Sets. Numerical experiments have been carried on several public data sets spanning different categories, properties, and size.

We selected the largest regression data set for each category in MoleculeNet:[2] QM9 (quantum mechanics), Lipophilicity (physical chemistry), and PDBbind (biophysics). We also included the recently published Alchemy data set[48] for quantum chemistry data. Based on previous works, we use the MAE or RMSE error metrics for each data set.[2,48]

To analyze *domain-shift* performance of the tested uncertainty estimation methods, we compared *in-domain* and *out-of-domain* metrics on the same data sets by changing the training/testing splits. While in-domain metrics are always evaluated through random splitting, the splits used for out-of-domain metrics are data set specific because we have relied on what has been already published and used for each data set, when possible. In particular, for QM9, we employ the recently introduced *scaffold splitting* technique:[1,2] molecules are split into bins based on their Murcko scaffold, with each bin belonging to only one among training, validation, and test set. For PDBbind, because it comes with time information, we use *time splitting*, as described in Wu et al.[2] For Alchemy, we use *size splitting*, as described in Chen et al.,[48] where a model trained mostly with smaller-sized molecules is used to predict molecules of bigger size. Last, for Lipophilicity, we introduce a *chemical element split* where a chemical element (fluorine, in our case) is only contained in test molecules and not included in training molecules.

Interested readers are referred to Supporting Information, section Data Preparation, for more details about how each data set was prepared. In total, we use four different data sets and split types; see Table 1.

Experimental Procedure. We evaluated the uncertainty estimation techniques previously reviewed (MC-dropout, ensembling, and bootstrapping) using the evaluation criteria

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 2. Error on the Test Data Sets[a]

	base model		MC-dropout		Deep Ensembles		bootstrapping	
	in	out	in	out	in	out	in	out
QM9	1.04	1.77	0.97	1.49	0.74	1.21	0.89	1.43
Alchemy	0.40	0.89	0.37	1.02	0.32	0.71	0.34	0.87
PDBbind	1.38	2.26	1.37	2.23	1.31	2.10	1.33	2.15
Lipophilicity	0.576	0.687	0.548	0.655	0.481	0.616	0.495	0.626

[a]Results are shown for the base model without any uncertainty estimation and for the model extended with each of the evaluated uncertainty estimation methods, for each data set. Both in-domain and out-of-domain performance are reported for each case. Error metric provided in Table 1.

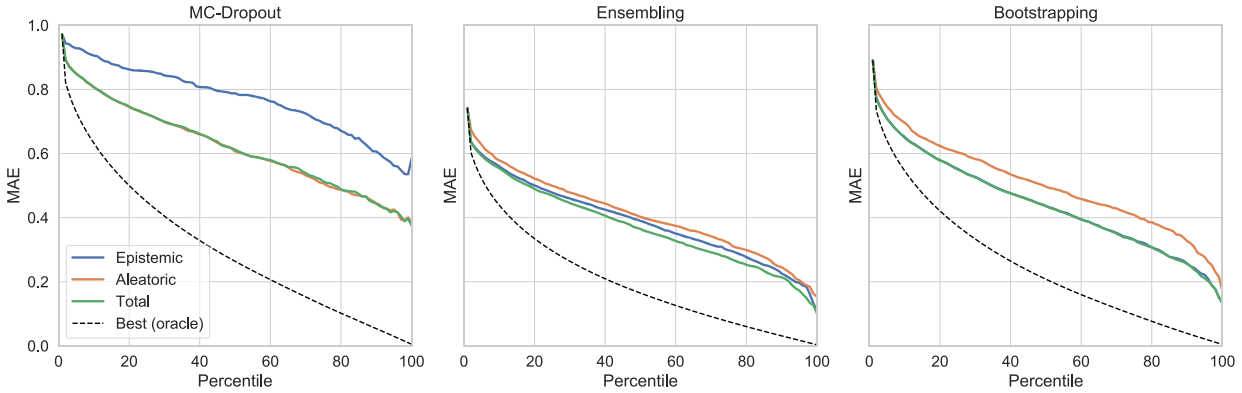


Figure 5. QM9, in-domain test set. Confidence curves for the different methods.

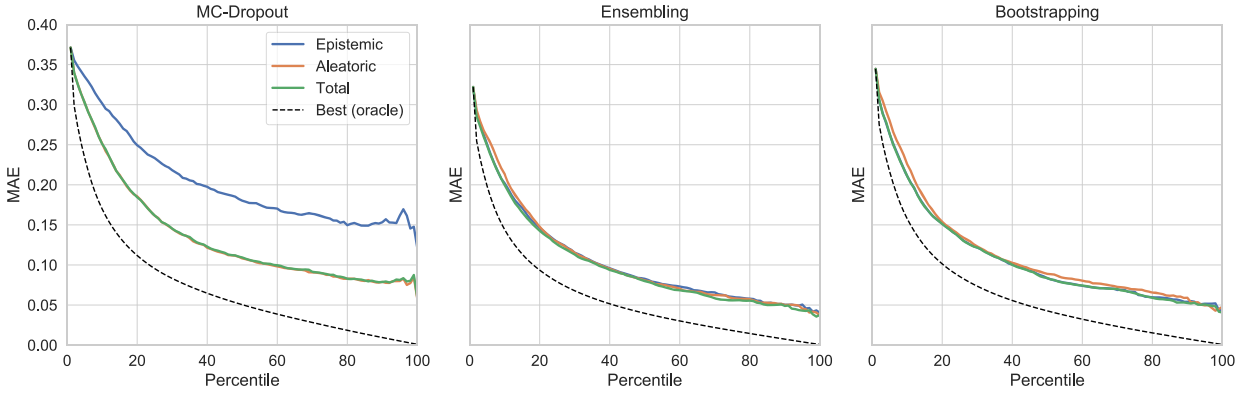


Figure 6. Alchemy, in-domain test set. Confidence curves for the different methods.

previously introduced. Other than including diagrams, we evaluated the considered methods quantitatively, as follows:

- For ranking-based evaluation, we use the area under the confidence-oracle error (AUCO) as a measure of total discrepancy with respect to the best possible ranking, the *error drop* as a measure of total error reduction for high-confident predictions, and the *decrease ratio* to assess the monotonicity of confidence curves.
- For confidence-based calibration, we use the area under the calibration error curve (AUCE) as a measure of total discrepancy with respect to perfect calibration and the MCE to account for the worst-case scenario[b].
- For error-based calibration, we use the ENCE as a measure of the (normalized) total discrepancy with respect to perfect calibration.
- For dispersion evaluation, we use the quartile-based coefficient of variation, c_v.
- For domain-shift performance, we evaluated and compared all the above metrics also in an *out-of-domain* setting (see Table 1).

Additional details about the experimental procedures are provided in Supporting Information.

RESULTS

We first detail error performance for the considered models, data sets, and splits. Next, we present results for uncertainty estimation evaluation.

Model Error Analysis. Table 2 lists the error on the test data sets for each uncertainty estimation method, with in-domain and out-of-domain splits.

The baseline is the chemprop model[1] without any uncertainty estimation. We notice how extending it to include uncertainty always leads to reductions in error for ensembling and bootstrapping and, in most of the cases, for MC-dropout. These improvements, often underestimated, are due to both aleatoric and epistemic estimation in the model. Indeed, modeling aleatoric uncertainty implicitly reduces the impact of noisy training samples, thus improving predictive performance. Modeling epistemic uncertainty allows averaging multiple weight configurations, avoiding overfitting, and overconfident

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

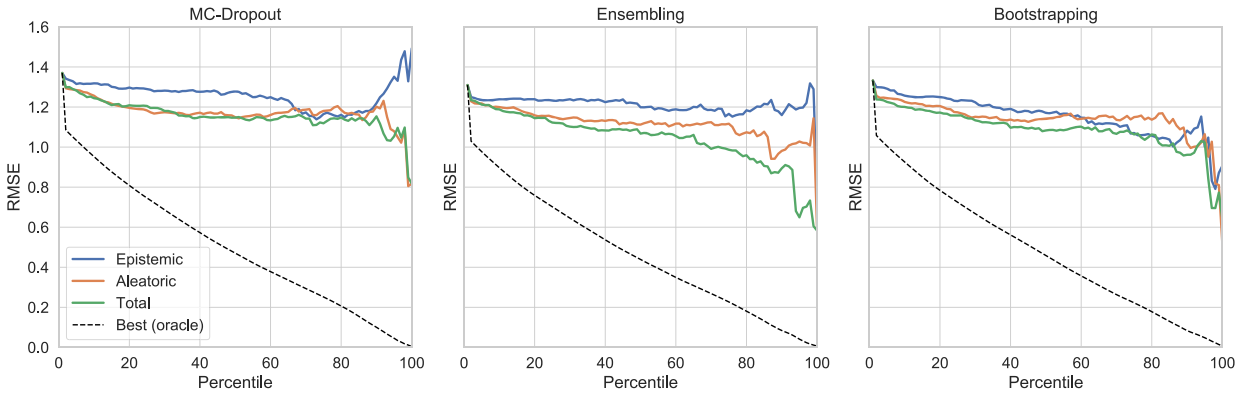


Figure 7. PDBbind, in-domain test set. Confidence curves for the different methods.

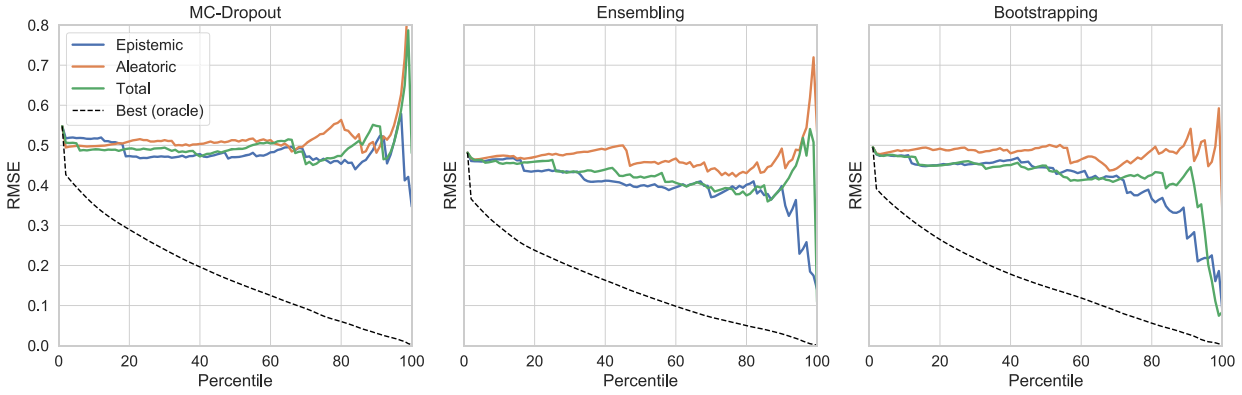


Figure 8. Lipophilicity, in-domain test set. Confidence curves for the different methods.

estimations, with a positive impact on predictions and generalization ability. These two contributions can independently reduce the overall error but act synergistically when both are present.

