Supporting Information: 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[*,†]

[†] *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA*

[‡] *Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milano, Italy*

[¶] *Department of Chemical Engineering, National Taiwan University, Taipei 10617, Taiwan*

E-mail: yipeili@ntu.edu.tw; whgreen@mit.edu

Experiment summary tables – in-domain

QM9

Table S1: Summary of metrics for the different methods (QM9, in-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	46.72	31.55	31.72	18.79	20.83	17.03	19.18	25.08	19.05
Error drop	1.67	2.55	2.62	6.72	4.93	7.40	6.85	5.23	6.85
Decr. Ratio	0.95	0.98	0.96	1.0	0.99	1.0	1.0	1.0	1.0
AUCE	44.79	29.44	28.74	2.62	19.90	3.62	1.36	31.31	1.69
MCE	0.85	0.50	0.48	0.087	0.33	0.061	0.051	0.53	0.044
ENCE	30.97	3.27	3.11	0.51	2.28	0.27	0.24	4.37	0.20
c_v	0.97	0.50	0.49	0.74	0.51	0.67	0.74	0.45	0.71

Alchemy

Table S2: Summary of metrics for the different methods (Alchemy, in-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	13.15	6.44	6.51	4.53	4.57	4.22	4.54	5.03	4.49
Error drop	3.07	6.48	6.31	7.99	8.71	8.55	7.97	7.20	8.48
Decr. Ratio	0.79	0.89	0.87	0.92	0.91	0.9	0.89	0.93	0.96
AUCE	48.18	36.15	35.91	9.88	17.50	2.46	8.99	32.32	6.99
MCE	0.95	0.63	0.63	0.16	0.28	0.075	0.14	0.55	0.12
ENCE	415.79	8.90	8.85	1.61	2.58	0.96	1.35	6.53	1.20
c_v	1.91	1.31	1.33	1.71	1.60	1.75	1.80	1.19	1.75

PDBbind

Table S3: Summary of metrics for the different methods (PDBbind, in-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	76.37	67.59	65.92	74.39	64.81	57.47	67.42	65.99	60.87
Error drop	0.92	1.67	1.67	1.82	2.22	2.25	1.47	2.60	2.09
Decr. Ratio	0.29	0.38	0.34	0.21	0.34	0.53	0.47	0.40	0.52
AUCE	34.01	6.53	4.83	30.52	6.10	3.53	26.46	9.20	4.88
MCE	0.57	0.11	0.09	0.52	0.11	0.08	0.44	0.16	0.10
ENCE	3.26	0.20	0.14	2.52	0.26	0.15	1.71	0.37	0.16
c_v	0.56	0.23	0.23	0.54	0.33	0.28	0.48	0.28	0.26

Lipophilicity

Table S4: Summary of metrics for the different methods (Lipophilicity, in-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	30.62	34.56	32.24	25.52	32.50	28.00	24.72	32.05	25.92
Error drop	1.57	0.68	1.14	3.44	0.90	4.41	5.72	1.52	5.89
Decr. Ratio	0.21	0.04	0.11	0.29	0.10	0.26	0.29	0.07	0.29
AUCE	22.33	16.14	9.33	24.17	17.10	10.47	16.18	27.35	11.00
MCE	0.37	0.27	0.16	0.40	0.29	0.17	0.27	0.45	0.19
ENCE	1.63	0.99	0.55	2.04	1.35	0.78	1.07	2.34	0.70
c_v	0.49	0.29	0.36	0.49	0.28	0.27	0.52	0.31	0.41

Experiment summary tables – out-of-domain

QM9 – scaffold split

Table S5: Summary of metrics for the different methods (QM9, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	73.35	52.29	52.93	34.12	38.68	33.38	36.27	40.05	35.81
Error drop	1.64	1.67	1.75	3.02	2.02	2.88	2.91	3.18	2.86
Decr. Ratio	0.86	0.90	0.91	0.99	0.95	0.98	0.98	0.99	1.0
AUCE	47.36	37.13	36.70	10.18	32.81	8.10	9.62	32.57	7.50
MCE	0.92	0.65	0.64	0.16	0.56	0.13	0.14	0.55	0.11
ENCE	107.26	5.49	5.38	0.61	3.75	0.48	0.51	3.33	0.40
c_v	1.51	0.46	0.47	0.63	0.45	0.59	0.61	0.42	0.58

