# Quality of uncertainty estimates from neural network potential ensembles

Leonid Kahle[*] and Federico Zipoli[†]

*National Centre for Computational Design and Discovery of Novel Materials MARVEL, IBM Research Europe, Zurich, Switzerland*

(Received 13 August 2021; accepted 3 January 2022; published 21 January 2022)

Neural network potentials (NNPs) combine the computational efficiency of classical interatomic potentials with the high accuracy and flexibility of the *ab initio* methods used to create the training set, but can also result in unphysical predictions when employed outside their training set distribution. Estimating the epistemic uncertainty of a NNP is required in active learning or on-the-fly generation of potentials. Inspired from their use in other machine-learning applications, NNP ensembles have been used for uncertainty prediction in several studies, with the caveat that ensembles do not provide a rigorous Bayesian estimate of the uncertainty. To test whether NNP ensembles provide accurate uncertainty estimates, we train such ensembles in four different case studies and compare the predicted uncertainty with the errors on out-of-distribution validation sets. Our results indicate that NNP ensembles are often overconfident, underestimating the uncertainty of the model, and require to be calibrated for each system and architecture. We also provide evidence that Bayesian NNPs, obtained by sampling the posterior distribution of the model parameters using Monte Carlo techniques, can provide better uncertainty estimates.

DOI: 10.1103/PhysRevE.105.015311

## I. INTRODUCTION

Interatomic potentials (IPs) are a long-established method to describe the potential energy and forces in atomic systems and have provided important insights into the physics of atomic structures in the past seven decades [1–4]. A notable advantage of IPs over first-principles approaches such as density functional theory (DFT) [5,6] and first-principles molecular dynamics [7] is their higher computational efficiency, allowing the simulation of larger systems over longer timescales. In recent years, machine-learning techniques have been successfully applied to develop force fields for atomistic simulations [8–10]. Among other machine-learning techniques, (deep) neural networks (NN) can be used as force and energy predictors, resulting in neural network potentials (NNPs), which have led to significant advances in atomistic simulation, combining high computational efficiency with great flexibility and accuracy [11–13]. One assumption behind IPs is that the energy of the system can be described as a sum over atomic energies: $E = \sum_i E_i$. Another assumption behind most NNPs is that the atomic energy is a function of the local neighborhood of that specific atom, that is, all atoms within a radial cutoff distance, $r_c$. Any IP, and therefore also any NNP, should be invariant to translation, rotation, and permutation of atoms in the system. This is most commonly achieved by constructing a descriptor of the atomic environment that displays those symmetries.

One approach to generate NNPs is to use NNs to learn parameters of a physics-inspired functional form, for example, to tackle long-range electrostatic contributions to energy and forces [14–17], which should be relevant in ionic systems. However, recent work has shown that NNPs are capable of accurate predictions in ionic systems without explicitly including long-range interactions [18,19]. NNPs can also be used directly for the prediction of energies, forces, and stresses, without enforcing a particular functional form. As a first example, Behler and Parrinello implemented a NNP [20] using atomic symmetry functions as an SO(3) invariant descriptor. Another example is DEEPMD developed by Zhang *et al.*, where the relevant descriptors are learned during the training and SO(3) invariance is encoded in the mathematical form of the first layers [21].

Convolutional neural networks and graph convolutional neural networks achieve invariance via the application of tailored convolution filters [22–26], one prominent example being Schnet [27,28]. A recent addition to the plethora of neural network potentials is tensor-field NNPs, which encode SO(3) equivariance in the convolutional operations [29,30]. Batzner *et al.* released NEQUIP [31], an efficient implementation of a NNP using equivariant convolutions.

While the architecture of the network and the type and size of the input descriptor are of great importance to build an accurate NNP, training the model with an extensive data set of high quality is just as relevant. One common approach is to sample configurations with *ab initio* methods, such as Born-Oppenheimer molecular dynamics at different thermodynamic conditions [11,32]. Another possibility is an active-learning approach, a repeated cycle of exploration, labeling, and (re)training of a model as follows:

(i) *exploring* configurations space, e.g., *via* molecular dynamics, using a NNP obtained in the previous iteration or from an initial training,

(ii) *labeling* a subset of configurations using an *ab initio* method,

(iii) *training* a new NNP with the labeled configurations.

[*] Present address: Materials Design Inc., San Diego, California 92131, USA; lkahle@materialsdesign.com

[†] fzi@zurich.ibm.com

As a first example, Marcolongo *et al.* [18] trained a NNP in this iterative fashion to model Li-ion diffusion in solid-state electrolytes, where the configurations in the second step were selected randomly from the molecular-dynamics trajectory of the first step and were therefore sampled according to the Boltzmann distribution. A possible disadvantage of this method is that data selected randomly could be redundant, adding no new information to the training set. Several works in the past have concentrated on detecting configurations that provide additional data that are not already covered by the training set [33,34]. This can either enable active exploration of configuration space to obtain better coverage, or so-called on-the-fly training during a molecular dynamics or Monte Carlo trajectory, proposed as early as 1997 by Vita and Car [35] and used or developed further in subsequent work [36]. Using NNPs in such a scenario can be challenging, as it is very hard to predict when a model is outside the training set distribution [37–39]. Neural networks are often described as "universal function approximators." That flexibility directly results in the difficulty to control prediction trends when departing from the training set distribution.

Detecting configurations that provide additional data translates to detecting atomic environments of high epistemic uncertainty. While aleatoric uncertainty is due to noise inherent in the labels, epistemic uncertainty is due to lack of data and is the subject of this work [40–42]. In the remainder, we simply refer to uncertainty, meaning epistemic uncertainty originating from data scarcity. One approach to estimate the uncertainty of the prediction of neural networks relies on the prediction of several instances, so-called neural-network ensembles, where the uncertainty is estimated from the deviation of the output of different models that were trained on the same data. The members of the ensemble ideally have different architectures or, as a minimal criterion, have the same architecture but are initialized independently. Ensemble networks have led to promising results in many machine-learning applications [43–45]. Ensemble NNPs were, for example, used by Zhang *et al.* [37], but also other work has relied on similar approaches to select configurations of high model uncertainty [34,46,47]. However, the uncertainty derived from ensembles is not, strictly speaking, a Bayesian uncertainty estimate [44,48,49]. The same applies to dropout techniques, which have also been used to estimate uncertainty in NNPs [38]. The method relying on the least approximations to obtain uncertainty estimates are Bayesian NNs [50–52], obtained by sampling the posterior distribution of NN parameters. Bayesian NNPs, which do not rely on ensembles or dropout, have not yet been developed to the best of our knowledge, most likely due to the increased complexity and high computational cost of exploring the posterior distribution of the NNP parameters using expensive sampling techniques such as Hamiltonian Monte Carlo [45,53–55].

In this work, we explore the relationship between the uncertainty predicted by NNP ensembles and the true error of the prediction, showing that ensembles are prone to common bias and overconfident predictions, depending on model architecture and the system, requiring careful calibration of the uncertainty. Furthermore, we compare choosing configurations based on the predicted uncertainty, as proposed by Zhang *et al.* [37], to random sampling during exploration, as done by Marcolongo *et al.* [18]. In Sec. II we explain the methods used in this work. The results are shown and discussed in Sec. III and we present our conclusion in Sec. IV.

## II. METHODS

### A. Systems and NNPs

In order to allow for general conclusions, we used training and validation data from four very different atomic systems: an atomic dimer, an aluminum surface slab, bulk liquid water, and a benzene molecule in vacuum. The training and validation data were produced with a Lennard-Jones potential, density functional theory, or the highly accurate coupled-cluster single-, double-, and perturbative triple-excitations method CCSD(T). The neural-network architectures we tested were a custom-built neural network, the DEEPMD [21] framework, and the NEQUIP [31] potential.

As a first case study, we trained a NNP on the potential energy surface (PES) of an atomic dimer, i.e., two atoms of the same species in vacuum. For this purpose, we chose a parametrized model as the ground truth, namely, the Lennard-Jones IP, where the energy is a function of the interatomic distance $r$:

$$E(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right], \tag{1}$$

where we set $\epsilon = 10.34\ \mathrm{meV} = 120\ \mathrm{K} \cdot k_B^{-1}$ and $\sigma = 0.34$ nm (which are the parameters used by Rahman [2] to study atomic correlation in liquid argon). In the range of interest (0.3 to 0.7 nm) we chose randomly ten training points according the Boltzmann weight $p(E) \propto e^{\frac{E}{k_B T}}$ with $T = 20$ K. Such a simple system allows for a significant reduction of the dimensionality of the problem and for a custom-built NNP, leading to better interpretability of the results.