Comparing uncertainty estimation methods, ensembling results in the lowest error, followed by bootstrapping, and then MC-dropout.

Uncertainty Estimation Evaluation. In the following, resulting plots are shown and discussed for each evaluation criterion. Additional plots are available in Supporting Information. All the *summary tables*, which include quantitative results, are available in Supporting Information (Tables S1−S4).

Ranking-Based Evaluation. The confidence curves for the different methods and data sets are shown in Figures 5−8. The related confidence-oracle error diagrams are included in Supporting Information. The derived AUCO and decrease ratio metrics for each case are reported in the first two lines of the summary tables (Supporting Information).

For QM9 and Alchemy, the confidence curves are mostly decreasing for all the considered methods. For QM9 [Alchemy], decreasing ratio $\geq$ 0.99 [0.90] for ensembling and bootstrapping, $\geq$0.95 [0.80] for MC-dropout. This means that each method can establish a qualitatively meaningful ranking of the predictions by their uncertainty, leading to errors up to 7−8 times lower than the overall test error for the highest percentiles. Confidence curves of MC-dropout are more noisy and lead to a lower error reduction—both relative (higher AUCO) and absolute (higher MAE)—with respect to the other two methods, especially for epistemic uncertainty. For all methods, total uncertainty leads to better or comparable results with respect to the best performing uncertainty contribution alone. Compared to bootstrapping, ensembling leads to slightly higher performance and allows reaching the lowest MAE in the highest percentiles for both uncertainty types and total uncertainty. Interestingly, the epistemic uncertainty estimated by bootstrapping results in an MAE comparable to ensembling in the highest percentiles, even if the initial MAE on the whole data set is worse. This is quantitatively measured by a higher or comparable error drop for bootstrapping. In contrast, aleatoric uncertainty estimated by bootstrapping leads to a worse performance than ensembling. Comparing QM9 and Alchemy, confidence curve shapes turn out to be markedly different.

Considering PDBbind and Lipophilicity, confidence curves are more noisy and their slope less steadily decreasing, with a decreasing ratio $\leq$0.50 in all cases. Bootstrapping has the best decreasing ratio for both uncertainty types and on both data sets.

On PDBbind, comparing individual uncertainty components, we observe that aleatoric uncertainty leads to a better AUCO than epistemic uncertainty for all methods. For MC-dropout and ensembling, epistemic uncertainty alone does not lead to a significant error reduction in the respective confidence curves, while for bootstrapping, we observe a more steady RMSE decrease. Aleatoric uncertainty performs similarly (AUCO and decreasing ratio) in the three cases. For all the tested estimation methods, we observe that total uncertainty leads to better or comparable results with respect to the best performing uncertainty contribution alone. Interestingly, epistemic uncertainty has a significant positive impact on total uncertainty even when it does not lead to a significant error reduction by itself. See, for example, ensembling confidence curves in Figure 6.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

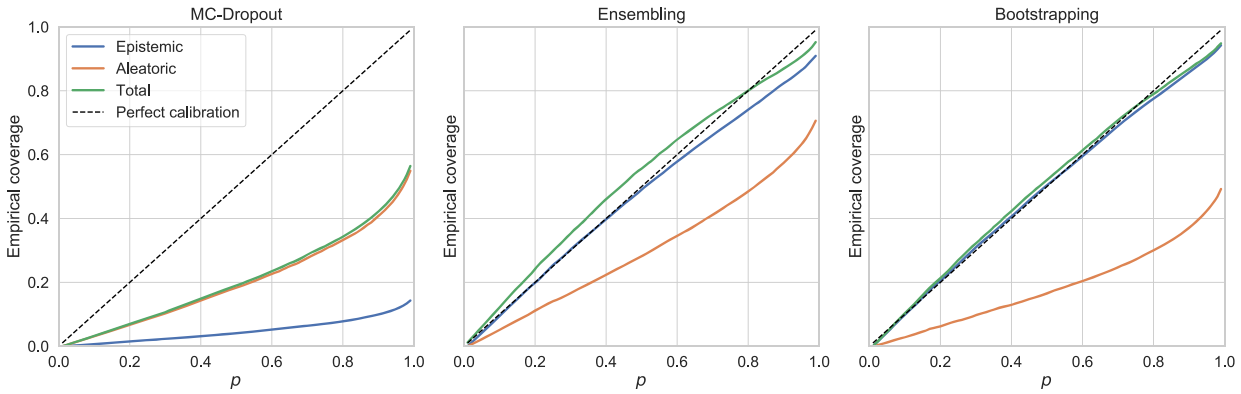


Figure 9. QM9, in-domain test set. Confidence-based calibration for the different methods.

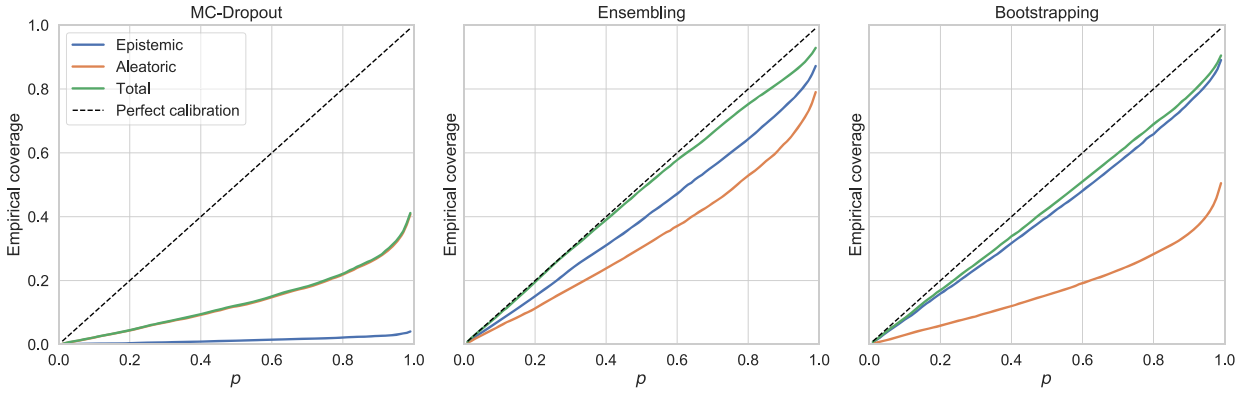


Figure 10. Alchemy, in-domain test set. Confidence-based calibration for the different methods.

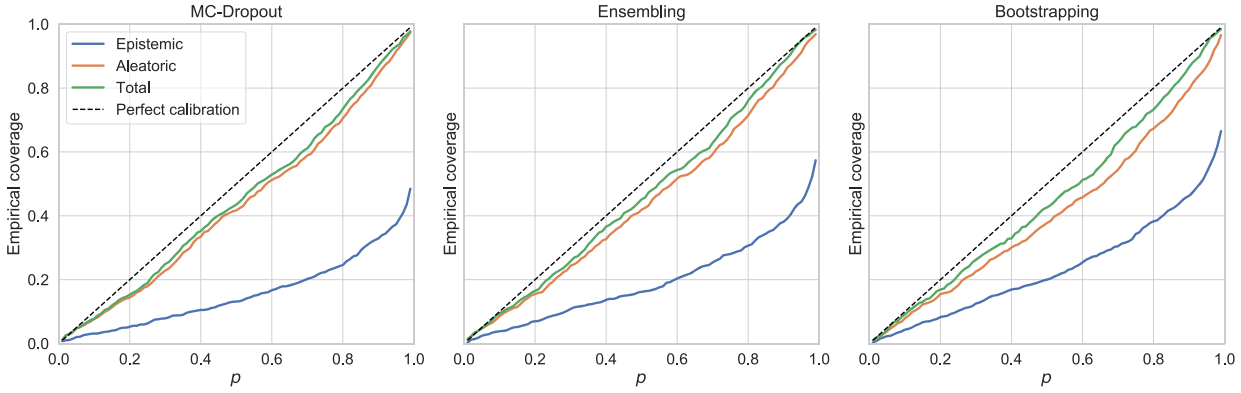


Figure 11. PDBbind, in-domain test set. Confidence-based calibration for the different methods.

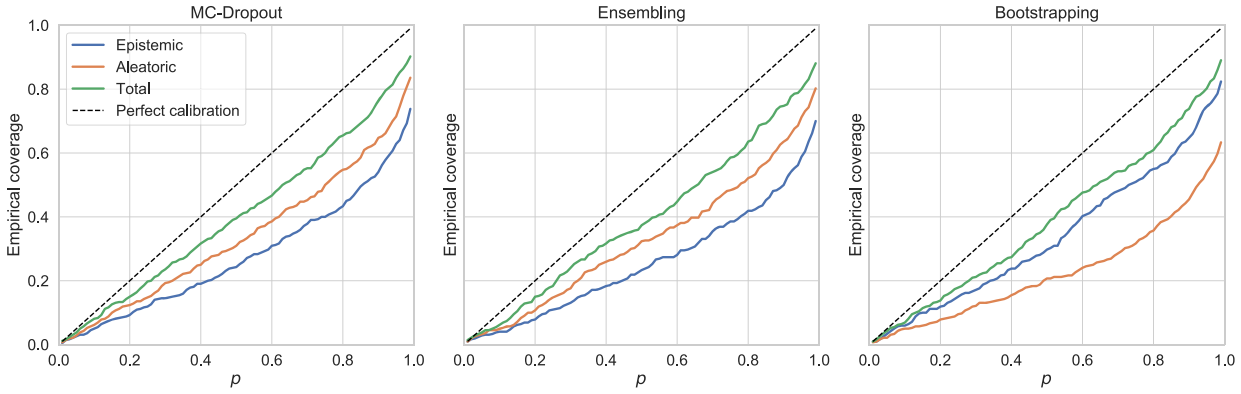


Figure 12. Lipophilicity, in-domain test set. Confidence-based calibration for the different methods.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

The confidence curves for the Lipophilicity data set are very noisy. For this data set, epistemic uncertainty leads to better AUCO and decreasing ratio than aleatoric uncertainty for all the tested methods. As in the other data sets, MC-dropout has the worst epistemic uncertainty performance. Ensembling and bootstrapping result in similar performance (AUCO and decreasing ratio) for epistemic uncertainty. The bootstrapping confidence curve reaches the lowest absolute RMSE and this, together with a higher initial error on the entire data set, results in a far higher error drop than ensembling. For all three tested methods, aleatoric uncertainty alone does not lead to significant error reductions (decreasing ratio ≤ 0.15) and the relative performances are comparable.