Table S6: Summary of relative metrics for the different methods (QM9, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	+57%	+66%	+67%	+82%	+86%	+96%	+89%	+60%	+88%
Error drop	-2%	-35%	-33%	-55%	-59%	-61%	-58%	-39%	-58%
Decr. Ratio	-9%	-8%	-5%	-1%	-4%	-2%	-2%	-1%	+0%
AUCE	+6%	+26%	+28%	+289%	+65%	+124%	+607%	+4%	+344%
MCE	+8%	+30%	+33%	+84%	+70%	+113%	+175%	+4%	+150%
ENCE	+246%	+68%	+73%	+20%	+64%	+78%	+113%	-24%	+100%
c_v	+56%	-8%	-4%	-15%	-12%	-12%	-18%	-7%	-18%

Alchemy – size split

Table S7: Summary of metrics for the different methods (Alchemy, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	57.16	57.79	56.86	12.80	20.61	12.13	15.90	28.86	15.73
Error drop	1.91	0.35	0.3	12.29	1.53	11.41	8.65	1.01	8.03
Decr. Ratio	0.68	0.58	0.59	0.92	0.82	0.93	0.92	0.75	0.93
AUCE	49.16	43.08	43.04	1.34	19.72	3.89	5.29	34.01	3.96
MCE	0.98	0.79	0.79	0.06	0.34	0.06	0.09	0.59	0.08
ENCE	2.72e+05	22.78	22.40	0.72	3.67	0.45	0.90	8.90	0.81
c_v	4.05	0.84	0.84	1.30	0.93	1.22	1.19	0.87	1.15

Table S8: Summary of relative metrics for the different methods (Alchemy, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	+335%	+797%	+773%	+183%	+351%	+187%	+250%	+474%	+250%
Error drop	-38%	-95%	-95%	+54%	-82%	+33%	+9%	-86%	-5%
Decr. Ratio	-14%	-35%	-32%	+0%	-10%	+3%	+3%	-19%	-3%
AUCE	+2%	+19%	+20%	-86%	+13%	+58%	-41%	+5%	-43%
MCE	+3%	+25%	+25%	-62%	+21%	-20%	-36%	+7%	-33%
ENCE	+65318%	+156%	+153%	-55%	+42%	-53%	-33%	+36%	-32%
c_v	+112%	-36%	-37%	-24%	-42%	-30%	-34%	-27%	-34%

PDBbind – time split

Table S9: Summary of metrics for the different methods (PDBbind, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	26.11	14.14	14.37	31.15	14.67	12.56	23.05	12.96	11.52
Error drop	1.12	1.36	1.38	1.40	1.79	1.57	1.24	1.58	1.74
Decr. Ratio	0.64	0.87	0.87	0.65	0.96	0.89	0.80	0.95	0.94
AUCE	49.5	48.1	47.76	49.50	49.09	48.62	49.47	49.12	48.32
MCE	0.99	0.92	0.92	0.99	0.95	0.92	0.98	0.95	0.92
ENCE	11.22	1.49	1.42	6.57	2.16	1.81	4.62	2.14	1.66
c_v	0.60	0.18	0.20	0.46	0.31	0.36	0.45	0.25	0.29

Table S10: Summary of relative metrics for the different methods (PDBbind, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	-66%	-79%	-78%	-58%	-77%	-78%	-66%	-80%	-81%
Error drop	+22%	-19%	-17%	-23%	-19%	-30%	-16%	-39%	-17%
Decr. Ratio	+121%	+129%	+156%	+210%	+182%	+68%	+70%	+137%	+81%
AUCE	+46%	+637%	+889%	+62%	+705%	+1277%	+87%	+434%	+890%
MCE	+74%	+736%	+922%	+90%	+764%	+1050%	+123%	+494%	+820%
ENCE	+244%	+645%	+914%	+161%	+731%	+1107%	+170%	+478%	+938%
c_v	+7%	-22%	-13%	-15%	-6%	+29%	-6%	-11%	+12%