We implemented a custom NNP using the `pytorch` framework, with an input size of 1 (the interatomic distance), a hidden layer of 64 neurons, and an output layer of 1 (the energy). A neural network requires a nonlinear activation function, and common choices of such activation functions are the hyperbolic tangent tanh, the sigmoid function, rectified linear units (ReLU), continuously differentiable exponential linear units (CELU) [56], and Gaussian error linear units (GELU). These activation functions are shown in Fig. S1 of the Supplemental Material [57]. We implemented NNPs with each type of activation function, as well as one NNP where the activation function of each neuron was randomly chosen among aforementioned functions. We trained each NNP with the Adam optimizer for 50 000 steps with a learning rate of $10^{-3}$.

We used the smooth edition of DEEPMD [21], as implemented in the DEEPMD-KIT package (version 1.3.3), to build a potential for bulk and surface aluminum (Al) and for liquid water. The DEEPMD NNPs had three layers for the local-embedding network, their sizes being 32, 64, and 128, respectively, and three layers for the fitting network, each consisting of 256 neurons. The training was performed using stochastic-gradient descent with an exponentially declining learning rate.

To have a more realistic and complex scenario than the atomic dimer, we trained the DEEPMD potential on forces and energies from an Al(100) surface with an adatom of the same species. The bulk portion of the system is representative of metals, while the presence of the surface leads to additional complexity. Extensive FPMD simulations of this system were performed by Nguyen *et al.* of a system of 295 atoms [58], consisting of six layers of Al (49 atoms per layer) with an additional atom on the surface. The calculations were performed with Born-Oppenheimer molecular dynamics as implemented in the QUANTUM ESPRESSO (QE) distribution [59] using the Perdew-Burke-Ernzerhof exchange-correlation functional [60] in the canonical (NVT) ensemble. The temperatures investigated were 300–600 K in steps of 100 K, and additionally 800 and 1000 K, with ≈30 ps of dynamics collected for every simulation. We selected evenly distributed snapshots from each trajectory: 144 snapshots from the trajectory at 1000 K, 72 snapshots at 800 K, and 36 snapshots from the simulations at 300–600 K. For each temperature, 12 snapshots were retained for validation in order to obtain a temperature-dependant validation error. We used the remaining snapshots (training set) to train a NNP using DEEPMD [21], with the parameters given in Fig. S2 of the Supplemental Material [57]. An additional 91 snapshots were created from an exploration with the NNP at 1000 K and added to the training set. Training the DEEPMD NNP with this data set, we obtained a trained model, $\mathcal{M}_{\mathrm{Al}}$. One hundred additional snapshots were sampled by exploring with $\mathcal{M}_{\mathrm{Al}}$ the same system at 1500 K in order to obtain an additional validation set, $\mathcal{D}^{\mathrm{Al}}_{1500}$.

We also trained a NNP with DEEPMD for the commonly studied system of bulk water, representative of highly ergodic liquid systems with high directionality of bonds, resulting in the complex chemistry of water. We used the data set created by Cheng *et al.* [61,62] of 64 water molecules in a cubic periodic cell. We split the data set into four equally large data sets $\mathcal{D}^{\mathrm{H_2O}}_{0\ldots3}$ of 300 configurations each according to the potential energy of the configuration, where the potential energy $E_i$ of a configuration in set $k$ is smaller than (or equal to) the energies of a configuration in set $k+1$:

$$E_i \leqslant E_j \forall i \in \mathcal{D}^{\mathrm{H_2O}}_{k} \wedge \forall j \in \mathcal{D}^{\mathrm{H_2O}}_{k+1}. \tag{2}$$

The energy distribution of the four sets of configurations is shown in Fig. S8 of the Supplemental Material [57]. The model was trained with $\mathcal{D}^{\mathrm{H_2O}}_{0}$ and $\mathcal{D}^{\mathrm{H_2O}}_{1\ldots3}$ were kept as validation sets. The parameters employed for DEEPMD (Al and water) are given in Fig. S2 of the Supplemental Material [57].

Last, we employed NEQUIP, developed by Batzner *et al.* [31], to fit a potential for the benzene molecule in vacuum, which is a good representative of covalently bonded systems. Our NEQUIP NNP had three convolution layers, and eight was the dimension of the hidden layer and the number of features. We used a data set of 1500 configurations of $C_6H_6$, produced by Chmiela *et al.* [63], who sampled configurations using MD in the canonical (NVT) ensemble at 500 K, and recalculated forces and energies using CCSD(T) for this data set. We ordered the set of 1500 configurations by potential energy and split the data into six batches of equal size, obtaining independent sets $\mathcal{D}^{\mathrm{C_6H_6}}_{0..5}$ of 250 configurations each. The energy distribution of the six sets of configurations is shown in Fig. S9 of the Supplemental Material [57]. Also in this case, the potential energy $E_i$ of a configuration in set $k$ is smaller than (or equal to) the energies of a configuration in set $k+1$:

$$E_i \leqslant E_j \forall i \in \mathcal{D}^{\mathrm{C_6H_6}}_{k} \wedge \forall j \in \mathcal{D}^{\mathrm{C_6H_6}}_{k+1}. \tag{3}$$

As in the case study of liquid water, we trained the NNP on the set of lowest-energy configurations ($\mathcal{D}^{\mathrm{C_6H_6}}_{0}$) and used $\mathcal{D}^{\mathrm{C_6H_6}}_{1\ldots5}$ as validation sets. The Adam optimizer was used for the training, with a learning rate of 0.01, and the training was stopped after 500 epochs (complete passes over the training data). All parameters are reported in Fig. S3 of the Supplemental Material [57].

### B. NNP ensembles

Unless stated differently, we trained $M = 8$ NNPs with the same data to estimate the ensemble uncertainty, which we found to be converged at that ensemble size. These models were initialized independently *via* the input of different seeds, which affected both the parameter initialization and the order of sample selection during stochastic gradient descent, where applicable. From the ensemble of NNPs, we computed the mean prediction and the standard deviation $\sigma$ for every force component $F_{I\alpha}$:

$$\bar{F}_{I\alpha} = \frac{1}{M}\sum_{m}^{M} F^{m}_{I\alpha}, \tag{4}$$

$$\sigma_{I\alpha} = \sqrt{\frac{1}{M}\sum_{m}^{M}\left(F^{m}_{I\alpha} - \bar{F}_{I\alpha}\right)^2}, \tag{5}$$

where the subscripts $I$ and $\alpha$ give atomic indices and spatial dimensions, respectively, and $m$ iterates over models of the ensemble. $\sigma_{I\alpha}$ is the predicted model uncertainty of the force in direction $\alpha$ of atom $I$.

The error $\epsilon$ of the model was calculated (on a validation/test set) from the difference of the predicted (mean) force $\bar{F}_{I\alpha}$ and the *ab initio* $F^{ai}_{I\alpha}$ force (the label):

$$\epsilon_{I\alpha} = \left|F^{ai}_{I\alpha} - \bar{F}_{I\alpha}\right|. \tag{6}$$

### C. Bayesian NNP

The simplicity of the NNP for the atomic dimer and the small amount of training data (ten one-dimensional training points) allowed for a Bayesian estimate of the posterior probability of the weights $\theta$, given the data $D = \{(x_i, y_i)\}$:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \tag{7}$$

where $p(D|\theta)$ is the likelihood of the data, $p(\theta)$ is the prior, and $p(D)$ the marginal likelihood. The likelihood of the data given the parameters $\theta$ of a function $f$, assuming Gaussian white noise of variance $\sigma^2$ in the observations, is given by

$$\begin{aligned} p(D|\theta) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{[y_i - f(x_i|\theta)]^2}{2\sigma^2}} \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{-\frac{\sum_{i=1}^{n}[y_i - f(x_i|\theta)]^2}{2\sigma^2}} \\ &\propto e^{-\frac{\mathcal{L}(D|\theta)}{T}}, \end{aligned} \tag{8}$$

where $\mathcal{L}(D|\theta) = \frac{1}{n}\sum_i^n[y_i - f(x_i|\theta)]^2$ is the mean-squared error loss function and $n$ is the number of samples.