Calibration and Dispersion. Confidence-Based Calibration. The confidence-based calibration plots for the different data sets are shown in Figures 9−12. The *empirical coverage*—which is the fraction of times the true value actually falls in a confidence interval—is reported for each symmetric confidence interval of probability p defined by the uncertainty. The derived AUCE and MCE metrics are reported in lines three and four of the summary tables (Supporting Information).

For all of the data sets, ensembling and bootstrapping calibration plots based on estimated total uncertainty are close to ideal. MC-dropout gives good calibration plots with some data sets but shows poor calibration on the QM9 and Alchemy data sets (Figures 9 and 10).

For all the considered data sets and methods, total uncertainty leads to better or comparable AUCE/MCE than aleatoric or epistemic uncertainty contribution alone.

Error-Based Calibration. The error-based calibration plots are shown in Figures 13−16. The derived ENCE is also reported in line five of the summary tables (Supporting Information) and in the figures.

Error-based calibration analysis offers a complementary view of uncertainty performance with respect to the confidence-based analysis already discussed. Instead of considering all predictions simultaneously, each dot only represents a subset of predictions

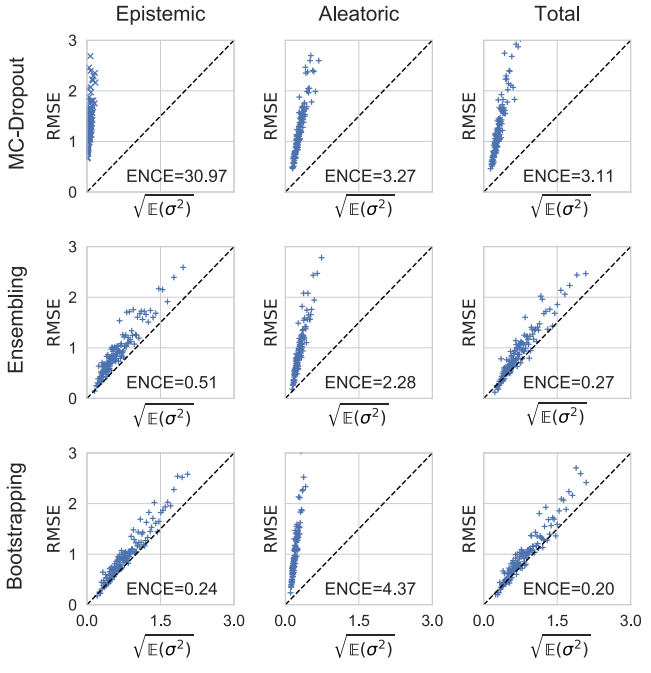


Figure 13. QM9, in-domain test set. Error-based calibration.

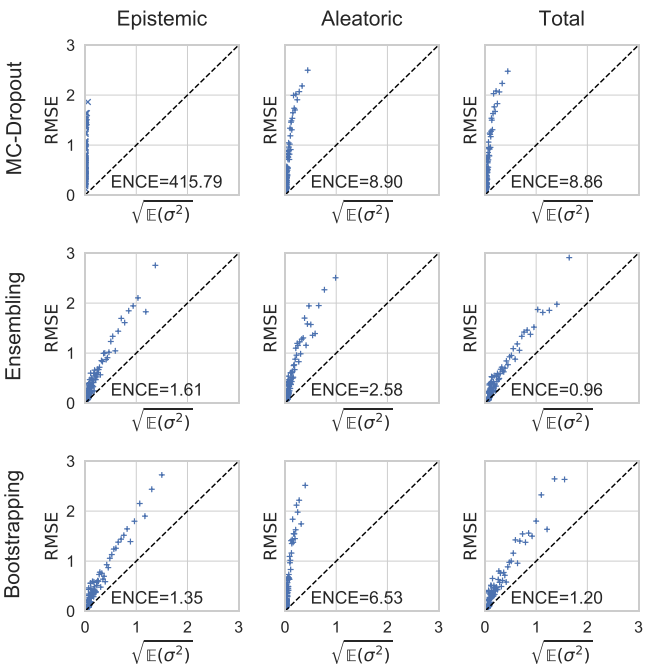


Figure 14. Alchemy, in-domain test set. Error-based calibration.

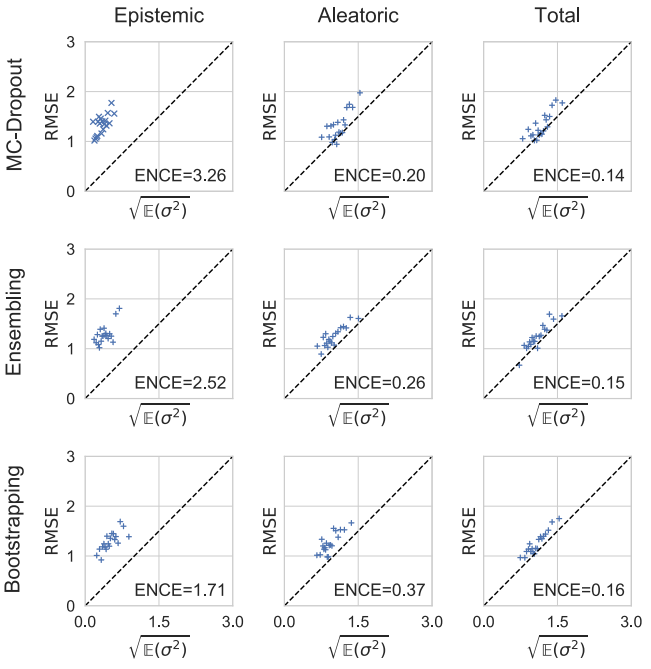


Figure 15. PDBbind, in-domain test set. Error-based calibration.

with similar uncertainty in direct relation with their average error.

Error-based calibration plots confirm what has been already discussed for confidence-based calibration plots: the estimated total uncertainty derived from ensembling or bootstrapping is well calibrated for all the data sets, while the uncertainty estimated from MC-dropout is miscalibrated for the QM9 and Alchemy data sets.

For all the data sets and uncertainty estimation methods, the error-based calibration plots yield strongly correlated patterns, with ensembling and bootstrapping leading to higher correlation than MC-dropout (e.g., on QM9, the Pearson correlation is

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

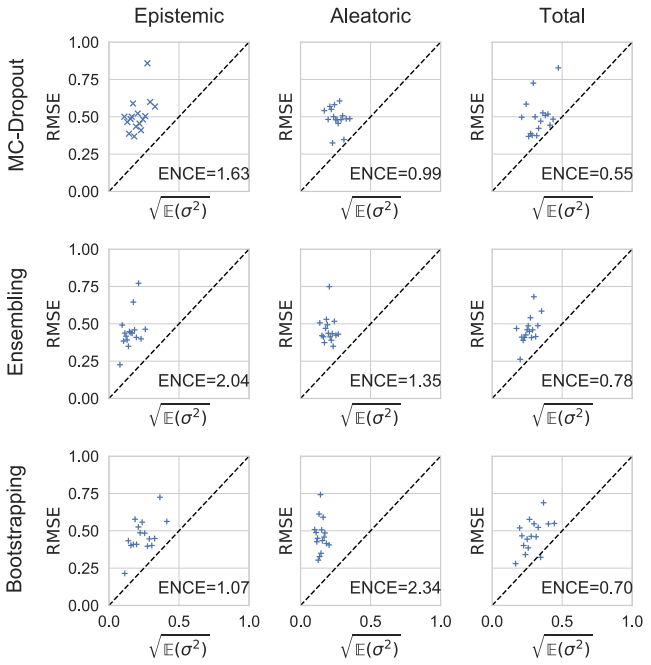


Figure 16. Lipophilicity, in-domain test set. Error-based calibration.

≈0.90−0.93 for ensembling and bootstrapping, ≈0.66−0.87 for MC-dropout). Moreover, even when the error is underestimated, error-calibration plots are often qualitatively characterized by "diagonal" patterns (slope ≈ 1), which correspond to simple translations with respect to perfect calibration (see, e.g., epistemic uncertainties for PDBbind or aleatoric uncertainties for QM9 and Alchemy). Notice that, without introducing error-based calibration plots (with their bins and axis metrics), uncertainty/error correlation is generally low and difficult to be detected reliably.[37]

Compared to confidence-based calibration, this kind of plot is less stable, especially for high values of σ. This is due to (1) the fact that the error is expected to be naturally higher as uncertainty increases (a property already taken into account in the ENCE computation) and (2) the fact that high uncertainty values are more sparse.

Dispersion. The dispersion coefficient is reported in the last line of the summary tables (Supporting Information). Results show no significant variations between the different methods. For all the considered data sets and methods, epistemic uncertainty appears to be more disperse than aleatoric uncertainty.

Out-of-Domain Uncertainty. Figures 17−28 display the same plots already discussed for random splitting but this time for the out-of-domain test data sets. All the related summary tables are available in Supporting Information (Tables S5, S7, S9, and S11). In addition to listing absolute metrics, for the out-of-domain cases, we also listed relative metrics with respect to in-domain test sets (Tables S6, S8, S10, and S12). In the following, the main differences with respect to random splitting are highlighted.