Lipophilicity – chemical element split

Table S11: Summary of metrics for the different methods (Lipophilicity, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	33.07	35.97	33.54	31.29	35.38	30.86	29.44	34.53	29.26
Error drop	1.12	0.95	1.09	1.22	1.05	1.57	1.73	1.05	1.65
Decr. Ratio	0.53	0.27	0.42	0.6	0.18	0.6	0.6	0.27	0.6
AUCE	30.42	28.48	22.03	36.11	37.38	31.40	27.37	33.40	22.31
MCE	0.51	0.47	0.36	0.63	0.65	0.53	0.45	0.57	0.36
ENCE	2.41	2.03	1.21	3.71	4.31	2.33	1.70	2.80	1.13
c_v	0.46	0.24	0.34	0.45	0.16	0.47	0.47	0.22	0.52

Table S12: Summary of relative metrics for the different methods (Lipophilicity, out-of-domain).

	MC-Dropout			Ensembling			Bootstrapping		
	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.	Epi.	Ale.	Tot.
AUCO	+8%	+4%	+4%	+23%	+9%	+10%	+19%	+8%	+13%
Error drop	-29%	+40%	-4%	-65%	+17%	-64%	-70%	-31%	-72%
Decr. Ratio	+152%	+575%	+282%	+107%	+80%	+131%	+107%	+286%	+107%
AUCE	+36%	+76%	+136%	+49%	+119%	+200%	+69%	+22%	+103%
MCE	+38%	+74%	+125%	+57%	+124%	+212%	+67%	+27%	+89%
ENCE	+48%	+105%	+120%	+82%	+219%	+199%	+59%	+20%	+61%
c_v	-6%	-17%	-6%	-8%	-43%	+74%	-10%	-29%	+27%

Confidence-oracle error plots

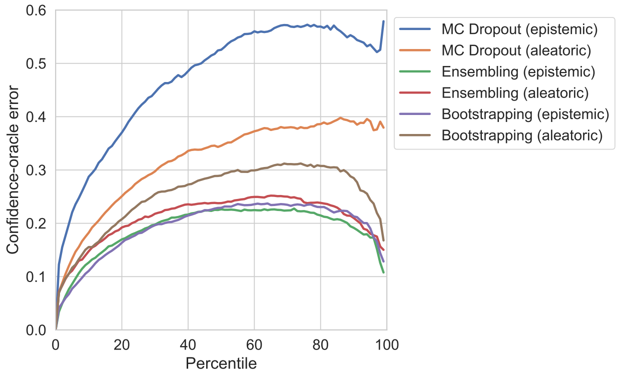

Figure S1: QM9, in-domain.

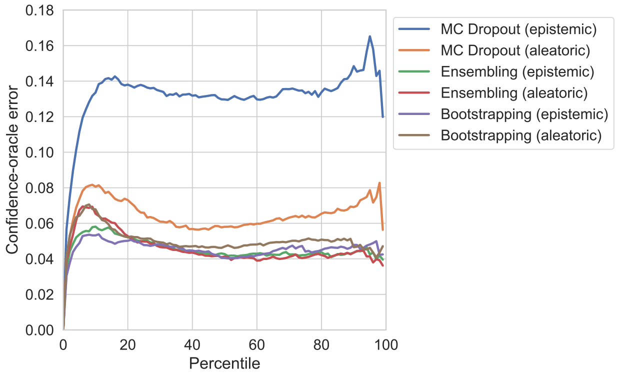

Figure S2: Alchemy, in-domain.

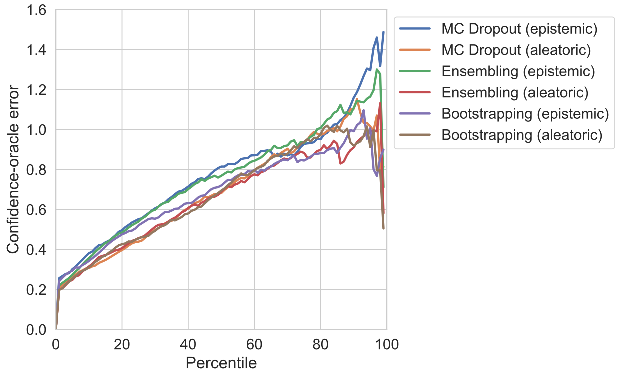

Figure S3: PDBbind, in-domain.