The prior of the parameters, $p(\theta)$, was set to the normal distribution $\mathcal{N}(0, \sigma_p^2)$ with $\sigma_p^2 = 1000$, which we evaluated *via* a curvature criterion. We ignored $p(D)$, the marginal likelihood, since it is independent of $\theta$. Inserting Eq. (8) and the Gaussian prior into Eq. (7), we obtain an energy or cost function of the parameters of the network that can be sampled at a fictitious temperature $T$ [50,54,55,64]:

$$p(\theta|D, T) \propto e^{-\frac{1}{T}\mathcal{L}(D|\theta) - \frac{\theta^2}{2\sigma_p^2}}. \tag{9}$$

We sampled this landscape using the Hamiltonian Monte Carlo (HMC) [65], with the Hamiltonian dynamics being integrated with the velocity Verlet integrator [66]. We set the fictitious temperature to 0.05 (resulting in a cold posterior [64]) and used 10 000 steps of Hamiltonian dynamics, with a time step of 0.015, which resulted in an acceptance rate of 0.675 of Monte Carlo moves, close to the optimal acceptance rate of 0.65 [65]. We sampled the parameter distribution by selecting 100 models from the MC trajectory to calculate mean and uncertainty as for the NNP ensembles.

### D. Statistical analysis

We interpret the ensemble uncertainty $\sigma_{I\alpha}$ as a predictor for the error $\epsilon_{I\alpha}$. A straightforward approach, also employed by Zhang *et al.* [67], is to classify configurations based on an uncertainty threshold, $\sigma_{\max}$. Configurations with all $\sigma_{I\alpha} < \sigma_{\max}$ were classified as low error and were classified as high-error configurations otherwise. The condition was given by $\epsilon_{I\alpha} < \epsilon_{\max}$. We note that $\sigma_{\max}$ and $\epsilon_{\max}$ did not have to be equal, in order to account for systematic underestimates. We formulated two distinct requirements for the uncertainty.

(i) A requirement for accurate results is that configurations that produce a high error, $\epsilon_{I\alpha} > \epsilon_{\max}$, due to lack of training data in that region of configuration space are detected *via* a high uncertainty, $\sigma_{I\alpha} > \sigma_{\max}$.

(ii) A computational requirement is to achieve high precision for high-error configurations, in order to avoid falsely selecting many configurations that are described well by the model.

The first requirement is given by the true positive rate (TPR), also called recall or sensitivity, which is given by the ratio of correctly positive prediction among all elements with the following condition:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{10}$$

where TP and FN refer to true positives ($\epsilon_{I\alpha} > \epsilon_{\max}$ and $\sigma_{I\alpha} > \sigma_{\max}$) and false negatives ($\epsilon_{I\alpha} > \epsilon_{\max}$ and $\sigma_{I\alpha} < \sigma_{\max}$), respectively. The second requirement, stating that we want to maximize the ratio of true high-error configuration to high-uncertainty configurations, is described by another metric, the precision or positive predictive value (PPV):

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{11}$$

where FP refers to the false positives ($\epsilon_{I\alpha} < \epsilon_{\max}$ and $\sigma_{I\alpha} > \sigma_{\max}$).

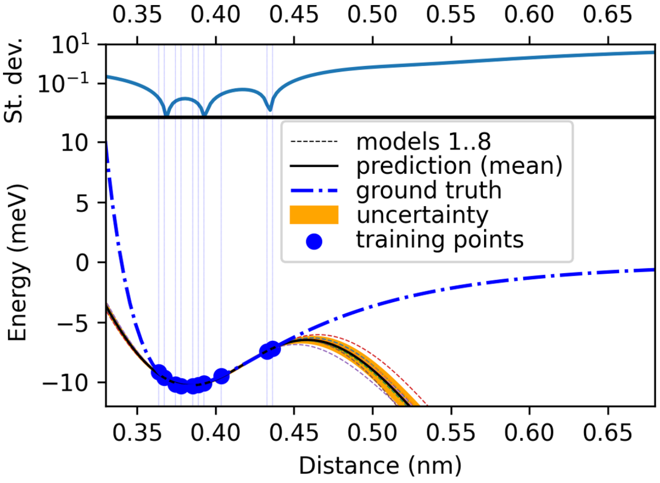

FIG. 1. The ground truth (blue dash-dotted lines) and the training points (blue dots and dotted vertical lines) are shown together with predictions from eight models trained on the same data but initialized differently. In the top panel we plot the logarithm of the standard deviation between the models. The model uses the hyperbolic tangent as an activation function. The top panel shows the standard deviation of the model prediction as a function of $r$ on a semilogarithmic scale.

## III. RESULTS AND DISCUSSION

### A. The atomic dimer

The custom NNP for the atomic dimer receives a scalar input (the distance), predicts the energy, and has 64 hidden units. The predictions of this model using the hyperbolic tangent (tanh) as the nonlinear activation function are shown for the range of $r$ from 0.33 to 0.68 nm in Fig. 1. The NNP ensemble mean is very accurate at predicting the PES in the region of training data, which we call the interpolation regime. In the extrapolation regime, where $r < 0.35$ nm or $r > 0.45$ nm, no data are supplied and the models deviate from the ground truth (blue dash-dotted lines). However, the members of the ensemble deviate in a very similar fashion, most probably due to common biases. This is not only the case when using this particular activation function: all activation functions (ReLU, CELU [56], GELU, and sigmoid) show the same behavior (see Fig. S4 of the Supplemental Material [57]). However, different activation functions result in different biases in the region of larger distances ($r > 0.45$ nm). Using tanh or sigmoid as activation functions leads to underestimates of the energy at $r > 0.45$ nm, whereas other activation functions (ReLU, CELU) lead to overestimates. We suspect that this behavior is because the networks with the latter activation functions are prone to extrapolate the slope from the closest training points. Using the uncertainty of the ensemble as a predictor for low accuracy [37] would result in this case study in false positives in data-scarce regions due to the bias in the model.

The NNP ensemble where every neuron within the hidden layer is randomly assigned one out of the different activation functions we study (tanh, sigmoid, GELU, ReLU, and CELU) produces predictions that are more diverse (see Fig. 2) outside the training set distribution. The ensemble displays higher variance of the output in the extrapolative region of

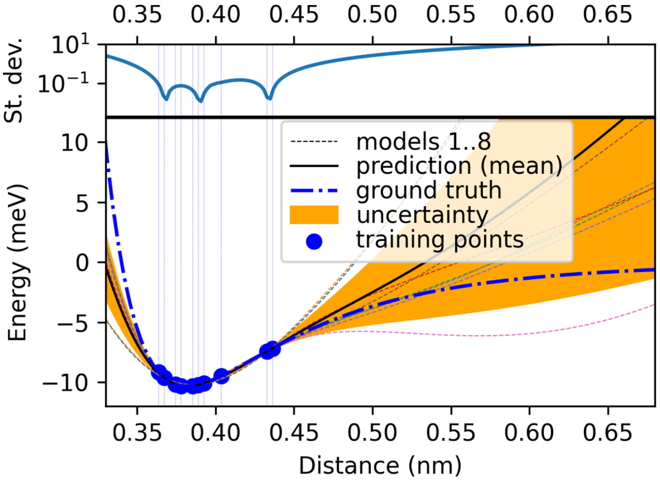

FIG. 2. Similar to Fig. 1, training data (blue dots) sampled from the ground truth (blue dash-dotted line) were used to train different models (dotted lines). The models are initialized independently and also have different architectures, ensured *via* a random assignment of activation function to every neuron in the hidden layer. The estimate of the uncertainty is improved due to reduced bias.

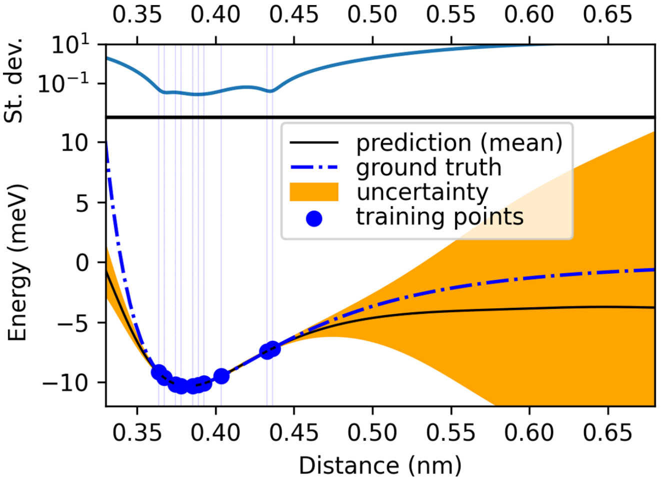

FIG. 3. Similar to Fig. 1, the training data (blue dots) are sampled from the ground truth (blue dash-dotted line). The dotted lines show the predictions of models sampled from the posterior parameter distribution, and the orange area depicts one standard deviation from the mean.