In absolute terms, as expected, uncertainty estimates for out-of-domain molecules are less accurate, so the quality indices have deteriorated for most of the considered methods and data sets. For the most part, the comparative performances of the three uncertainty estimation methods are qualitatively similar with respect to those obtained in the in-domain setting.

For QM9 (scaffold split), domain shift leads to quantitatively worse ranking-based evaluations, for example, a significant increase in AUCO with a decrease in error drop and decreasing ratio, but the plots are similar to those for a random split. In contrast, the calibration analysis highlights an important qualitative change with respect to in-domain results for ensembling and bootstrapping: on this test set, the uncertainty is always markedly underestimated.

Results on Lipophilicity (chemical element split) are qualitatively similar to QM9 in terms of changes in ranking-based metrics and error underestimation. As in the in-domain case, all the plots are more noisy for Lipophilicity than for the other data sets.

Results on PDBbind (time split) are characterized by a peculiar behavior. For all methods, the oracle−confidence curves, instead of converging to ≈0 as in all other considered cases, are significantly higher (RMSE 0.91−1.41) even for the highest percentiles. This means that the models trained on older data do not predict *any* of the newer data points very well. The higher oracle curves lead to lower AUCO in this case: the uncertainty estimates are pretty accurate even though the model is not very accurate for this test set. Out-of-domain calibration analysis confirms the underestimation trend observed for QM9 and Lipophilicity, highlighting an even more drastic change for PDBbind. Indeed, for all methods and uncertainty types, confidence-based calibration plots are characterized by empirical coverage = 0 for $p < 95\%$. Nonetheless, as in the in-domain setting, error-based calibration plots highlight very high uncertainty−error correlation (0.85−0.98 for MC-dropout, 0.93−0.99 for bootstrapping and ensembling) even though the uncertainty estimates are much smaller than the true errors.

Epistemic uncertainty performance for ensembling and bootstrapping on Alchemy (size split) is markedly different from the other data sets. While aleatoric uncertainty estimates degrade for all methods and both from a ranking-based and a calibration-based point of view, epistemic uncertainty estimates turn out to be comparable from a ranking-based point of view and slightly *more calibrated* compared to the in-domain test set. Even though the out-of-domain test error is larger than the in-domain error, the uncertainty estimates for both Deep Ensembles and bootstrapping accurately predict this increase.

DISCUSSION

Comparison of Uncertainty Estimation Methods. Deep ensembles and bootstrapping perform the best across most metrics (including error) and consistently outperform MC-dropout, with Deep Ensembles resulting in the best overall performance. This is in line with recent results[c] already presented for image classification/regression[12,47,49] and optical flow estimation.[17,19]

The comparison between ensembling and bootstrapping raises multiple interesting observations. On the one hand, ensembling has an advantage for overall error, AUCO, and aleatoric calibration, especially in the in-domain setting. On the other hand, bootstrapping often leads to higher error drops and decreasing ratios (i.e., more stable confidence curves which also lead to a lower error, in proportion, when we consider small percentages of high-confidence predictions), has a consistent advantage for in-domain and out-of-domain epistemic uncertainty calibration (which, in some cases, translates to better total uncertainty calibration), and shows, in proportion, a lower performance loss in the out-of-domain setting.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

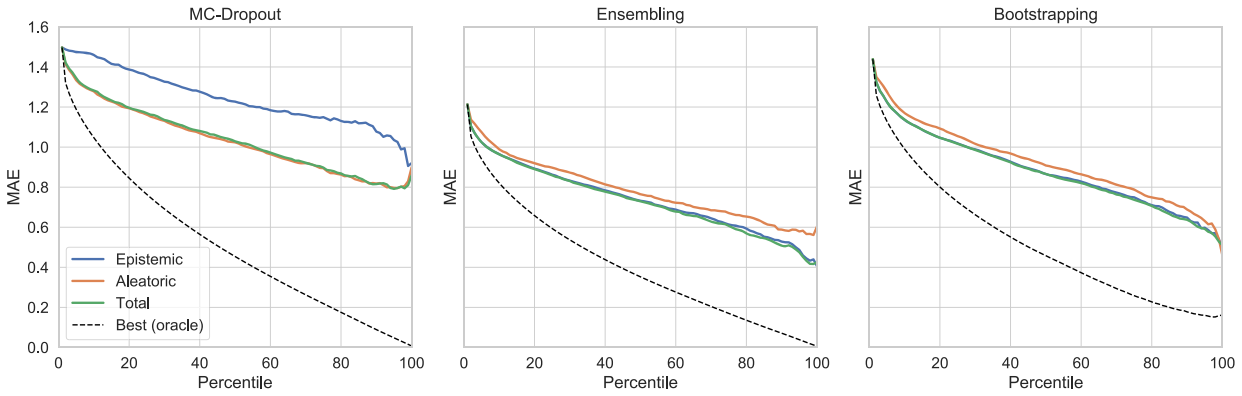


Figure 17. QM9, out-of-domain test set. Confidence curves for the different methods.

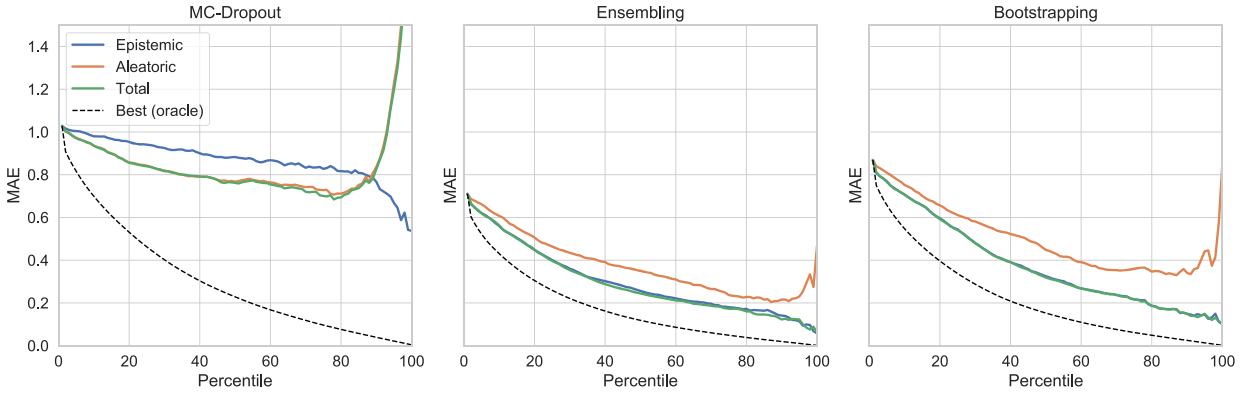


Figure 18. Alchemy, out-of-domain test set. Confidence curves for the different methods.

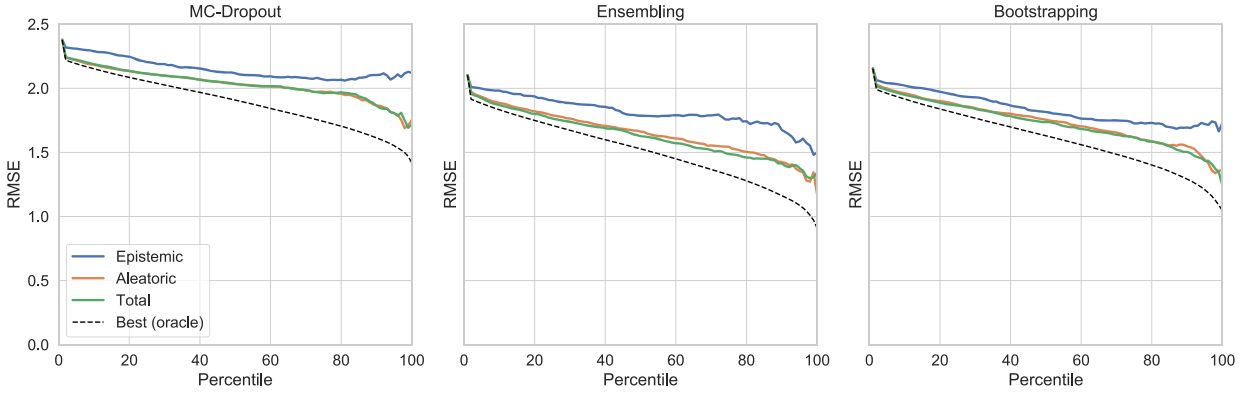


Figure 19. PDBbind, out-of-domain test set. Confidence curves for the different methods.

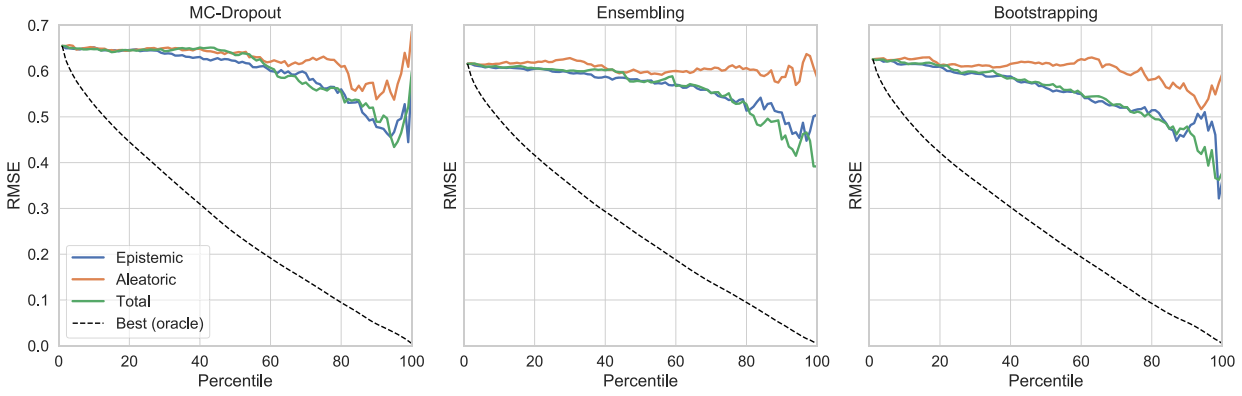


Figure 20. Lipophilicity, out-of-domain test set. Confidence curves for the different methods.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

This behavior can be explained by considering the effects of substituting each training data set with a bootstrap sample. Each network only sees a fraction of the starting training data set, thus increasing individual and ensembled error. Because aleatoric uncertainty is estimated from data, it follows a trend similar to error and it degrades. However, bootstrapping promotes diversity in ensembled models, which is key for epistemic uncertainty estimation, thus improving its calibration. We can argue that as training size increases—as long as the target molecular space is kept unchanged—bootstrapping becomes more advantageous, because each bootstrap sample becomes a better approximator of the underlying distribution, thus avoiding losses in error and aleatoric calibration in each single instance and in the ensembled model but keeping an advantage for epistemic calibration. Moreover, as we have observed, bootstrapping performance degrades less than ensembling in the out-of-domain setting. This can be explained by a gain of generalization power given by the additional diversity of bootstrapping.