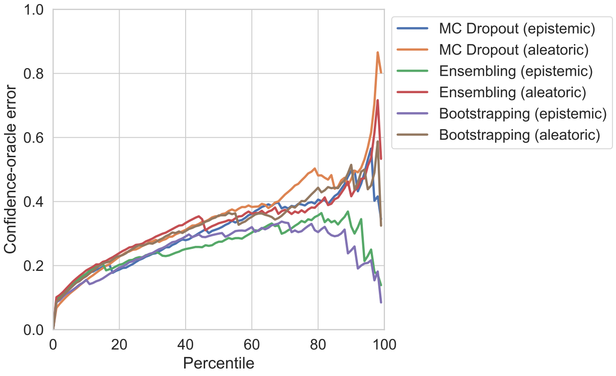

Figure S4: Lipophilicity, in-domain.

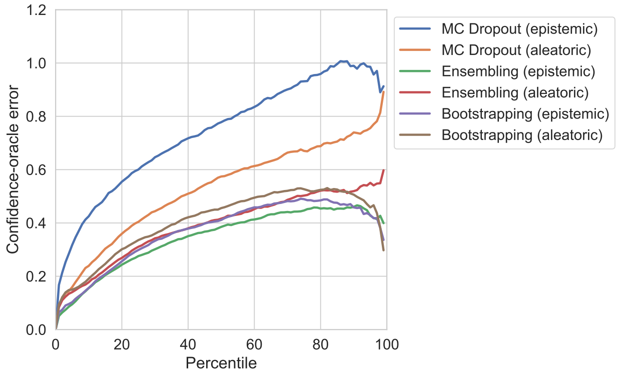

Figure S5: QM9, out-of-domain.

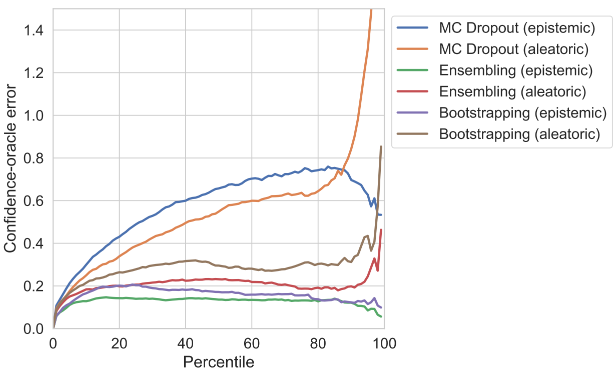

Figure S6: Alchemy, out-of-domain.

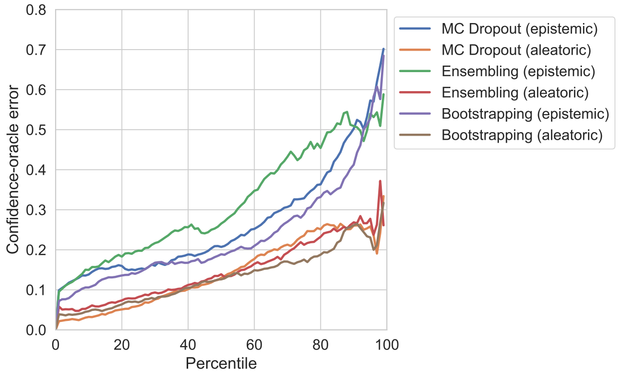

Figure S7: PDBbind, out-of-domain.

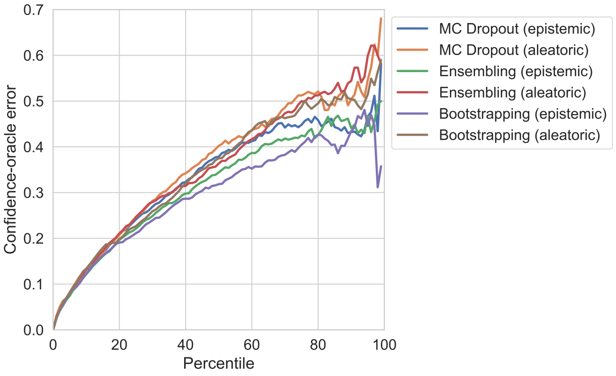

Figure S8: Lipophilicity, out-of-domain.

Experimental details

Implementation and experimental settings

We implemented the tested uncertainty estimation methods starting from the `chemprop` base model made available in Yang et al.[S1], based on the PyTorch framework in Python. The new software developed has been made available: `https://github.com/gscalia/chemprop/tree/uncertainty`.

Training hyperparameters for the model were selected using the `hyperopt` package, as described for the original `chemprop` implementation.[S1] The impact of hyperparameters on uncertainty estimates is further discussed in this Supporting Information.

We used $q = 100$ to compute all the confidence curves (i.e., we used percentiles). We used bins of 100 for QM9 and Alchemy, 50 for PDBbind and 25 for Lipophilicity to compute error-based calibration plots. We used smaller bins to account for smaller datasets.

Data preparation

Table S13: Summary of the datasets and split types for the experiments.

Dataset	**Category**	**Size**	**Property**	**Metric**	**Split**	
					in-domain	out-of-domain
QM9	quantum chemistry	130828	enthalpy $[kcal \cdot mol^{-1}]$	MAE	random	scaffold
Alchemy	quantum chemistry	103727	heat capacity $[cal \cdot (mol \cdot K)^{-1}]$	MAE	random	size
PDBbind	biophysics	11908	protein binding affinity $[-log\,(Kd/Ki)]$	RMSE	random	time
Lipophilicity	physical chemistry	4200	octanol/water distribution coefficients $[logD]$	RMSE	random	chemical element

Datasets and split types used in this work are summarized in Table S13. More details about each dataset, including its preparation, are discussed in the following.

In general, we relied as much as possible on what has already been published in the literature and used in recent benchmarks. All the data used in this work were made available in the respective publications. Furthermore, the out-of-domain splits used in our experiments are in line with what has been previously used,[S1–S3] with the exception of the chemical element split introduced in this work.

In both the in-domain and out-of-domain experiments, the same split was used to test all the different uncertainty estimation methods.

QM9

QM9 is part of the datasets included in the MoleculeNet benchmark.[S2] The formation enthalpies of 130,828 stable organic molecules composed of C, H, O, N and F atoms were used to train and test the model. These reference data were derived from the QM9 dataset, which was calculated at the B3LYP/6-31G(2df,p) level of theory with the rigid rotor-harmonic oscillator approximation (RRHO).[S4] As discussed in previous work, these calculated enthalpies are themselves associated with significant errors, primarily due to weaknesses of B3LYP such as the absence of long-range dispersion interaction but also the lack of rotor or conformer corrections in the calculations.[S5–S8] We note that it is possible to use a small amount of high-accuracy coupled cluster training data via a transfer learning approach to minimize the influence of DFT errors. Interested readers are referred to the recent work of Grambow et al.[S9] In this work, we used the QM9 data as is without any attempt to correct its errors in order to investigate the effects of aleatoric uncertainties.

Data were prepared as described in Grambow et al.[S9]. Readers are referred to this work for additional details and the original dataset. We used a 80:10:10 split for training, validation, and test sets, both in the in-domain and out-of-domain settings. Random splitting was used for in-domain analysis, while scaffold splitting was used for out-of-domain analysis.

For scaffold splitting, molecules are split into bins based on their Murcko scaffold, with each bin belonging to only one among training, validation, and test set. We used scaffold splitting as described and implemented in Yang et al.[S1].

PDBbind

PDBbind is part of the datasets included in the MoleculeNet benchmark.[S2] For this work, since we are interesting in evaluating *scalable* uncertainty estimation methods, we used the "full" version of PDBbind[S2] which includes the largest amount of molecules.

We used data as distributed in Yang et al.[S1]. We used a 80:10:10 split for training, validation, and test sets, both in the in-domain and out-of-domain settings. Random splitting was used for in-domain analysis. For out-of-domain analysis, we used *time splitting* as suggested in Wu et al.[S2], where the model trained on molecules published before a certain year is tested on molecules published after that year.

Lipophilicity

Lipophilicity is part of the datasets included in the MoleculeNet benchmark.[S2] We used data as distributed in Yang et al.[S1].

We used a random 80:10:10 split for training, validation, and test sets for the in-domain experiment.

For the out-of-domain experiment, we introduced a *chemical element* split. For this, we split the Lipophilicity dataset between F-containing molecules (901 elements) and the rest (3299 molecules). The latter subset was used for training and validation (90:10 split), while F-containing molecules were only used as test set.

Alchemy

Alchemy is a quantum chemistry dataset for benchmarking AI models recently published in the context of the Alchemy Contest.[S3] It expands the size and diversity of existing quantum datasets (namely, QM9). Interested readers are referred to the original paper for more details about this dataset and the comparison with QM9. For this work we used the heat capacities of 103,567 molecules. Data were prepared as follows.

We split our data based on the *size split* described in Chen et al.[S3]. A first subset (ALCHEMY-SMALL) includes 99,776 smaller molecules, with only $\approx 5\%$ of them having more than 10 heavy atoms. A second subset (ALCHEMY-LARGE) includes 3,951 molecules, each with more than 10 heavy atoms. Notice that this is the only split used in this work that does not lead to two totally disjointed sets, since a very small fraction of molecules with more than 10 heavy atoms is included also in the training set, in line with Chen et al.[S3].

For the in-domain tests, ALCHEMY-SMALL was split randomly with a 80:10:10 split for training, validation, and test sets. This ensures that each set has the same distribution and is mostly composed of smaller molecules.

For the out-of-domain tests, we used *the same* models trained for the *in-domain* experiments and we tested them on ALCHEMY-LARGE.

Additional discussion

Hyperparameters

It is worth noting that uncertainty estimates obtained through DNN-based methods are, in general, affected by the *training hyperparameters* of the network. Indeed, aleatoric uncertainty, being an output of the network, depends on the training hyperparameters just as any other network output does. Epistemic uncertainty is affected by the network weights — which affect output variability — and are a function of the training hyperparameters.

In this work, we selected the DNN training hyperparameters as follows: i) Hyperparameters were tuned by minimizing the loss of the base network (using the `hyperopt` package). ii) Once optimal hyperparameter were selected, the resulting network was extended with the different uncertainty estimation methods. This allowed investigating the performance of the uncertainty estimation methods as used in practice.

On top of training hyperparameters, uncertainty estimation methods themselves could introduce additional hyperparameters. For example, MC-Dropout introduces the dropout probability p that directly affects the magnitude of the predicted epistemic uncertainty.[S10]

As previously mentioned, for this work we selected *practical* uncertainty estimation methods. Accordingly, all of the selected methods do *not* introduce additional hyperparameters affecting uncertainty estimates (with the exception of the sampling size, discussed later). Indeed, ensembling is intrinsically hyperparameter free,[S11] and, in particular, its usage in conjunction with early stopping does not require defining new hyperparameters (in contrast, for example, to anchored ensembling[S12] discussed next). Bootstrapping increases the diversity in the models without introducing new hyperparameters. Regarding MC-Dropout, we avoided the standard method to avoid introducing a hyperparameter with a direct impact on uncertainty estimates. Instead, we used Concrete Dropout,[S10] which allows automatically tuning per-layer dropout probability p during training, with the goal of converging to the *optimal* probability for optimal epistemic uncertainty estimates. As described in Gal et al.[S10], Concrete Dropout results in a performance that is comparable to a grid search for p. While avoiding selecting the dropout probability p, Concrete Dropout requires setting another hyperparameter: the prior length scale l (which, in turn, sets the weight regularizer and the dropout regularizer; see the original paper for more details about their meaning). However, the prior length scale l is a training hyperparameter and, as such, can be jointly optimized with the other training hyperparameters on a validation set. Once optimized, the probability p leading to optimal epistemic uncertainty was automatically found during training.

As explained in the Methods section, all the uncertainty estimation techniques considered in this work were based on approximate Bayesian inference, which relies on Monte Carlo integration over (approximate) samples of the posterior distributions. For this reason, the number of samples M is an inherent hyperparameter of all the considered methods. In this respect, even though it is not possible to formally identify the precise number of samples which result in “good enough” performance, previous works highlighted how uncertainty estimates quickly converge, with diminishing returns beyond 5-15 samples.[S13,S14] In this work we used 15 samples for ensembling and bootstrapping, 150 for MC-Dropout. MC-Dropout employs weight sharing between different instances and it does not require a separate training for each one, allowing a larger M in practice. This difference in the number of instances reflects realistic condition of use. Preliminary experiments increasing the number of samples did not report significant variations in the outcomes, except for an asymptotically smaller general improvement in all the metrics for all the tested methods.