$r > 0.45$ nm, which we interpret as evidence that bias has been reduced. The region of smaller distances ($r < 0.35$ nm) is biased towards slopes of smaller magnitude. Compared to the results of the ensembles with the same architecture, diversifying the NNP ensemble results in lower bias, which is also discussed by Jeong *et al.* [47]. The uncertainty of ensembles with a varying model architecture is a better predictor for the accuracy of the model when predicting unseen data and could be useful in active learning. Current approaches employing NNP ensembles [37,46] do not use varying architectures of the ensemble members to reduce the collective bias. The results of this simple case study of the atomic dimer indicate that this could help in improving the uncertainty estimate.

As a last example, we estimated the posterior distribution of the weights using HMC. The Bayesian model predicts (see Fig. 3) a high uncertainty in the extrapolation regime, for both $r < 0.35$ and $r > 0.45$, and also an increased uncertainty in the region around $r = 0.42$ nm due to lower data density. The Bayesian model compares well to ensembles with the same and with varying architecture, as the bias is further reduced. The uncertainty predicted by the Bayesian NNP resembles most closely our expectation due to the sharp increase of the uncertainty when extrapolating and a moderate increase in uncertainty when interpolating in regions of low training data density.

### B. Al(100) surface

The ensemble of eight DEEPMD models, trained on energies and forces of atoms in and on an Al surface, results in an accurate NNP within and close to the training set distribution. The forces on Al atoms for the configurations of three validation sets ($\mathcal{D}^{\mathrm{Al}}_{300}$, $\mathcal{D}^{\mathrm{Al}}_{600}$, and $\mathcal{D}^{\mathrm{Al}}_{1000}$) are plotted in Fig. 4, showing excellent agreement, evidence that the ensemble is able to capture the interatomic interactions of Al in the bulk and on the surface. We observe that the validation sets sampled at higher temperatures lead to higher forces and higher root-mean-square errors (RMSE) in the prediction, which is expected because the system explores a larger region of phase space at higher temperatures, which translates to a more complex training problem. A histogram of the error for each temperature we studied is shown in Fig. 5, confirming this behavior. The validation set sampled at 1500 K has the highest error. We remind the reader that no configurations sampled at that temperature were used in training, and we mark these as out-of-distribution (ood) samples.

One question of interest is how to detect these ood samples, or—more generally—high-error configurations. The relationship between predicted uncertainty and true error is shown for three validation sets ($\mathcal{D}^{\mathrm{Al}}_{1500}$, $\mathcal{D}^{\mathrm{Al}}_{1000}$, and $\mathcal{D}^{\mathrm{Al}}_{300}$) in Fig. 6.

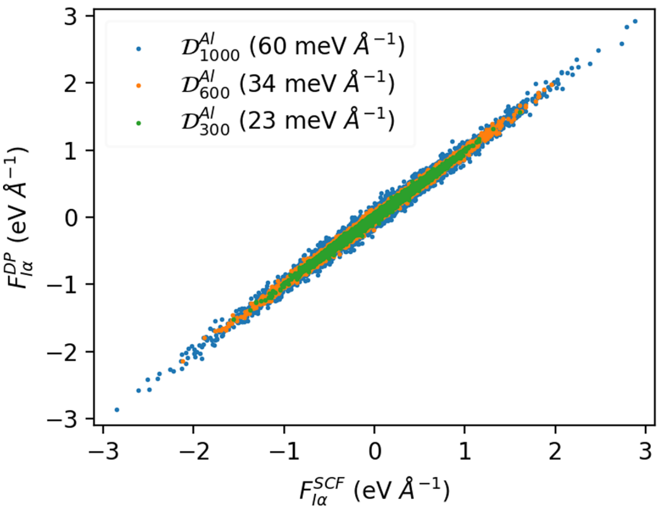

FIG. 4. The prediction of a force component ($y$ axis) against the DFT-calculated force ($x$ axis) for the validations sets $\mathcal{D}^{\mathrm{Al}}_{300}$, $\mathcal{D}^{\mathrm{Al}}_{600}$, and $\mathcal{D}^{\mathrm{Al}}_{1000}$ in blue, orange, and green, respectively. The RMSE of the forces is given in brackets in the legend.

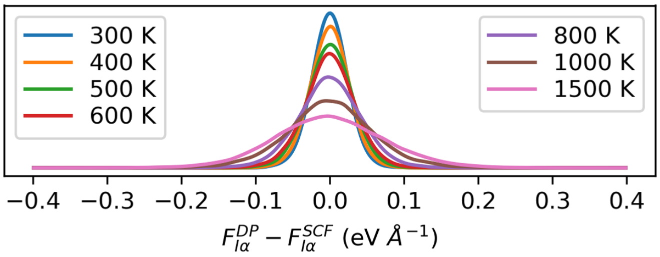

FIG. 5. Histograms of the difference of the predicted and calculated force components reveal an approximately Gaussian-distributed error with increasing deviation with higher temperature.

We observe that the error $\epsilon_{I\alpha}$ is generally about 1 order of magnitude higher than the predicted uncertainty $\sigma_{I\alpha}$.

No perfect correlation between $\epsilon_{I\alpha}$ and $\sigma_{I\alpha}$ is expected, and such a correlation is not a necessary criterion to select snapshots with unacceptable error. A more pragmatic criterion is that a positive constant $c$ can be found such that $\epsilon_{I\alpha} < c \cdot \sigma_{I\alpha}$ for all (or a significant fraction of) $I\alpha$. Such a constant or slope, for example, is visible in Fig. 2(e) of Ref. [68], where the error estimate is given by a Gaussian process regression model. We draw such a line as a guide to the eye in Fig. 6, choosing the line of smallest slope above the data points (and going through the origin). The slope of this line is 62, which means that in order to achieve a very high degree of certainty that the error is below a given threshold $\epsilon_{\max}$, the uncertainty threshold $\sigma_{\max}$ has to be almost 2 orders of magnitude above $\epsilon_{\max}$. We also draw the line that lies above 90% of the data points, resulting in a slope of 14. This shows that even if one is willing to accept a significant amount of high-error configurations, the uncertainty threshold nevertheless lies an order of magnitude above the error threshold. A reliability diagram is shown in the top panel of Fig. 6, where we bin the data points by their uncertainty and plot the mean error for every bin. The mean error increases when the uncertainty is higher, but there are strong signs of miscalibration. A well-calibrated model would result in a reliability diagram close to an identity function.

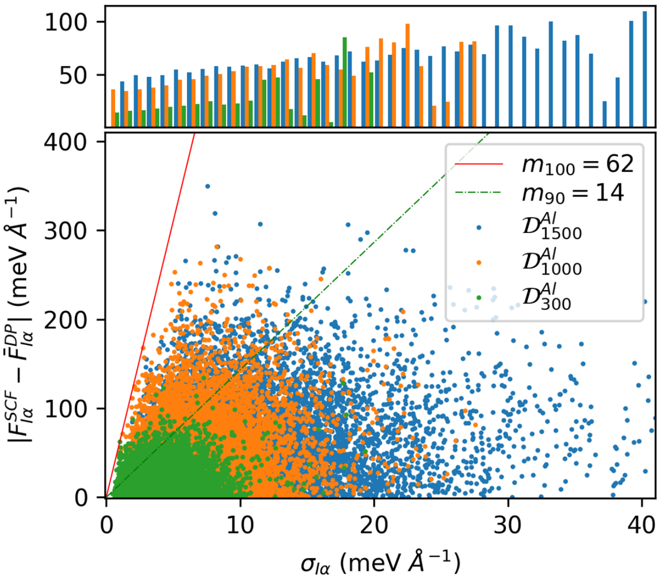

FIG. 6. The ensemble uncertainty $\sigma_{I\alpha}$ [calculated via Eq. (5)] is plotted against the error $\epsilon_{I\alpha}$, calculated as the squared difference between the predicted value and the label, for three validation set $\mathcal{D}^{\mathrm{Al}}_{1500}$, $\mathcal{D}^{\mathrm{Al}}_{1500}$, and $\mathcal{D}^{\mathrm{Al}}_{1500}$ in blue, orange, and green, respectively. The red solid line gives the line of smallest slope the bounds the data points from above. In the top panel, the reliability diagram shows the mean error for all data points with a given uncertainty, with the same color encoding.