In previous studies for DNN-based image regression/classification, bootstrapping did not result in significant improvements over ensembling.[12,50] We can speculate that these differences are due to the peculiarities of chemical space, characterized, for example, by a larger intrinsic variability that can be exploited by bootstrapping. Results obtained for bootstrapping justify its recent use in active learning methodologies for molecular property prediction,[37] where model uncertainty (epistemic uncertainty) and generalization power are required.

Discussing the theoretical reasons leading to explicit ensembles (Deep Ensembles) outperforming weight-sharing ensemble (MC-dropout) is beyond the scope of this paper and is a subject of active research. Recently, it has been shown that Deep Ensembles allow exploring different modes in function space, thereby sampling diverse functions, while MC-dropout tends to focus on a single mode in function space with low diversity in the predictions.[50]

In our analysis, MC-dropout for molecular property prediction has shown two main limitations. First, it results in a higher error than ensembling or bootstrapping, with higher confidence curves both in relative and absolute terms. Second, MC-dropout tends to give overly optimistic epistemic uncertainty estimates which are also less correlated to the true error than estimates produced by the other tested methods. This is especially detrimental for situations where the epistemic uncertainty is large.

Nonetheless, MC-dropout has the practical advantage of weight sharing that can result in lower training time/memory consumption. Other techniques have been proposed to train an ensemble of models in a fraction of the time required for a "true" ensemble, such as *snapshot ensembling*,[51] and should be the subject of future work.

Aleatoric and Epistemic Uncertainty. Even though the methods investigated in this work jointly model aleatoric and epistemic uncertainties, their separate evaluation has allowed direct comparisons. As we have observed, their performance and their relative contribution to total uncertainty turn out to be data set-dependent.

As aleatoric and epistemic uncertainties are conceptually orthogonal, one might expect them to not be strongly correlated. However, here we show that they both result in high rank-order correlation with respect to true error (as showed by decreasing confidence curves), so at least from a rank-order point of view, the two contributions are often correlated. An analytical comparison (see Supporting Information, Table S14 for all the results) shows that—with few exceptions—when Spearman rank-order correlation is high, Pearson correlation is lower (e.g., on Alchemy, Spearman rank-order correlation = 0.47/0.87, Pearson correlation = 0.01/0.30). This shows that, in most cases, even if the two uncertainty types rank molecules are in the same order, their values are not directly proportional.

Because aleatoric uncertainty captures inherent data noise, one may wonder whether it correlates with error in the data with respect to a more accurate ground truth. For example, considering density-functional theory (DFT) calculations (QM9 data set), the ground truth can be better approximated by high level quantum chemistry calculations such as coupled cluster theory with a generous basis set[52,53] or experimental results. Our analysis (see Supporting Information, section Additional analyses for all the details) shows that aleatoric uncertainty calculated from our model trained on DFT calculations does *not* correlate with the DFT error measured against a higher level of theory and experiments (Spearman correlation $\approx$ 0.13). This can be explained as follows. DFT errors are relatively *internally consistent*; therefore, even though large, they will not be characterized by high variability (i.e., aleatoric uncertainty). As computed, aleatoric uncertainty can help detect data errors when these come from inconsistencies, not systematic errors.

In most cases, we observe that total uncertainty outperforms or matches the best performing individual uncertainty component, and this is the case even when one of the two components leads to remarkably better metrics than the other (see, e.g., the comparison between aleatoric and epistemic uncertainty for PDBbind, both for ranking-based and calibration-based plots). We can explain this behavior as follows.

All our uncertainty evaluation metrics (with the exception of dispersion) take into consideration how, directly or indirectly, uncertainty estimates relate to true errors. Aleatoric uncertainty should relate to the inherent noise in the observed property, while epistemic uncertainty should relate to the error in the trained function. However, the only observable error (the true error) includes *both* these contributions. Therefore, it is the total uncertainty that should best model the observed error and, therefore, result in the best performance. Given our experiments, the usage of the total uncertainty can be generally suggested in applications.

From this, we can also speculate that evaluating the individual uncertainty contributions can allow pinpointing their *relative importance* in the context of different data sets, that is, understanding if the observed error is primarily due to the neural network approximating function or inherent data variability. This is discussed in the next section.

Data Sets and Uncertainty Types. In general, the relative contribution of the two uncertainty components with respect to total uncertainty and their comparative performance turn out to be strongly data set-dependent.

On QM9 and Alchemy, epistemic uncertainty appears to be predominant. This can be clearly observed for ensembling and bootstrapping, both for ranking-based and, especially, calibration-based evaluation (Figures 9 and 10). Indeed, the reason why MC-dropout falls behind the two other approaches seems to be its worse epistemic uncertainty estimation. As a consequence, the total error seems to be primarily due to the neural network rather than the inherent data noise. This can be traced back to the fact that these data sets are derived from

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

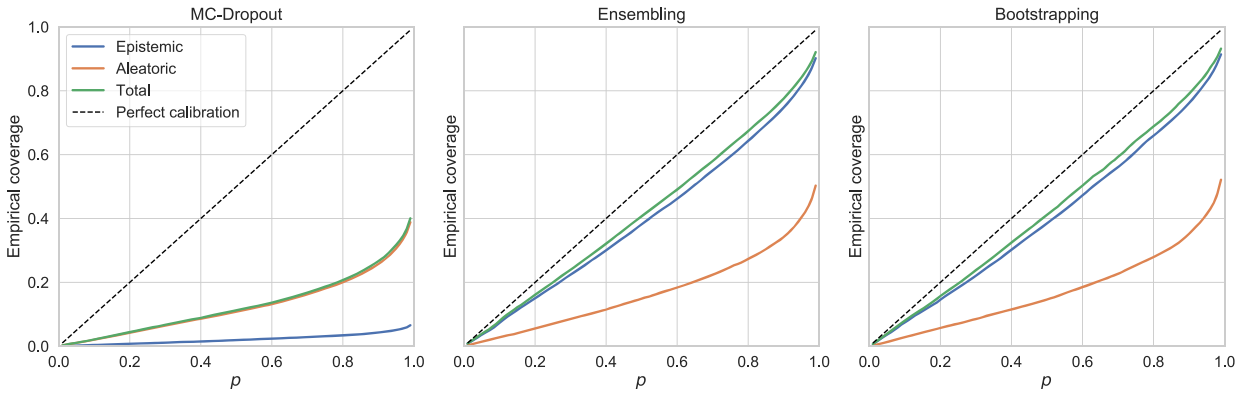


Figure 21. QM9, out-of-domain test set. Confidence-based calibration for the different methods.

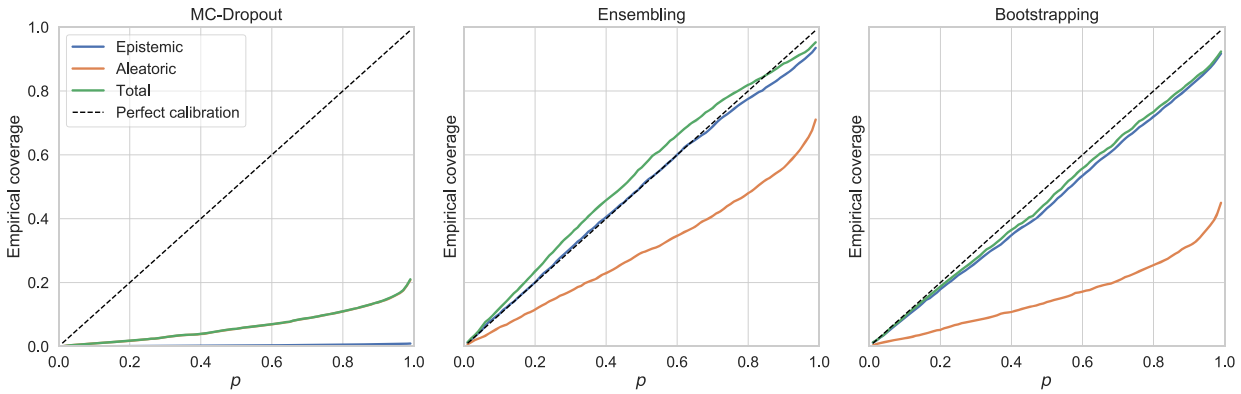


Figure 22. Alchemy, out-of-domain test set. Confidence-based calibration for the different methods.

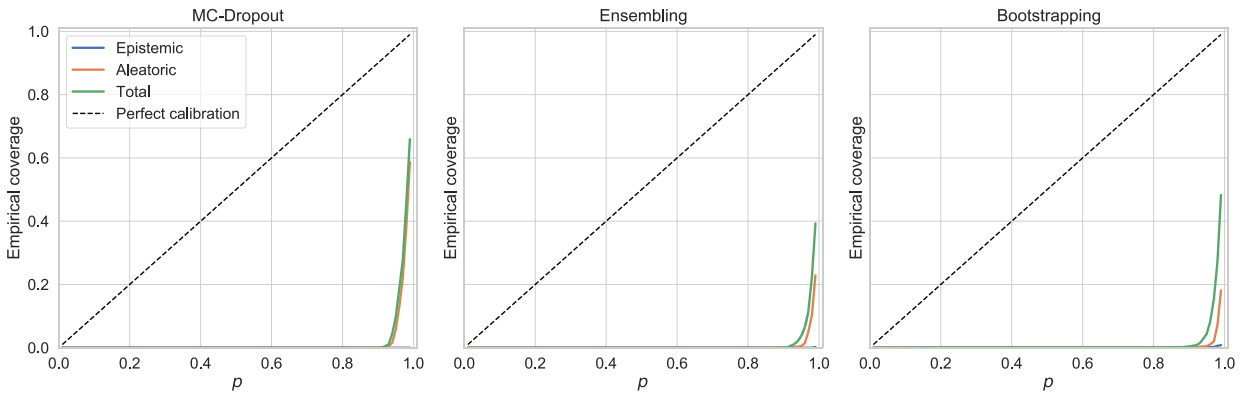


Figure 23. PDBbind, out-of-domain test set. Confidence-based calibration for the different methods.