Anchored Ensembles and Early Stopping

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.[S12,S15]

Anchored ensembling[S12] modifies traditional ensembling by leveraging the *randomised MAP sampling* technique. This technique exploits the fact that injecting some noise in the loss function of a MAP estimate allows sampling from the true posterior. Therefore, an ensemble of such models is a simple and scalable approach for approximate Bayesian inference.

It is known that the commonly used L_2 regularization for NN (weight decay) corresponds to the MAP estimate with Gaussian priors,[S16] which can be interpreted as reducing the magnitude of weights for which the network does not express a strong preference. The anchored ensembling algorithm proposes to add noise to this loss function by changing the means of the priors. For regression, this leads to the following loss for the i-th model in the ensemble:

$$\mathcal{L} = \frac{1}{N}\|\mathbf{y} - \tilde{\mathbf{y}}^{(i)}\|_2^2 + \frac{1}{N}\lambda\|\theta^{(i)} - \theta_0^{(i)}\|_2^2 \tag{1}$$

where $\mathbf{y}$ are the target outputs and $\theta_0^{(i)}$, which equals zero for standard L_2 regularization, is the mean of the prior of the i-th model.

Following this approach, each model in the ensemble has its parameters *anchored* to a different $\theta_{0,i}$, and this promotes the diversity of the solutions reached by the different models.

An important limitation of this approach is the need for additional hyperparameters that must be tuned. They include at least the regularization coefficient λ — that expresses the ratio between data variance and prior variance of the weights — and the noise distribution $\theta_{0,i} \sim \mathcal{N}(0, \Sigma_0)$. As originally described,[S12] the algorithm also employs a regularization matrix $\mathbf{\Gamma}$ instead of the scalar λ, to allow specifying per-layer regularization.

The work presented in Duvenaud et al.[S15] gives an interesting interpretation to a commonly exploited regularization method — *early stopping* — as approximate nonparametric Bayesian variational inference. In particular, they show how training a model to minimize the negative log-likelihood with stochastic gradient descent (SGD)[i] can be interpreted as obtaining the approximate posterior $q_t(\theta)$ parametrized by the number t of SGD steps, and demonstrate how early stopping leads to an optimal $\tilde{t}$. Within this context, the initial distribution of the model $p(\theta_0)$ is interpreted as the *prior*.

In practice, $q_{\tilde{t}}(\theta)$ allows sampling from the variational posterior, and therefore ensembling different random restarts allows obtaining independent samples from the posterior, that can then be used as in traditional ensembling. Even if the approach, as originally described, does not take into consideration SGD with momentum, recent work also shows how SGD with momentum can be interpreted as Bayesian inference.[S17]

Not only is this approach practical, but ensembling with early stopping is usually already exploited for property prediction in state-of-the-art systems.[S1] In this work we use it as a Bayesian alternative for uncertainty estimation.

We can draw a parallelism between the two approaches described above. It has been shown that early stopping for NNs is conceptually similar to L_2 regularization, while an exact

[i] The approach is compatible also with minibatches.

equivalence holds in the simpler case of a linear model with a quadratic loss function.[S16] Intuitively, both approaches restrict the optimization procedure to the vicinity of a pre-defined value — θ_0 for L_2 regularization, the initial configuration for early stopping. In our case, we noticed that these two values have the same role of *prior* in the two approaches,[S12,S15] highlighting an interesting similarity. Even though they are based on different theoretical foundations, in practice both the approaches increase the diversity in the ensembled instances by injecting some randomness into their regularization. An intrinsic advantage of early stopping over weight decay is that early stopping automatically determines the correct amount of regularization, instead of requiring external hyperparameter optimization.[S16]

Therefore, given the objective of this paper of evaluating scalable and practical uncertainty quantification techniques, we used ensembling with early stopping for our extensive tests. Anchored ensembling and the impact of different priors for uncertainty estimation will be the subject of future work.

Additional analyses

Aleatoric and epistemic uncertainty correlation

In Table S14 the Spearman rank-order correlation and the Pearson correlation between aleatoric and epistemic uncertanties are shown for each dataset and uncertainty estimation method. Values are shown as Spearman/Pearson correlation.

Table S14: Spearman/Pearson correlation between aleatoric and epistemic uncertainties

	MC-Dropout	Ensembling	Bootstrapping
QM9	0.25/0.01	0.54/0.66	0.71/0.28
Alchemy	0.47/0.01	0.86/0.30	0.87/0.15
PDBbind	0.26/0.20	0.69/0.04	0.18/0.12
Lipophilicity	0.74/0.68	0.17/0.21	0.24/0.35

Aleatoric uncertainty and ground truth error correlation

Since 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 DFT calculations (QM9 dataset), the ground truth can be better approximated by high level quantum chemistry calculations like coupled cluster theory with a generous basis set[S9,S18] or experimental results.

For this experiment, we collected 826 enthalpy values derived from experiments and higher level calculations (CCSD(T)-F12/cc-pVDZ-F12) to examine whether the estimated aleatoric uncertainties correlate with the nois in training data, i.e., the differences between the DFT enthalpies calculated at the B3LYP level of theory and the true formation enthalpies. These more accurate enthalpy values were collected from the work of Grambow et al.[S9] and can be found in the supporting information of the original paper. Interested readers are referred to the detailed description of how the data were derived in the Datasets section of Grambow et al.[S9].

Aleatoric uncertainties and errors with respect to ground truth values report a low correlation: Pearson correlation ≈ 0.04, Spearman correlation ≈ 0.13. A possible explanation for this behavior is discussed in the paper.

In-domain and out-of-domain epistemic uncertainty median

In Table S15 the uncertainty median is shown for each dataset and uncertainty estimation method as *in-domain median/out-of-domain median.*

Table S15: In-domain/out-of-domain uncertainty median

	MC-Dropout	Ensembling	Bootstrapping
QM9	0.0027/0.0007	0.30/0.67	0.54/1.21
Alchemy	3.85e-06/1.06e-06	0.009/0.302	0.107/0.385
PDBbind	0.115/0.035	0.153/0.076	0.250/0.139
Lipophilicity	0.042/0.040	0.024/0.018	0.056/0.057

References

(S1) 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.

(S2) Wu, Z.; Ramsundar, B.; Feinberg, E.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513–530.

(S3) 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. *arXiv preprint arXiv:1906.09427* **2019**,

(S4) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.

(S5) Cohen, A. J.; Mori-Sánchez, P.; Yang, W. Challenges for Density Functional Theory. *Chem. Rev.* **2012**, *112*, 289–320.

(S6) Simm, G. N.; Reiher, M. Systematic Error Estimation for Chemical Reaction Energies. *J. Chem. Theory Comput.* **2016**, *12*, 2762–2773.

(S7) Proppe, J.; Husch, T.; Simm, G. N.; Reiher, M. Uncertainty quantification for quantum chemical models of complex reaction networks. *Faraday Discuss.* **2017**, *195*, 497–520.

(S8) Li, Y.-P.; Bell, A. T.; Head-Gordon, M. Thermodynamics of Anharmonic Systems: Uncoupled Mode Approximations for Molecules. *J. Chem. Theory Comput.* **2016**, *12*, 2861–2870.

(S9) 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.

(S10) Gal, Y.; Hron, J.; Kendall, A. Concrete dropout. Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017; pp 3584–3593.

(S11) 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.

(S12) Pearce, T.; Zaki, M.; Brintrup, A.; Neel, A. Uncertainty in neural networks: Bayesian ensembling. *arXiv preprint arXiv:1810.05546* **2018**,

(S13) 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. *arXiv preprint arXiv:1906.02530* **2019**,

(S14) Gustafsson, F. K.; Danelljan, M.; Schön, T. B. Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision. *arXiv preprint arXiv:1906.01620* **2019**,

(S15) 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.

(S16) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.

(S17) Mandt, S.; Hoffman, M. D.; Blei, D. M. Stochastic gradient descent as approximate bayesian inference. *J. Mach. Learn. Res.* **2017**, *18*, 4873–4907.

(S18) 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*, 1–8.