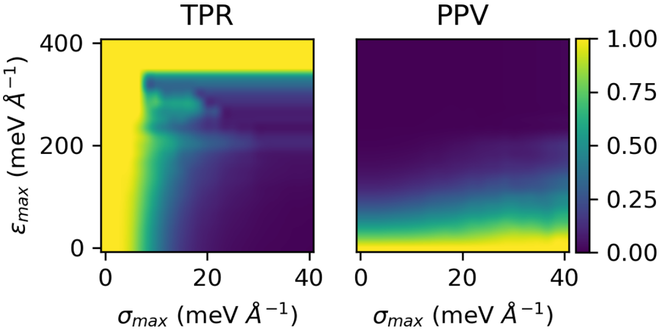

FIG. 7. For the validation set $\mathcal{D}^{\mathrm{Al}}_{1500}$ we plot the recall or TPR as a heat map over different true errors and predicted errors (left panel). On the right, we plot the precision or PPV for the same dataset.

Similar to previous work [39,67], we assume that a maximum true error, $\epsilon_{\max}$, exists above which the properties sampled by the dynamics are no longer trustworthy, requiring a need to recalculate this point during active exploration or marking it for labeling. We classify force components of the validation set $\mathcal{D}^{\mathrm{Al}}_{1500}$ set according to the true error $\epsilon_{I\alpha} > \epsilon_{\max}$ and try to predict this class using $\sigma_{I\alpha} > \sigma_{\max}$. The TPR, introduced in Eq. (10), is shown in the left panel of Fig. 7. For low $\sigma_{\max}$ (below 10 meV $Å^{-1}$), it is possible to guarantee that the true error is bounded by a wide range of $\epsilon_{\max}$. In general, the region of high recall or TPR is at the top left of the heat map, towards high $\epsilon_{\max}$ and low $\sigma_{\max}$. The region of high computational efficiency is given by a high precision or PPV and is concentrated in the bottom right, towards high $\sigma_{\max}$ and low $\epsilon_{\max}$. The only region of high precision or PPV and high recall or TPR is in the bottom left; however, this is the region where $\epsilon_{\max}$ and $\sigma_{\max}$ are so low that almost all data points are true positives (see Fig. S5 of the Supplemental Material [57]). In summary, only by giving very strict criteria on the predicted uncertainty is it guaranteed that the true error is bound, which implies that we can only exclude a negligible amount of the data set as low-error points. In an on-the-fly training scenario, where the low error of force components is important, the number of useful calculations will be very low due to the high number of false positives and the low number of true positives (see Fig. S5 of the Supplemental Material [57]).

In an active-learning scenario, finding all high-error points is not necessary as long as a sufficient number of high-error configurations are discovered to enrich the training set with ood data, allowing us to relax a very strict threshold $\sigma_{\max}$. In such a scenario, it is important to find training points for the next iteration that are more likely to be outside the previous training set than if sampled randomly. In Fig. 8 we plot histograms of the true error in the force components

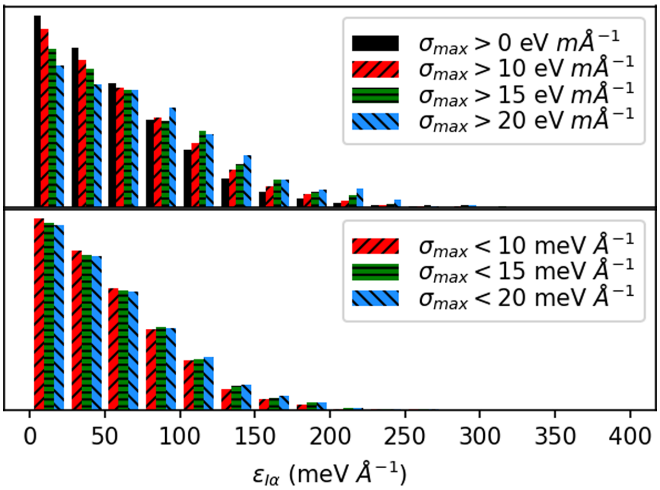

FIG. 8. The true error distribution of snapshots is plotted for all configurations of the validation set $\mathcal{D}^{\mathrm{Al}}_{1500}$ for configurations that have a predicted uncertain above (top panel) or below (bottom panel) a given threshold. The black bars display the true error histogram for all configurations. The histograms being very similar to each other shows that for $\mathcal{D}^{\mathrm{Al}}_{1500}$ it is not possible to single out high-error configurations via a threshold on the model uncertainty.

$\epsilon_{I\alpha}$ when screening for all predictions above $\sigma_{\max}$ ($\sigma_{\max} = 0$ therefore includes all force components). We observe a shift of the histogram towards larger true errors when selecting for components with higher uncertainties; however, this shift is marginal. This confirms our interpretation of the right panel of Fig. 7, that it is not possible to achieve a high PPV except when declaring almost all data points as high-error points *via* a very low threshold $\epsilon_{\max}$.

To summarize, we note that ensemble uncertainties are about 1 order of magnitude lower than the errors. This observation needs to be accounted for when relying on ensemble uncertainties. When specifying the error threshold $\epsilon_{\max}$, one needs to investigate which uncertainty criterion $\sigma_{\max}$ this requires. As this case study shows, it is possible that $\sigma_{\max}$ needs to be significantly larger than $\epsilon_{\max}$. Furthermore, the results of this case study indicate that sampling based on an uncertainty criterion [37] or sampling randomly [18] should result in very similar distributions. Random sampling is, for obvious reasons, easier to implement and computationally more efficient and would, therefore, be preferable.

### C. Water

In Fig. 9 we show the forces predicted by the DEEPMD NNP ensemble against the labels for the three validation sets of bulk water. We note that the training and validation configurations do not originate solely from a molecular-dynamics trajectory, but also from different sampling strategies that lead to significantly higher forces [62] than in the previous case study. The error in the forces increases slightly for snapshots of higher potential energies, and the highest RMSE is found for $\mathcal{D}^{\mathrm{H_2O}}_3$. Nevertheless, the good reproduction of forces in unseen data gives evidence of a very accurate NNP within and close to the training set distribution, gently reducing in accuracy as one moves away from the training examples.

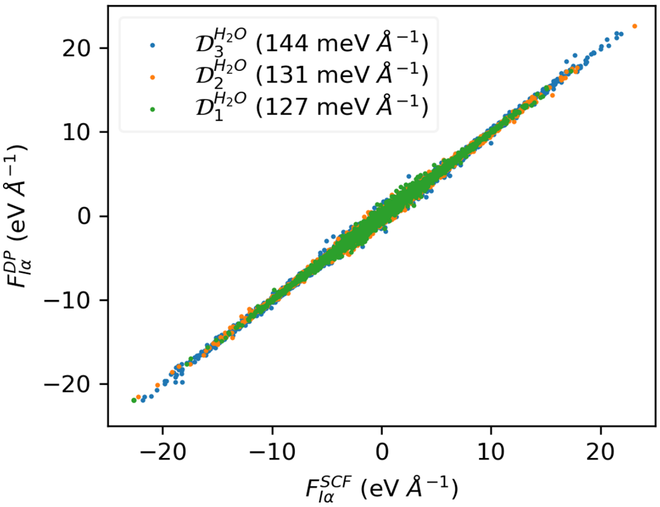

FIG. 9. We plot the DFT forces against the predicted forces of DEEPMD for the validation sets $\mathcal{D}^{\mathrm{H_2O}}_1$, $\mathcal{D}^{\mathrm{H_2O}}_2$, and $\mathcal{D}^{\mathrm{H_2O}}_3$. RMSE is given in brackets inside the legend

The predicted uncertainty against the true error for all validation sets, plotted in Fig. 10, displays a behavior similar to that in the case study of Al (cf. Fig. 6). The first similarity is that the uncertainty predicted by the NNP ensemble is underestimated with respect to the error by the mean prediction, also by approximately 1 order of magnitude. A distinction, however, is that there is no line of finite slope that bounds all data points from above (i.e., it is not possible to bound the error rigorously), due to the presence of configurations that have been predicted with zero uncertainty but display finite true errors. A line that lies above 90% of the validation points has a slope of 7.7, which is a bit lower than in the previous case study. Nevertheless, the conclusion is the same as for Al(100): the uncertainty threshold needs to be an order of magnitude

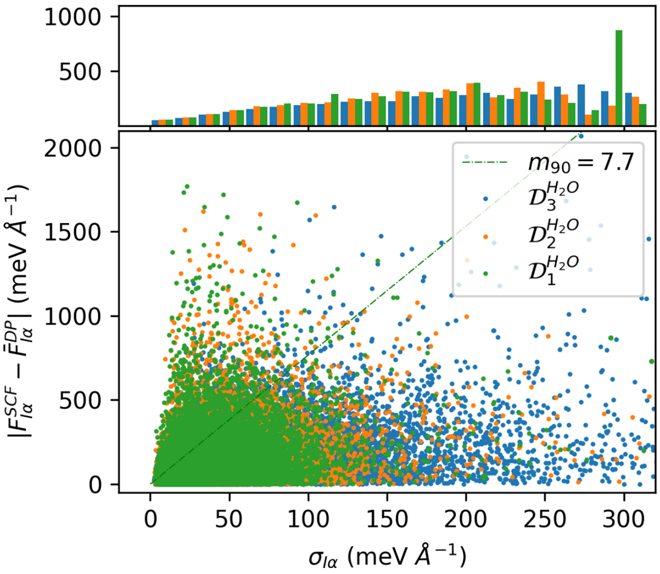

FIG. 10. The predicted uncertainty $\sigma_{I\alpha}$ against the true error in liquid water for the validation sets ($\mathcal{D}^{\mathrm{H_2O}}_1$, $\mathcal{D}^{\mathrm{H_2O}}_2$, and $\mathcal{D}^{\mathrm{H_2O}}_3$), shown as green, orange, and blue dots, respectively.

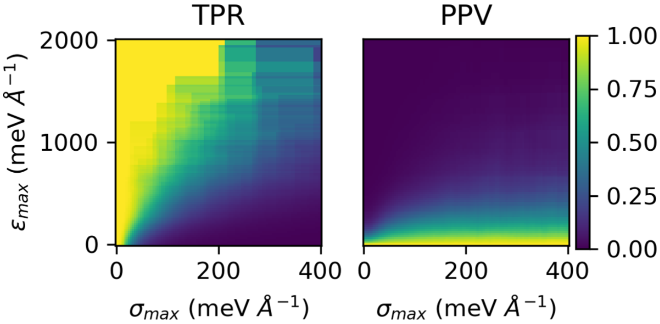

FIG. 11. Recall (TPR) and precision (PPV) for the $\mathcal{D}_3^{H_2O}$ validation set for liquid water.

higher than the error one is willing to accept. Another similarity is the presence of false positives; i.e., force components are predicted relatively accurately despite the high uncertainty associated with that prediction. The reliability diagram in the top panel of Fig. 10 bears closer resemblance to an identity than in the case of Al (cf. Fig. 6). We speculate that this is because the validation set $\mathcal{D}_3^{H_2O}$ of liquid water is described worse by the ensemble than the validation set of Al(100) ($\mathcal{D}_{1500}^{Al}$). The fact that the more accurate a model is, the more likely the model is to be overconfident, has also been observed in other work and is believed to constitute a general trend [69].

As is the case for Al(100), also in liquid water we find regions of high TPR for high-error thresholds $\epsilon_{max}$ and low uncertainty bounds $\sigma_{max}$, meaning that a strict uncertainty bound $\sigma_{max}$ is almost guaranteed to find all examples of high true error $\epsilon_{max}$ (see Fig. 11). However, regions of high TPR are also regions of low PPV, implying that the computational efficiency in that regime is low. As a result, there is no region with a $F_1$ score close to 1 (see Fig. S6 of the Supplemental Material [57]), except for the regime of very strict bounds on uncertainty and error.

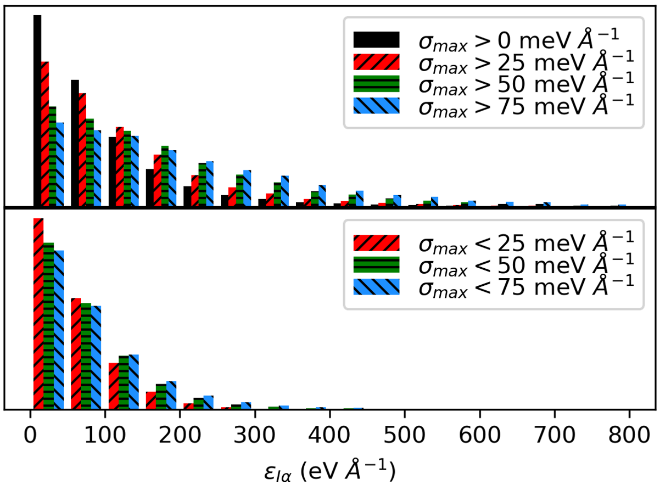

FIG. 12. Histogram of the true error for different subsets of $\mathcal{D}_3^{H_2O}$, based on the predicted model uncertainty. The black bars show the distribution of error distribution of the entire validation set. With increasing model uncertainty threshold, the histogram becomes skewed towards higher-error configurations.

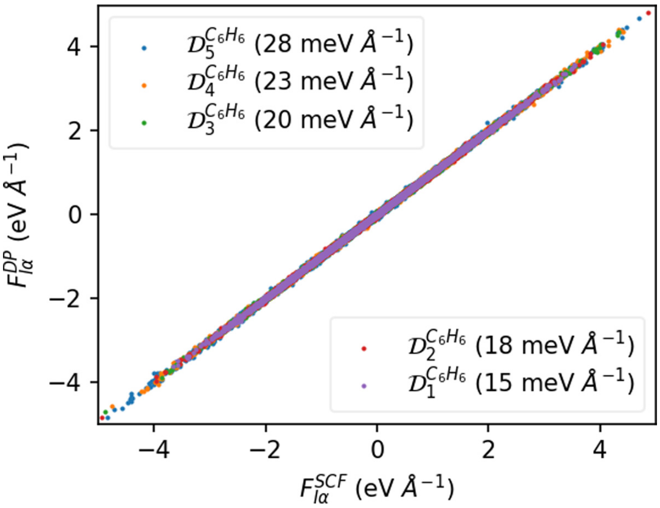

FIG. 13. We plot the CCSD(T) forces against the predicted forces of the NNP ensemble for the validation sets $\mathcal{D}_{1...5}^{C_6H_6}$. RMSE is given in brackets inside the legend.

In Fig. 12, we plot histograms of the true error distributions for subsets of the validation sets with a minimal (top panel) or maximal (bottom panel) uncertainty. Comparing to the case study of Al (cf. Fig. 8), the histograms change more significantly with increased threshold, meaning that one is likelier to select high-error configurations by using an uncertainty criterion (compared to random selection), which confirms the findings above. The histogram is still peaked at low true errors, but the difference in the histograms gives evidence that the uncertainty criterion can improve the selection of configurations for labeling and retraining in an active-learning scenario.

In this case study, sampling based on the ensemble uncertainty is shown to be advantageous, compared to random sampling, as the probability of including high-error configurations is increased. This has to be weighted against the higher computational cost and complexity of the former approach. Our results for water do not indicate that the true error can be bounded by a limit on the ensemble uncertainty, merely that the probability of picking a high-error configuration is higher if selecting configurations with high ensemble uncertainty, compared to random choice. As in the previous case study, to rely on the ensemble uncertainty would require calibration, as the predicted model uncertainties are of a magnitude significantly lower than that of the model errors.

### D. Benzene

As a last case study we show the results of the NEQUIP NNP ensemble trained on the benzene data set. NEQUIP generalizes well for this training set and can be trained with a comparatively small number of training points. The forces estimated by the NNP ensemble trained on the 250 configurations lowest in potential energy are plotted against the CCSD(T) predictions in Fig. 13 for all validation sets. The RMSE increases for validation sets of higher potential energy, as in the previous example (cf. Sec. III C). The lower complexity of the benzene molecule, compared to water or Al, has allowed us to study

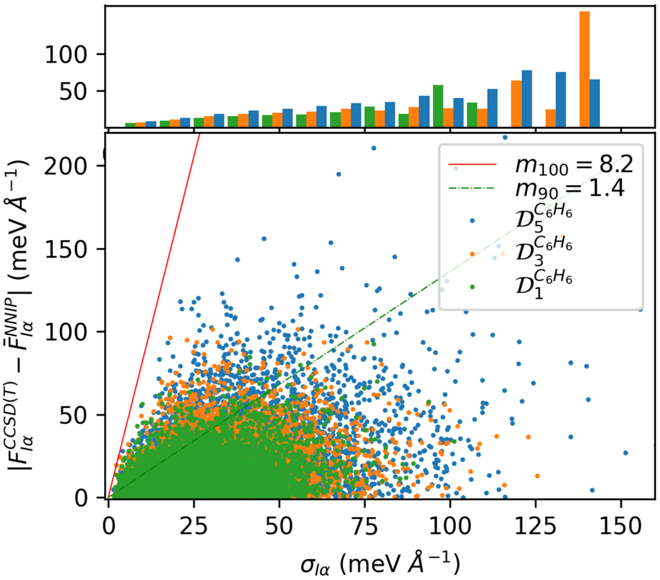

FIG. 14. The predicted uncertainty $\sigma_{I\alpha}$ against the true error of an isolated benzene for the validation sets $\mathcal{D}_1^{C_6H_6}$, $\mathcal{D}_3^{C_6H_6}$, and $\mathcal{D}_5^{C_6H_6}$, shown as green, orange, and blue dots, respectively.

which configurations result in comparatively larger model errors. Based on a simple descriptor of the atomic environment, we find that atoms in the training set whose descriptor falls outside the convex hull of training-set descriptors can have higher errors (see Fig. S10 of the Supplemental Material [57]).

The uncertainty estimates given by the ensemble deviation are plotted against the error in Fig. 14. Here, as for the Al case study (see Sec. III B), it is possible to heuristically bound the error by a line of slope 8.2, which is significantly smaller than a slope of 62 in the case of Al(100), evidence for a lower degree of overconfidence of the ensemble. The line that lies above 90% of the validation points has a slope of 1.4, close to unity. Therefore, unlike the previous examples, the true error and predicted uncertainties are of the same order of magnitude, which means that this system and ensemble are better calibrated. Furthermore, the reliability diagram in the top panel of Fig. 14 resembles the identity function more closely than in the case of Al(100) (cf. Fig. 6), especially if we exclude the right portion of the histogram where the data points are significantly less. We observe that, unlike for liquid water and Al(100), the NEQUIP ensemble is not overconfident: predictions of an uncertainty $\sigma_{I\alpha}$ have a mean error $\epsilon_{I\alpha}$ that is of the same order. Nevertheless, Fig. 15 implies that the same problem persists as for the uncertainty estimates in Al(100) and liquid water; namely, that it is not possible to separate efficiently configurations (or force components) of high uncertainty without also including a large amount of false negatives (configurations that are accurately described, but where the ensemble estimates a high uncertainty). In Fig. 16, we show the histograms of the error distribution for subsets of the validation set $\mathcal{D}_5^{C_6H_6}$, screened by the ensemble uncertainty. As for the case of liquid water, the histogram is shifted towards higher-error configurations when excluding low-uncertainty configurations from the validation set. Nevertheless, the histogram remains with its maximum value at low-error configurations, which means that many false positives are included, and the histograms are not significantly different from a random selection. Also in this case study, there is evidence to suggest that sampling based on the uncertainty [37] can result in an improved training set with a higher likelihood of unseen new data, compared to random sampling.

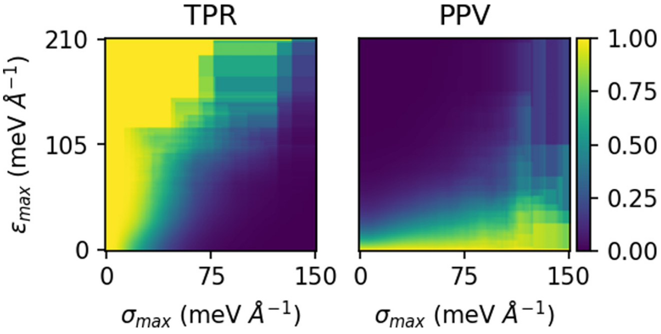

FIG. 15. Recall (TPR) and precision (PPV) for the $\mathcal{D}_5^{C_6H_6}$ validation set.

To conclude, the NEQUIP ensemble for benzene is very accurate without being overconfident of the prediction, which is not the case for DEEPMD in Al(100) or water. The focus of this work, however, is not to compare different implementations or architectures of NNPs, which would be a promising question for future research, but rather to highlight that, for certain implementations and systems, NNP ensembles can be overconfident and that there is high variance in the degree of overconfidence. The proper calibration of NNP ensembles in order to be accurate without being overconfident needs to be studied further. A very interesting perspective on this question is provided by Guo *et al.* [69], who observed that the more accurate a NN model is (due to added parameters and more flexibility), the more it is overconfident.

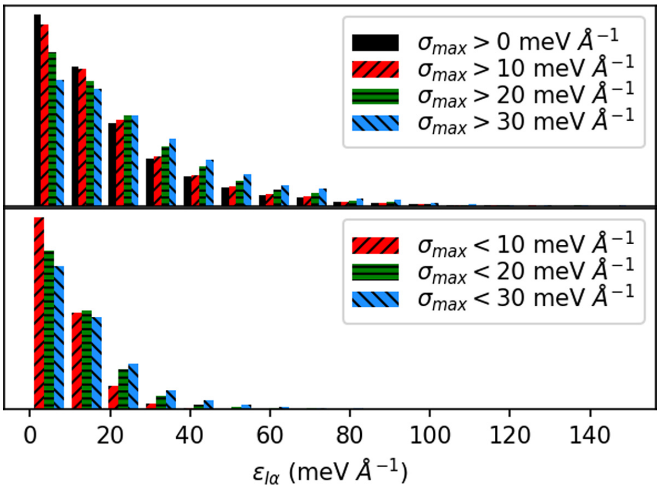

FIG. 16. Histogram of the prediction error for different subsets of $\mathcal{D}_5^{C_6H_6}$, selected based on the ensemble model uncertainty. The black bars show the distribution of error distribution of the entire validation set. With increasing the model uncertainty threshold, the histogram becomes skewed towards higher-error configurations.

## IV. CONCLUSIONS

Zhang *et al.* [67] have shown substantial evidence that selecting data to label based on the uncertainty of NNP ensembles can result in high-quality models and trajectories. On the other hand, Marcolongo *et al.* [18] performed an iterative training where samples for the retraining were chosen randomly, i.e., Boltzmann distributed, from a molecular-dynamics trajectory, also resulting in models of high predictive accuracy. The results of this work reconcile these findings: selecting configurations based on the ensemble uncertainty is in many cases not significantly different from random sampling. By applying low thresholds on the uncertainty, the error of the model can be bound, but this will incur many false positives, resulting in additional computational effort in a learn-on-the-fly or active-learning scenario.

A second finding is that the uncertainty can be significantly underestimated, in the case study of the DEEPMD ensemble trained on Al(100) by an order of magnitude, where an uncertainty threshold of, e.g., $10\,\mathrm{meV}\,\text{Å}^{-1}$ results in errors of $\approx 100\,\mathrm{meV}\,\text{Å}^{-1}$. When employing NNP ensembles, we advise the reader to calibrate the uncertainty threshold on a validation set, as is also advisable in other machine-learning applications [69,70]. Further research is needed on how to properly calibrate NNP ensembles, and an especially interesting avenue is, in our opinion, to explore whether there is an accuracy-confidence trade-off, as observed in other machine-learning applications [69].

Last, our results indicate that the uncertainty estimates can be improved. The NNP ensemble trained on the atomic dimer reveals that ensembles can be biased in a similar manner when the same NN architecture is used, which is also found in similar computational experiments [45]. One method to avoid such common bias is to ensure different architectures of the NNP, for example, by randomizing the activation function for every neuron. A second, more rigorous method, is the implementation of a Bayesian NNP, which can be obtained by sampling the posterior distribution of parameters with HMC and which results in a significantly improved estimate of the uncertainty. Training a Bayesian NNP, however, comes at great complexity and large computational expense, and we defer its implementation and validation to future work.

## ACKNOWLEDGMENTS

This research was supported by the NCCR MARVEL, funded by the Swiss National Science Foundation. We are indebted to Simon Batzner and Alby Musaelian for their generous help with NEQUIP. L.K. thanks Maria R. Cervera for fruitful discussions.

[1] B. J. Alder, S. P. Frankel, and V. A. Lewinson, J. Chem. Phys. **23**, 417 (1955).
[2] A. Rahman, Phys. Rev. **136**, A405 (1964).
[3] F. Ercolessi and J. B. Adams, Europhys. Lett. **26**, 583 (1994).
[4] M. A. González, École thématique Soc. Fr. Neutronique **12**, 169 (2011).
[5] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[6] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[7] R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985).
[8] M. Ceriotti, J. Chem. Phys. **150**, 150901 (2019).
[9] F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, Chem. Rev. **121**, 9759 (2021).
[10] J. Behler, Chem. Rev. **121**, 10037 (2021).
[11] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csnyi, A. V. Shapeev, A. P. Thompson, M. A. Wood, and S. P. Ong, J. Phys. Chem. A **124**, 731 (2020).
[12] X.-G. Li, C. Chen, H. Zheng, Y. Zuo, and S. P. Ong, npj Comput. Mater. **6**, 1 (2020).
[13] J. Qi, S. Banerjee, Y. Zuo, C. Chen, Z. Zhu, M. L. Holekevi Chandrappa, X. Li, and S. P. Ong, Mater. Today Phys. **21**, 100463 (2021).
[14] S. A. Ghasemi, A. Hofstetter, S. Saha, and S. Goedecker, Phys. Rev. B **92**, 045131 (2015).
[15] P. Bleiziffer, K. Schaller, and S. Riniker, J. Chem. Inf. Model. **58**, 579 (2018).
[16] B. Nebgen, N. Lubbers, J. S. Smith, A. E. Sifain, A. Lokhov, O. Isayev, A. E. Roitberg, K. Barros, and S. Tretiak, J. Chem. Theory Comput. **14**, 4687 (2018).
[17] Z. Deng, C. Chen, X.-G. Li, and S. P. Ong, npj Comput. Mater. **5**, 1 (2019).
[18] A. Marcolongo, T. Binninger, F. Zipoli, and T. Laino, ChemSystemsChem **2**, e1900031 (2020).
[19] J. Huang, L. Zhang, H. Wang, J. Zhao, J. Cheng, and W. E, J. Chem. Phys. **154**, 094703 (2021).
[20] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[21] L. Zhang, J. Han, H. Wang, R. Car, and W. E, Phys. Rev. Lett. **120**, 143001 (2018).
[22] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, in *Proceedings of the 28th International Conference on Neural Information Processing Systems—Volume 2*, NIPS'15 (MIT, Cambridge, MA, USA, 2015), pp. 2224–2232.
[23] T. S. Hy, S. Trivedi, H. Pan, B. M. Anderson, and R. Kondor, J. Chem. Phys. **148**, 241745 (2018).
[24] K. Ryczko, K. Mills, I. Luchak, C. Homenick, and I. Tamblyn, Comput. Mater. Sci. **149**, 134 (2018).
[25] T. Xie and J. C. Grossman, Phys. Rev. Lett. **120**, 145301 (2018).
[26] J. Klicpera, J. Groß, and S. Günnemann, arXiv:2003.03123.
[27] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, edited by U. von Luxburg, I. Guyon, S. Bengio, H. Wallach, and R. Fergus Curran Associates Inc. Red Hook, NY, 2017), pp. 992–1002.
[28] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, J. Chem. Phys. **148**, 241722 (2018).
[29] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, arXiv:1802.08219.

[30] J. P. Mailoa, M. Kornbluth, S. Batzner, G. Samsonidze, S. T. Lam, J. Vandermause, C. Ablitt, N. Molinari, and B. Kozinsky, Nat. Mach. Intell. **1**, 471 (2019).

[31] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, arXiv:2101.03164.

[32] L. Zhang, J. Han, H. Wang, W. Saidi, R. Car, and W. E, in *Advances in Neural Information Processing Systems*, Vol. 31, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., 2018).

[33] Z. Li, J. R. Kermode, and A. De Vita, Phys. Rev. Lett. **114**, 096405 (2015).

[34] K. Miwa and H. Ohno, Phys. Rev. Mater. **1**, 053801 (2017).

[35] A. D. Vita and R. Car, MRS Online Proceedings Library **491**, 473 (1997).

[36] G. Csányi, T. Albaret, M. C. Payne, and A. De Vita, Phys. Rev. Lett. **93**, 175503 (2004).

[37] L. Zhang, D.-Y. Lin, H. Wang, R. Car, and W. E, Phys. Rev. Mater. **3**, 023804 (2019).

[38] M. Wen and E. B. Tadmor, npj Comput. Mater. **6**, 1 (2020).

[39] C. Schran, K. Brezina, and O. Marsalek, J. Chem. Phys. **153**, 104105 (2020).

[40] A. Kendall and Y. Gal, arXiv:1703.04977.

[41] N. Tagasovska and D. Lopez-Paz, in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019).

[42] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, Inf. Fusion **76**, 243 (2021).

[43] B. Lakshminarayanan, A. Pritzel, and C. Blundell, in *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).

[44] T. Pearce, F. Leibfried, and A. Brintrup, in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research*, Vol. 108, edited by S. Chiappa and R. Calandra (PMLR, UK, 2020) pp. 234–244.

[45] J. Yao, W. Pan, S. Ghosh, and F. Doshi-Velez, arXiv:1906.09686.

[46] L. Chen, I. Sukuba, M. Probst, and A. Kaiser, RSC Adv. **10**, 4293 (2020).

[47] W. Jeong, D. Yoo, K. Lee, J. Jung, and S. Han, J. Phys. Chem. Lett. **11**, 6090 (2020).

[48] Y. Gal, Ph.D. thesis, University of Cambridge, Cambridge, England, 2016).

[49] F. D'Angelo and V. Fortuin, arXiv:2106.11642.

[50] D. J. C. MacKay, Neural Comput. **4**, 448 (1992).

[51] R. M. Neal, *Bayesian Learning for Neural Networks* (Springer, New York, 1996).

[52] G. M. Martin, D. T. Frazier, and C. P. Robert, arXiv:2004.06425.

[53] R. M. Neal, in *Advances in Neural Information Processing Systems*, Vol. 5, edited by S. Hanson, J. Cowan, and C. Giles (Morgan-Kaufmann, USA, 1993).

[54] R. Chandra, K. Jain, R. V. Deo, and S. Cripps, Neurocomputing **359**, 315 (2019).

[55] R. J. N. Baldock and N. Marzari, arXiv:1904.04154.

[56] J. T. Barron, arXiv:1704.07483.

[57] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.105.015311 for details of the neural-network hyperparameters and additional results.

[58] N. L. Nguyen, F. Baletto, and N. Marzari, Mater. Cloud Arch. **2018.0002/v1**, doi:10.24435/materialscloud:2018.0002/v1 (2018).

[59] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. D. Corso, S. d. Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos *et al.*, J. Phys.: Condens. Matter **21**, 395502 (2009).

[60] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[61] B. Cheng, E. Engel, J. Behler, C. Dellago, and M. Ceriotti, Mater. Cloud Arch. **2018.0020/v1**, doi:10.24435/materialscloud:2018.0020/v1 (2018).

[62] B. Cheng, E. A. Engel, J. Behler, C. Dellago, and M. Ceriotti, Proc. Natl. Acad. Sci. USA **116**, 1110 (2019).

[63] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, Nat. Commun. **9**, 3887 (2018).

[64] F. Wenzel, K. Roth, B. S. Veeling, J. Witkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, arXiv:2002.02405.

[65] R. M. Neal, in Handbook of Markov Chain Monte Carlo, edited by S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng (CRC, Boca Raton, FL, 2011), pp. 113–160.

[66] L. Verlet, Phys. Rev. **159**, 98 (1967).

[67] Y. Zhang, H. Wang, W. Chen, J. Zeng, L. Zhang, H. Wang, and W. E, Comput. Phys. Commun. **253**, 107206 (2020).

[68] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, npj Comput. Mater. **6**, 1 (2020).

[69] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, in *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, edited by D. Precup, and Y. W. Teh (PMLR, 2017), pp. 1321–1330.

[70] N. Seedat and C. Kanan, arXiv:1911.00104.