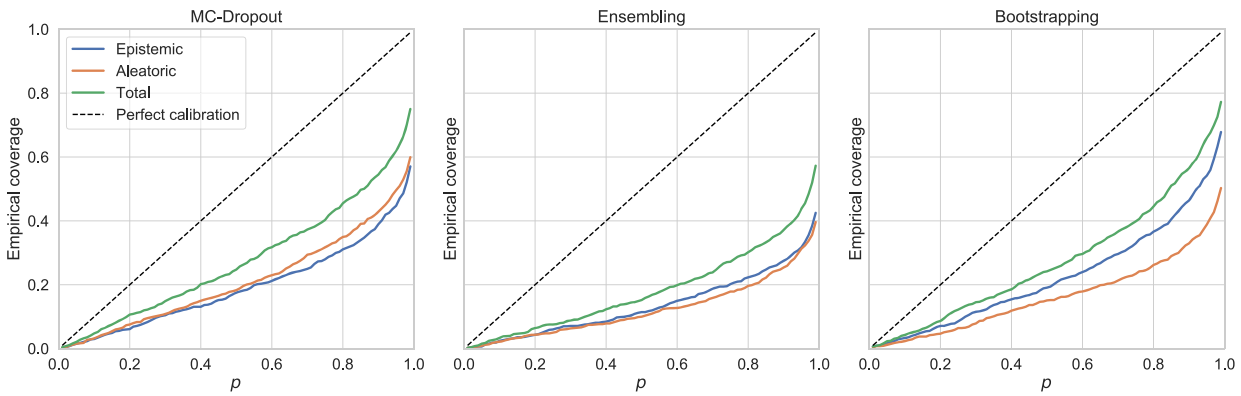


Figure 24. Lipophilicity, out-of-domain test set. Confidence-based calibration for the different methods.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

electronic structure theory, which is relatively internally consistent, and therefore results in low inherent variability (aleatoric uncertainty).

On PDBbind, aleatoric uncertainty appears to be predominant instead (Figure 11). Not only does it lead to better confidence curves but it also appears to correctly predict most of the error (Figure 15). We can explain this behavior by considering the fact that PDBbind includes experimentally measured binding affinities collected over a relatively long period of time; therefore, a higher level of variability in the data is expected, and total error seems to be primarily due to this inherent noise.

Likewise, Lipophilicity is characterized by a relatively high aleatoric uncertainty because of its experimental origin (Figure 12). However, unlike PDBbind, epistemic uncertainty provides a relatively important contribution to total uncertainty in matching the true error and empirical coverages. We can explain this behavior by considering their different sizes (Table 1): more training data samples reduce predicted epistemic uncertainty. In this case, total error seems to be due to both inherent noise and the neural network.

The above analysis provides an interesting link between data sets and uncertainty types. However, it also has some limitations. First, we observe that the relative contribution of the two uncertainty components could depend on the estimation method. For example, considering Lipophilicity, all methods clearly show that both components are key contributors to total uncertainty; however, we cannot exactly conclude whether aleatoric uncertainty is more important (as resulting from MC-dropout and ensembling) or vice versa (as resulting from bootstrapping). Moreover, as discussed in the next section, both uncertainty components are significantly underestimated by all methods if one uses an out-of-domain test set and that could further hinder this analysis. Nonetheless, these results take a step in the direction of relating uncertainty estimates to data set features.

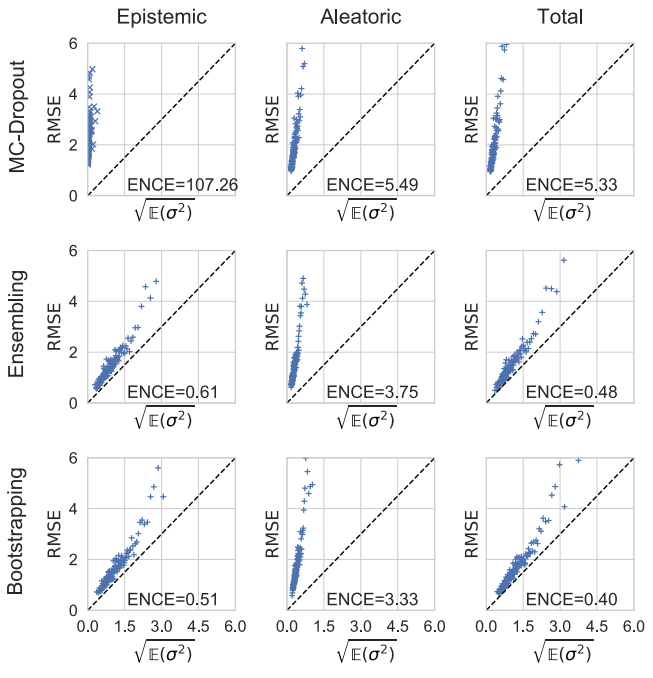


Figure 25. QM9, out-of-domain test set. Error-based calibration.

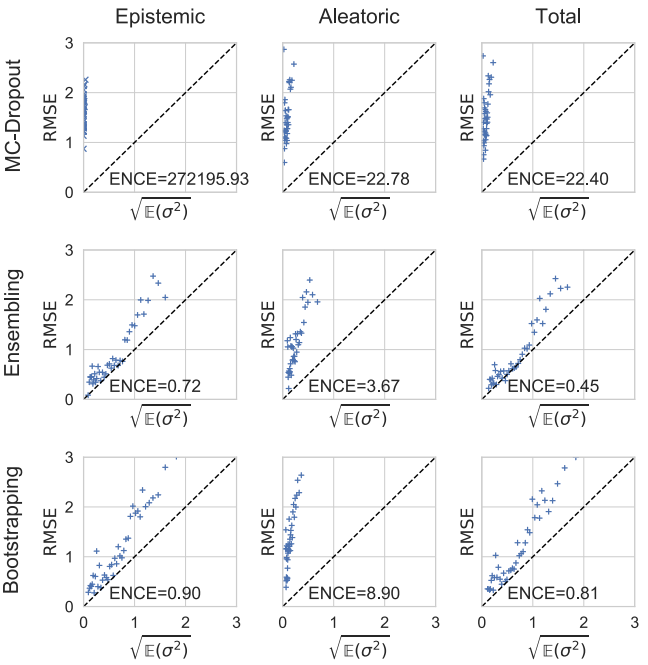


Figure 26. Alchemy, out-of-domain test set. Error-based calibration.

Domain Shift Analysis and Domain Adaptation. Domain shift analysis is characterized by mixed results. On the one hand, ranking-based performance does not appear to be drastically affected by out-of-domain molecules. Ranking-based analysis suggests that even though predicted on molecules quite different with respect to those seen during training, uncertainty is effective for reliably detecting smaller subsets of low-error molecules. Out-of-domain error on high percentiles is often lower with respect to overall in-domain error (this is the case for QM9, Alchemy, and Lipophilicity). On the other hand, calibration performance is strongly affected by out-of-domain molecules and is *underestimated* in most of the cases. The latter result is in line with what has been recently observed in Li et al.[37]

In all of our out-of-domain experiments with the exception of size split, the error increases without the uncertainty being able to completely capture this rise—thus leading to overly optimistic uncertainty estimates. The extent of this underestimation trend seems to be split dependent, being the highest for PDBbind (time split), followed by QM9 (scaffold split) and Lipophilicity (chemical element split) with comparable results. Alchemy (size split) is the only case where the out-of-domain uncertainty estimates are well calibrated.

We observe that this latter result could depend on at least two different reasons. First, as defined by Chen et al.[48] (and, consequently, in our experiments[d]), this split does not create two totally disjointed training and test distributions because a low fraction of molecules of bigger size (≈5%) is present in the training set. Second, GCNNs are designed to be robust with respect to the number of atoms, and the size split does not prevent the same scaffolds and chemical elements to be present in both the training and the test data sets.

For a well-calibrated behavior, we would expect uncertainty (in particular *epistemic uncertainty*) to increase and match the expected increase in error. However, evaluating the epistemic uncertainty median for all the considered data sets and estimation methods (see Table S15 in the Supporting Information), we observe that—with the exception of Alchemy (size split)—out-of-domain epistemic uncertainty not only fails

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

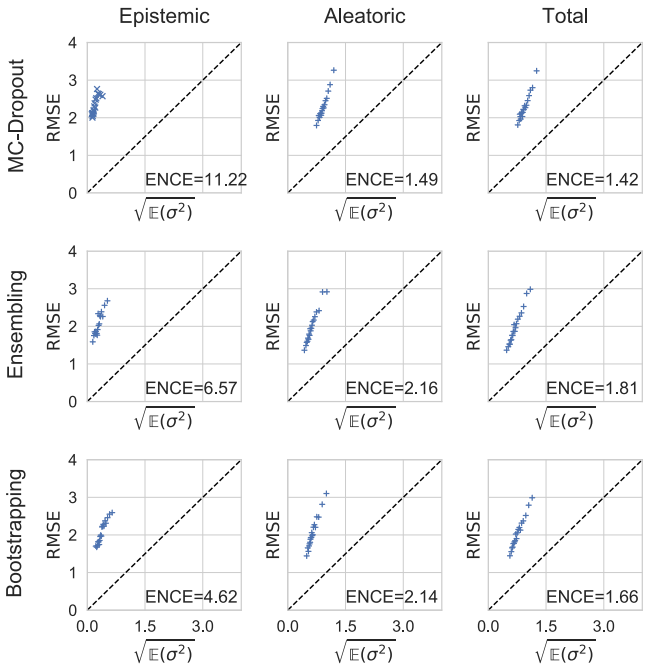


Figure 27. PDBbind, out-of-domain test set. Error-based calibration.

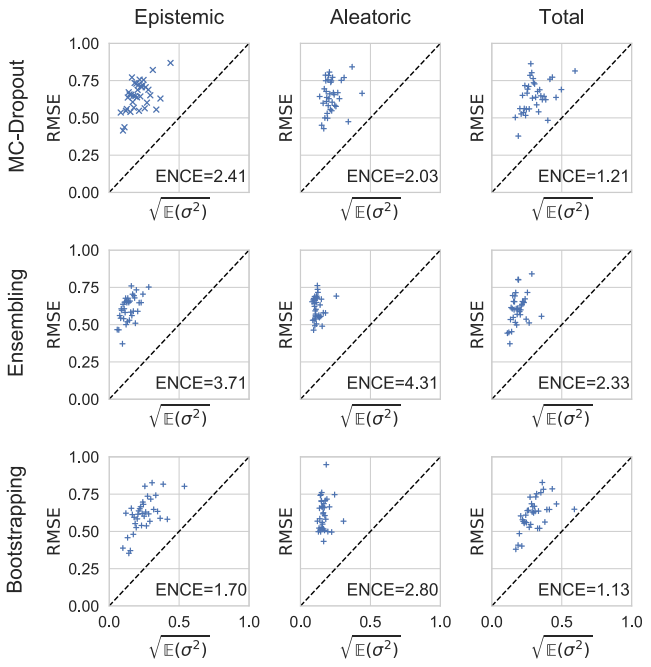


Figure 28. Lipophilicity, out-of-domain test set. Error-based calibration.

to match the increase in error but in several cases (PDBbind and Lipophilicity) does not increase *at all* with respect to in-domain estimates.

However, even though out-of-domain uncertainty estimates are often miscalibrated, they are usually highly correlated with the true error. For example, error-based calibration has correlation ≥0.85 in all cases for PDBbind (Figure 27). This suggests that simple recalibration techniques[21,22] should be effective.

When discussed in the context of applicability domain,[23] the tested uncertainty estimation methods are characterized by mixed results. On the one hand, mean out-of-domain error for uncertainty-aware DNNs is generally lower than base model error, especially for ensembling and bootstrapping (Table 2). This can be explained by a gain of generalization ability given by Bayesian inference, which is especially effective in the out-of-domain setting, suggesting that the domain of applicability of Bayesian DNNs is broader than point estimate (i.e., "standard") DNNs. Moreover, our analysis shows that by considering subsets of out-of-domain molecules with low predicted uncertainty, we can reliably obtain subsets of out-of-domain predictions with comparable or lower error with respect to in-domain molecules (as in QM9, Alchemy, and Lipophilicity data sets), or, at least, significantly lower than mean test error (as in PDBbind data set). On the other hand, predicted uncertainty does not appear to be effective for a prior assessment of the applicability of a model on a given out-of-domain data set, nor to reliably predict to what extent the error will increase in absolute terms with respect to in-domain molecules.

Comparison of Evaluation Criteria. Until now, we mainly compared uncertainty models. However, the obtained results also allow for the comparison of the usefulness of different *evaluation methods*.

Taking into consideration, calibration allows identifying several important patterns that do not emerge from confidence curves only, such as their relative contribution in matching true error and empirical coverage for the different methods and data sets. In contrast, even recent work that seeks to obtain "uncertainty-calibrated prediction of molecular properties"[7] do not explicitly take into consideration calibration evaluation in the results.

The comparison between the two considered definitions of calibration is more subtle. Qualitatively, most of the conclusions derived by confidence-based calibration, such as when and to what extent one uncertainty component is more calibrated than the other, are also reflected in error-based calibration. However, we also notice how, in some cases, one of the two definitions allows capturing more information about the true uncertainty behavior.

On the one hand, even if error-based calibration directly relates error and uncertainty according to its definition, the inherent non-uniformity of uncertainty estimates makes it difficult to obtain reliable statistics in some uncertainty ranges (high uncertainty ranges in our experiments) and when the data set is smaller (Lipophilicity in our experiments), with less stable results. This also prevents assessing if the error in these ranges is due to uncertainty estimates themselves or to insufficient data for computing reliable statistics. On the other hand, when uncertainty turns out to be largely underestimated (e.g., out-of-domain PDBbind), calibration-based confidence turns out to be basically useless (mostly zero), while error-based calibration still provides meaningful insights, such as the distance with respect to perfect calibration. Moreover, error-based calibration allows bringing out the uncertainty/error correlation, if any, and avoids issues when recalibration techniques are employed.[22] Therefore, we can conclude that the choice between these two different definitions is context-dependent. If the data set is large enough to enable meaningful estimates for all the bins, error-based calibration should be preferred because it allows for a more direct comparison and it avoids issues with largely underestimated uncertainty and recalibration techniques. Instead, if the uncertainty distribution is highly skewed and only a few samples are available in some ranges, confidence-based calibration can overcome this and results in less noisy plots.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

In general, we observe that the quantitative measures introduced to summarize uncertainty performance correctly capture the different behaviors and have allowed a more fine-grained comparison. However, in specific cases, some evaluation metrics turn out to be only conditionally useful. For example, evaluating the error drop leads to meaningful comparisons when the decreasing ratio is high (e.g., QM9), but it loses effectiveness for noisy confidence plots (e.g., Lipophilicity).

CONCLUSIONS AND FUTURE WORK

In this paper, we compared three state-of-the-art approaches for uncertainty estimation in neural networks in the context of GCNNs for molecular property prediction: MC-dropout with concrete dropout, Deep Ensembles, and bootstrapping. We selected those approximate Bayesian inference techniques satisfying some specific application-oriented criteria: scalability, lack of hyperparameters, and independence from the underlying network architecture. These techniques have been first reviewed in a unified framework that separates aleatoric and epistemic uncertainty, also in the light of recent interpretations given to ensembling, and then experimentally compared on four public data sets based on a set of introduced criteria. Those criteria have been selected to evaluate uncertainty from different perspectives: based on its ability to define a ranking of most confident predictions, based on uncertainty calibration (two different recent definitions for regression have been employed), based on dispersion that measures estimated heterogeneity, and based on robustness to domain shifts in the test set with respect to the training set, with different split criteria being employed.

The obtained results lead to multiple interesting conclusions. Ensembling and bootstrapping appear to consistently outperform MC-dropout, confirming the results recently presented for other domains and different network types also for GCNN-based molecular property prediction. The comparison between ensembling and bootstrapping leads to more mixed results. Even though ensembling is better with respect to most of the considered metrics, including overall MAE, bootstrapping appears to outperform ensembling for others, notably epistemic uncertainty calibration. This is not in line with what has been previously described in the context of image regression/classification, highlighting an interesting property of the chemical space for the model and the data sets considered. Furthermore, the results presented have led to a better understanding about the role of aleatoric/epistemic uncertainty for DNN-based molecular property prediction. We investigated the relationship between these two uncertainty types, showing how, even though often correlated from a rank-order point of view, they are not directly proportional. Moreover, we discussed why aleatoric uncertainty generally does not capture error in the data with respect to a more precise ground truth, analyzing the correlation between aleatoric uncertainty estimated from QM9 and DFT errors with respect to higher level of theory and experiments. In addition, we showed how evaluating the individual uncertainty contributions can allow pinpointing their relative importance in the context of different data sets, with remarkable differences between experimental data sets and those derived from electronic structure theory. Finally, our experiments have led to a better understanding of the performance of DNN-based uncertainty estimates under test set domain shifts. Comparing uncertainty estimates for in-domain and out-of-domain test sets, we showed that uncertainty generalization ability is metric- and data set-dependent, and we discussed these results in the context of determining the applicable domain of the model.

One of the main limitations we found for all the tested methods is out-of-domain uncertainty underestimation. This hinders their usage for domain applicability analysis and overcoming this weakness should be a major goal of future work. To allow correct out-of-domain calibration, future work could target recalibration techniques using existing methods[21,22] or new methods. For the latter, a promising direction is represented by the increase of diversity in the ensembled models. This might not be the result of diversity in the data, as in bootstrapping, but instead come from the model itself.[54,55] Balancing diversity, training data size, and the number of hyperparameters appears to be a challenging tradeoff.

Even though our results about which uncertainty type is prevalent for each data set are congruent with the data set origin (e.g., experimental vs computed), future work should further investigate these trends for additional data sets and models.

In addition, our analysis mainly focused on how uncertainty relates (directly or indirectly) to model error. Providing chemical explanations about *why* one sort of uncertainty is significantly larger than the other for a certain molecule should be the subject of future work.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.9b00975.

> Summary tables with quantitative metrics both for in-domain and out-of-domain test sets, confidence-oracle error plots for all the experiments, details about the implementation and the experimental settings, details about data preparation, discussion on the role of hyperparameters on uncertainty estimates, discussion on the Bayesian interpretation of ensembling, analytical results on the correlation between aleatoric and epistemic uncertainty, analytical results on the correlation between aleatoric uncertainty and a more precise ground truth, and analytical results on epistemic uncertainty median for in-domain and out-of-domain test sets (PDF)

AUTHOR INFORMATION

Corresponding Authors

Yi-Pei Li − *Department of Chemical Engineering, National Taiwan University, Taipei 10617, Taiwan;* orcid.org/0000-0002-1314-3276; Email: yipeili@ntu.edu.tw

William H. Green − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0003-2603-9694; Email: whgreen@mit.edu

Authors

Gabriele Scalia − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States; Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milano, Italy;* orcid.org/0000-0003-3305-9220

Colin A. Grambow − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-2204-9046

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Barbara Pernici − *Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milano, Italy;* orcid.org/0000-0002-2034-9774

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.9b00975

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the MIT Consortium for Machine Learning for Pharmaceutical Discovery and Synthesis. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract no. DE-AC02-05CH11231. G.S. acknowledges the support of the "Progetto Roberto Rocca" Doctoral Fellowship. Y.P.L. is supported by Taiwan MOST Young Scholar Fellowship Einstein Program (108-2636-E-002-017).

■ ADDITIONAL NOTES

[a]The confidence-oracle error has been called sparsification error in computer vision;[19] in that context AUCO has been called area under the sparsification error curve.[19]

[b]We did not use ECE in our tests because it does not add significant information to AUCE for confidence-based calibration.

[c]In contrast to previous comparisons, that used the "base" version of MC-dropout[12,17,19,49] we employed concrete MC-dropout that was independently proven superior to standard MC-dropout[20,40] but to our knowledge has not been directly compared to ensembling and bootstrapping before.

[d]See data preparation in the Supporting Information for more details.

■ REFERENCES

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

(2) 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.

(3) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chem. Sci.* **2018**, *9*, 5441−5451.

(4) Duvenaud, D.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015; Vol. *2*, pp 2224−2232.

(5) Gal, Y. Uncertainty in deep learning. Ph.D. Thesis, University of Cambridge, 2016.

(6) Kendall, A.; Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017; pp 5580−5590.

(7) Zhang, Y.; Lee, A. A. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chem. Sci.* **2019**, *10*, 8154−8163.

(8) Proppe, J.; Reiher, M. Reliable Estimation of Prediction Uncertainty for Physicochemical Property Models. *J. Chem. Theory Comput.* **2017**, *13*, 3297−3317.

(9) Neal, R. M. *Bayesian Learning for Neural Networks*; Springer-Verlag, 1996.

(10) Gal, Y.; Ghahramani, Z. Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016; Vol. *48*, pp 1050−1059.

(11) Ryu, S.; Kwon, Y.; Kim, W. Y. A Bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chem. Sci.* **2019**, *10*, 8438−8446.

(12) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017; pp 6405−6416.

(13) De Fauw, J. D.; Ledsam, J. R.; Romera-Paredes, B.; Nikolov, S.; Tomasev, N.; Blackwell, S.; Askham, H.; Glorot, X.; O'Donoghue, B.; Visentin, D.; van den Driessche, G.; Lakshminarayanan, B.; Meyer, C.; Mackinder, F.; Bouton, S.; Ayoub, K.; Chopra, R.; King, D.; Karthikesalingam, A.; Hughes, C. O.; Raine, R.; Hughes, J.; Sim, D. A.; Egan, C.; Tufail, A.; Montgomery, H.; Hassabis, D.; Rees, G.; Back, T.; Khaw, P. T.; Suleyman, M.; Cornebise, J.; Keane, P. A.; Ronneberger, O. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nat. Med.* **2018**, *24*, 1342−1350.

(14) Tomašev, N.; Glorot, X.; Rae, J.; Zielinski, M.; Askham, H.; Saraiva, A.; Mottram, A.; Meyer, C.; Ravuri, S.; Protsyuk, I.; Connell, A.; Hughes, C.; Karthikesalingam, A.; Cornebise, J.; Montgomery, H.; Rees, G.; Laing, C.; Baker, C.; Peterson, K.; Reeves, R.; Hassabis, D.; King, D.; Suleyman, M.; Back, T.; Nielson, C.; Ledsam, J.; Mohamed, S. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature* **2019**, *572*, 116−119.

(15) Duvenaud, D.; Maclaurin, D.; Adams, R. P. Early Stopping as Nonparametric Variational Inference. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, 2016; pp 1070−1077.

(16) Pearce, T.; Zaki, M.; Brintrup, A.; Neel, A. Uncertainty in neural networks: Bayesian ensembling. **2018**, arXiv:1810.05546. arXiv preprint.

(17) Gustafsson, F. K.; Danelljan, M.; Schön, T. B. Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision. **2019**, arXiv:1906.01620. arXiv preprint.

(18) Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K. Q. On Calibration of Modern Neural Networks. *Proceedings of the 34th International Conference on Machine Learning*, 2017; Vol. *70*, pp 1321−1330.

(19) Ilg, E.; Cicek, O.; Galesso, S.; Klein, A.; Makansi, O.; Hutter, F.; Brox, T. Uncertainty estimates and multi-hypotheses networks for optical flow. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018; pp 652−667.

(20) Mukhoti, J.; Gal, Y. Evaluating Bayesian Deep Learning Methods for Semantic Segmentation. **2018**, arXiv:1811.12709. arXiv preprint.

(21) Kuleshov, V.; Fenner, N.; Ermon, S. Accurate Uncertainties for Deep Learning Using Calibrated Regression. *Proceedings of the 35th International Conference on Machine Learning*, 2018; pp 2796−2804.

(22) Levi, D.; Gispan, L.; Giladi, N.; Fetaya, E. Evaluating and Calibrating Uncertainty Prediction in Regression Tasks. **2019**, arXiv:1905.11659. arXiv preprint.

(23) Liu, R.; Glover, K. P.; Feasel, M. G.; Wallqvist, A. General Approach to Estimate Error Bars for Quantitative Structure-Activity Relationship Predictions of Molecular Activity. *J. Chem. Inf. Model.* **2018**, *58*, 1561−1575.

(24) Cortés-Ciriano, I.; Bender, A. Deep confidence: a computationally efficient framework for calculating reliable prediction errors for deep neural networks. *J. Chem. Inf. Model.* **2018**, *59*, 1269−1281.

(25) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional embedding of attributed molecular graphs for physical property prediction. *J. Chem. Inf. Model.* **2017**, *57*, 1757−1772.

(26) Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018; pp 7482−7491.

(27) Le, Q. V.; Smola, A. J.; Canu, S. Heteroscedastic Gaussian process regression. *Proceedings of the 22nd International Conference on Machine Learning*, 2005; pp 489−496.
(28) 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)*, 1994; pp 55−60.
(29) Bishop, C. M. *Mixture Density Networks*, 1994.
(30) Choi, S.; Lee, K.; Lim, S.; Oh, S. Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018; pp 6915−6922.
(31) Kristiadi, A.; Fischer, A. Predictive Uncertainty Quantification with Compound Density Networks. **2019**, arXiv:1902.01080. arXiv preprint.
(32) Ma, Y.-A.; Chen, T.; Fox, E. B. A Complete Recipe for Stochastic Gradient MCMC. *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015; Vol. *2*, pp 2917−2925.
(33) Zhang, R.; Li, C.; Zhang, J.; Chen, C.; Wilson, A. G. Cyclical stochastic gradient MCMC for Bayesian deep learning. **2019**, arXiv:1902.03932, arXiv preprint.
(34) Graves, A. Practical Variational Inference for Neural Networks. *Proceedings of the 24th International Conference on Neural Information Processing Systems*, 2011; pp 2348−2356.
(35) Hernández-Lobato, J. M.; Adams, R. P. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. *Proceedings of the 32Nd International Conference on International Conference on Machine Learning*, 2015; Vol. *37*, pp 1861−1869.
(36) Liu, Q.; Wang, D. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016; pp 2378−2386.
(37) Li, Y.-P.; Han, K.; Grambow, C. A.; Green, W. H. Self-Evolving Machine: A Continuously Improving Model for Molecular Thermochemistry. *J. Phys. Chem. A* **2019**, *123*, 2142−2152.
(38) Smith, J. S.; Nebgen, B.; Lubbers, N.; Isayev, O.; Roitberg, A. E. Less is more: Sampling chemical space with active learning. *J. Chem. Phys.* **2018**, *148*, 241733.
(39) Peterson, A. A.; Christensen, R.; Khorshidi, A. Addressing uncertainty in atomistic machine learning. *Phys. Chem. Chem. Phys.* **2017**, *19*, 10978−10985.
(40) Gal, Y.; Hron, J.; Kendall, A. Concrete dropout. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017; pp 3584−3593.
(41) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929−1958.
(42) Dietterich, T. G. Ensemble Methods in Machine Learning. *Proceedings of the First International Workshop on Multiple Classifier Systems*, 2000; pp 1−15.
(43) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.
(44) Niculescu-Mizil, A.; Caruana, R. Predicting good probabilities with supervised learning. *Proceedings of the 22nd international conference on Machine learning*, 2005; pp 625−632.
(45) DeGroot, M. H.; Fienberg, S. E. The comparison and evaluation of forecasters. *J. Royal Stat. Soc. D* **1983**, *32*, 12−22.
(46) Gneiting, T.; Balabdaoui, F.; Raftery, A. E. Probabilistic forecasts, calibration and sharpness. *J. Royal Stat. Soc. B* **2007**, *69*, 243−268.
(47) Beluch, W. H.; Genewein, T.; Nurnberger, A.; Kohler, J. M. The Power of Ensembles for Active Learning in Image Classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018; pp 9368−9377.
(48) Chen, G.; Chen, P.; Hsieh, C.-Y.; Lee, C.-K.; Liao, B.; Liao, R.; Liu, W.; Qiu, J.; Sun, Q.; Tang, J.; Zemel, R.; Zhang, S. Alchemy: A Quantum Chemistry Dataset for Benchmarking AI Models. **2019**, arXiv:1906.09427, arXiv preprint.
(49) Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J. V.; Lakshminarayanan, B.; Snoek, J. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. **2019**, arXiv:1906.02530, arXiv preprint.
(50) Fort, S.; Hu, H.; Lakshminarayanan, B. Deep Ensembles: A Loss Landscape Perspective. **2019**, arXiv:1912.02757, arXiv preprint.
(51) Huang, G.; Li, Y.; Pleiss, G.; Liu, Z.; Hopcroft, J. E.; Weinberger, K. Q. Snapshot Ensembles: Train 1, Get M for Free. *International Conference on Learning Representations, ICLR*, 2017.
(52) Smith, J. S.; Nebgen, B. T.; Zubatyuk, R.; Lubbers, N.; Devereux, C.; Barros, K.; Tretiak, S.; Isayev, O.; Roitberg, A. E. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.* **2019**, *10*, 2903.
(53) Grambow, C. A.; Li, Y.-P.; Green, W. H. Accurate Thermochemistry with Small Data Sets: A Bond Additivity Correction and Transfer Learning Approach. *J. Phys. Chem. A* **2019**, *123*, 5826−5835.
(54) Lee, S.; Purushwalkam, S.; Cogswell, M.; Crandall, D.; Batra, D. Why M heads are better than one: Training a diverse ensemble of deep networks. **2015**, arXiv:1511.06314, arXiv preprint.
(55) Pearce, T.; Anastassacos, N.; Zaki, M.; Neely, A. Bayesian Inference with Anchored Ensembles of Neural Networks, and Application to Reinforcement Learning. **2018**, arXiv:1805.11324, arXiv preprint.

https://dx.doi.org/10.1021/acs.jcim.9b00975
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX