Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 035009 https://doi.org/10.1088/2632-2153/abe294

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
9 September 2020

REVISED
11 January 2021

ACCEPTED FOR PUBLICATION
2 February 2021

PUBLISHED
12 May 2021

PAPER

Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design

Viktor Zaverkin and Johannes Kästner

Institute for Theoretical Chemistry, University of Stuttgart, Pfaffenwaldring 55, 70569 Stuttgart, Germany

E-mail: kaestner@theochem.uni-stuttgart.de

Keywords: molecular machine learning, atomistic neural networks, active learning, optimal experimental design, computational chemistry

Abstract

Machine learning has been proven to have the potential to bridge the gap between the accuracy of *ab initio* methods and the efficiency of empirical force fields. Neural networks are one of the most frequently used approaches to construct high-dimensional potential energy surfaces. Unfortunately, they lack an inherent uncertainty estimation which is necessary for efficient and automated sampling through the chemical and conformational space to find extrapolative configurations. The identification of the latter is needed for the construction of transferable and uniformly accurate potential energy surfaces. In this paper, we propose an active learning approach that uses the estimated model's output variance derived in the framework of the optimal experimental design. This method has several advantages compared to the established active learning approaches, e.g. Query-by-Committee, Monte Carlo dropout, feature and latent distances, in terms of the predictive power and computational efficiency. We have shown that the application of the proposed active learning scheme leads to transferable and uniformly accurate potential energy surfaces constructed using only a small fraction of data points. Additionally, it is possible to define a natural threshold value for the proposed uncertainty metric which offers the possibility to generate highly informative training data on-the-fly.

1. Introduction

Quantum chemistry (QC) aims to describe the physical and chemical properties of atomistic systems using quantum mechanics. Computational chemistry (CC) uses QC approaches to obtain potential energy surfaces (PESs). Most other physical and chemical properties can be derived from the latter. Application of QC methods to even moderately large atomistic systems is computationally very expensive and, therefore, the development of empirical force fields (FFs) became the cornerstone of the modern CC [1–3]. Empirical FFs are highly efficient but suffer from limited transferability [4] and are generally not able to describe bond breaking and bond formation. Thus, there is a great demand for efficient and accurate PES models.

The recent development of machine learning (ML) methods changes the way of modeling molecular and material systems [5]. Being able to learn efficiently complex and highly non-linear functional relationships ML methods give the promise to bridge the gap between the computational efficiency of FFs and the accuracy of QC. In this paper, we use the ML approach recently developed in our group, which is referred to as Gaussian moment neural networks (GM-NNs) [6].

Employing ML algorithms, it is possible now to parametrize PESs using *ab initio* data to obtain models that can predict energies, atomic forces and Hessians with the *ab initio* accuracy and efficiency of FFs. An important issue appears; trained on QC data there is no guarantee that the parametrized model will properly predict properties of configurations far from the training data set. The generation of appropriate training data appears to be an especially challenging task if one takes into account the dimensionality of the chemical and conformational spaces [7]. Additionally, data sets built based on human intuition tend to be clustered,

sparse, and incomplete. They contain thousands to millions of data points each of them required the calculation of *ab initio* energies and forces. The latter can prohibit the application of ML methods due to the high computational cost.

This problem can be resolved by allowing the ML models to detect the most informative structures and perform the *ab initio* calculations only for them. This can be done on-the-fly, selecting extrapolative structures when running, e.g. the molecular dynamics (MD) simulation, or offline on the fixed data sets improving the generalization and the transferability of the potential. Both possibilities are related to active learning [8], an area of supervised learning whose aim is to learn general-purpose models with a minimal number of training data. The key quantity needed to perform active learning is the query strategy, i.e. an algorithmic criterion for deciding whether a given configuration has to be included in the training set.

A general overview of AL approaches can be found in [8]. In the context of interatomic potentials, a very natural query strategy can be defined for Gaussian process (GP) models using their inherent Bayesian predictive variance. Recently, this approach was successfully applied to model PESs of single- and multi-element systems on-the-fly [9] as well as to construct reactive PESs for H_3 and two prototypical reactive systems [10]. The on-the-fly training of machine-learned force fields was first proposed in [11], while the model error was evaluated employing *ab initio* calculations due to the poor correlation between the internal error of their GP model and the true model error [12]. Besides uncertainty-driven AL algorithms, genetic algorithms were applied for the optimization of training data sets [13] as well as a method based on selecting small building blocks, AMONs, from a dictionary to generate training instances on-the-fly has been recently proposed [14].

In this paper, we focus on methods that can be applied to neural networks. Query-by-Committee (QBC) is one of the most frequently used AL approaches in the literature [15–18]. It estimates the uncertainty of NNs using an ensemble of NN models. While widely employed in the chemistry community, training an ensemble of models increases the computational effort to the number of models used. Another approach to obtain the uncertainty of NNs is the Monte Carlo dropout approach [19, 20]. The cost is reduced to running the model multiple times rather than of the training of an ensemble. Finally, the uncertainty metric can be constructed by measuring the distances in the feature [18, 21, 22] and the latent spaces [20]. This can be prohibited due to the size of the system, the dimension of the feature space, and the size of the NN.

In this work, we propose another AL approach for atomistic NNs which uses the expected change in the model's output variance obtained in the framework of optimal experimental design (OED) [23–25]. To the best of our knowledge, a different OED framework was used previously for the linear regression problems [26, 27], and no application to atomistic NNs was proposed.

In the proposed AL scheme, the model's output variance is calculated using the Fisher information matrix computed using only the weights of the output layer. This can be done since the successive layers act to filter the redundant information present in the previous layers increasing the informativeness of the parameters of the last layer. The proposed AL approach can be applied to select new query points according to the model's output variance in both energies and forces.

The advantages of this approach are that (a) it introduces no overhead into model training or evaluation, (b) it can be applied to both simple and complex NN architectures that have been used for chemical property prediction, (c) it naturally ignores the redundant information present in the input layer and selects new query points based only on the model parameters, (d) it is possible to define an efficient algorithm which can query a large amount of data within few minutes.

The paper has the following structure: first, we shortly introduce the GM-NN model, derive the estimated output variance of NNs, and propose three different query strategies for atomistic NNs. Then, in section 3, we apply our active learning method to the ethanol [6], the QM9 [28, 29], and the N-ASW [7] data sets and compare the results to those obtained by the random selection strategy. We show that the expected change in the estimated output variance is correlated with the generalization error, and discuss the possibility of applying the proposed approach on-the-fly. The concluding remarks are given in section 4.

2. Method

In this work, we consider the problem of learning an input-output mapping $\mathcal{X} \to \mathcal{Y}$ from a set of N_{train} training samples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}$, with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ [23]. In the case of molecular machine learning x_i are typically the atomic coordinates, i.e. $\mathcal{X} \subset \mathbb{R}^{N_{\text{at}} \times 3}$ with N_{at} being the number of atoms, and y_i are molecular physicochemical properties. Here we consider y_i being the scalar total energy of the system, i.e. $\mathcal{Y} \subset \mathbb{R}$, or its atomic forces, i.e. $\mathcal{Y} \subset \mathbb{R}^{N_{\text{at}} \times 3}$.

We denote a general parametrized learner as $f(\mathbf{w}, \cdot)$. Its output can be written as $\hat{y}_i = f(\mathbf{w}, x_i)$. The learner is trained by adjusting parameters $\mathbf{w}$ so that the mean squared loss,

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(\hat{y}_i - y_i\right)^2, \tag{1}$$

is minimized. Converged training procedure results in the best set of parameters, $\hat{\mathbf{w}}$, used subsequently for predictions on the test data or during a real-time simulation.

For this work, we use the GM-NN [6] as the parametrized learner with corresponding weight matrices and bias vectors $\mathbf{w} = \{\boldsymbol{W}_k, \boldsymbol{b}_k\}$. A brief overview of the architecture, molecular descriptor, and training procedure of the machine learning (ML) model is given in section 2.1. For more details about the employed model, see elsewhere [6].

The main focus of this work is the setting in which it is allowed to the parametrized learner, $f(\hat{\mathbf{w}}, \cdot)$, to select a new training input x^* from a set of candidate inputs, which we call the pool:

$$\mathcal{P} = \{x_i\}_{i=1}^{N_{\text{pool}}} \subset \mathbb{R}^{N_{\text{at}} \times 3}, \tag{2}$$

with N_{pool} unlabeled instances. Labeling, here the calculation of *ab initio* energies and forces, is performed only on the selected instances since this process is assumed to be computationally expensive.

Given the above conditions, the main issue remains the selection of new training samples without labeling, which would minimize the generalization error of the model. For this purpose, we propose an active learning scheme that selects new training instances according to the expected change in the estimated output variance of the learner. The latter is derived by employing techniques from the field of optimal experimental design (OED) [25]. This work is based on the applications of OED to feed-forward NNs dated from the end of the 20th century [23, 24] and can be referred to as variance reduction query strategy [8].

Section 2.2 briefly reviews the derivation of the expected change in the estimated variance of NNs when adding a new training instance. Section 2.3 presents query strategies used for the active learning of atomistic NNs.

2.1. GM-NN model

As was mentioned before, we have selected the GM-NN approach [6], which uses feed-forward NNs to represent the high-dimensional potential energy surface (PES), as the parametrized learner. In this approach, a single potential energy E of a molecular or solid-state structure is written as a sum of 'atomic' energy contributions:

$$\hat{E} = \sum_i \hat{E}_i \left(\{\mathbf{R}_{ij}, Z_i, Z_j\}_{j \neq i}\right). \tag{3}$$

These $\hat{E}_i$ depend on the local environment of the atom i within a predefined cutoff sphere of the radius R_c. In the above equation index j runs only over all neighbors within R_c. The choice of R_c depends strongly on the studied system and, therefore, will be specified separately for each data set in section 3.

The description of the local environment in the GM-NN model is given through a set of novel symmetry-preserving local atomic descriptors, the Gaussian moments (GM). In addition to the geometric information, GMs include information about the atomic species of both the central and neighbor atoms. Therefore, for all 'atomic' energy contributions, only a single NN has to be trained, in contrast to using an individual NN for each species as frequently required in the literature. The computational cost and memory usage of the GM-NN model scale linearly with the system size because atomic neighbor lists are employed. Throughout this work, we use a shallow neural network with two hidden layers consisting of 256 and 128 nodes each (abbreviated GM-sNN in [6]).

To train the GM-sNN model the loss function:

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} w_{\text{E}} \left\|\Delta E_i\right\|^2 + \frac{w_{\text{F}}}{3N_{\text{at}}} \sum_{j=1}^{N_{\text{at}}} \left\|\Delta \mathbf{F}_{ij}\right\|^2, \tag{4}$$

is minimized. Here, N_{at} is the number of atoms in the respective structure, ΔE_i and $\Delta \mathbf{F}_{ij}$ are the differences between the GM-sNN prediction and the reference data for energies and forces, respectively. The parameters w_{E} and w_{F} were set to 1 au and 100 au Å^2, respectively. Every GM-sNN model was trained using the AMSGrad optimizer [30] with 32 molecules per mini-batch. The learning rate was set to 10^{-3} and kept constant throughout the whole training procedure. All models used in section 3 were implemented in the Tensorflow framework [31] and were trained for 5000 training epochs on an NVIDIA Tesla V100-SXM2-32GB GPU.

2.2. Variance estimation for NNs

The purpose of this section is to derive an estimator for the output variance of NNs and the respective change in the variance when a new data point is added to the training set. The latter can be used to indirectly minimize the generalization error of NNs. This holds since the learner's expected future error can be decomposed into three terms [32]: (a) the *noise* of the data introduced by, e.g. the *ab initio* method, which is independent of the model and the training set; (b) the *bias* of the model, i.e. the error introduced by the model class itself; (c) the *variance* of the model. Thus, minimizing the variance of the model is guaranteed to minimize the future generalization error of the model [8].

Following derivations in references [23, 24] the estimated output variance of the NN at the training point x_i can be written as

$$\sigma^2_{\hat{y}}(x_i) \approx \mathcal{L}\left(\frac{\partial \hat{y}_i}{\partial \mathbf{w}}\right)^T \left(\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^2}\right)^{-1} \left(\frac{\partial \hat{y}_i}{\partial \mathbf{w}}\right), \tag{5}$$

where $\mathcal{L}$ is the mean squared loss of the NN given in equation (1) or equation (4), the network sensitivity is defined by $\mathbf{g}(x_i) = \partial \hat{y}_i / \partial \mathbf{w}$, and the Fisher information matrix, $\mathbf{A}$, is defined as

$$\mathbf{A} = \frac{1}{\mathcal{L}} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^2}. \tag{6}$$

The inverse of the Fischer information matrix can be referred to as the parameter covariance matrix.

By using the chain rule one can easily obtain the Fisher information matrix element:

$$A_{ab} = \frac{2}{\mathcal{L} N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \frac{\partial \hat{y}_i}{\partial w_a} \frac{\partial \hat{y}_i}{\partial w_b} + (\hat{y}_i - y_i) \frac{\partial^2 \hat{y}_i}{\partial w_a \partial w_b}, \tag{7}$$

where the prefactor can be seen as the noise level. For the final expression given in equation (14) the noise parameter results in a multiplicative prefactor and, therefore, does not have any impact on the method performance. To be consistent with [23] we use the training residual $\mathcal{L}$ as a noise estimate. Assuming that the trained model is already close to the optimal minimum, i.e. the prediction for x_i is fairly good, the Fisher information matrix can be approximated as

$$\mathbf{A} \approx \frac{1}{\mathcal{L}} \sum_{i=1}^{N_{\text{train}}} \mathbf{g}(x_i) \mathbf{g}^T(x_i). \tag{8}$$

We have mentioned before that the generalization error of the model is correlated with its output variance. The variance reduction query becomes

$$x^* = \underset{x^* \in \mathcal{P}}{\operatorname{argmin}} \langle \sigma^2_{\hat{y}}(x_i, x^*) \rangle_{x_i \in \mathcal{D}}, \tag{9}$$

where the expression $\langle \sigma^2_{\hat{y}}(x_i, x^*) \rangle_{x_i \in \mathcal{D}}$ is the estimated mean output variance across the input distribution, i.e. the learner's output variance averaged over the training samples after the model has been retrained on the instance x^* and its corresponding label. Note that in the following we write instead of $\langle \cdot \rangle_{x_i \in \mathcal{D}}$ only $\langle \cdot \rangle_{\mathcal{D}}$ to make the notation somewhat shorter.

Different approaches can be used to find an optimal instance x^* without re-training the model. For example, in references [26, 27] the so-called D-optimality approach was employed. In this work, we follow the approach proposed in [23].

After adding a new training instance x^* the Fisher information matrix can be approximated as

$$\mathbf{A}^* \approx \mathbf{A} + \frac{1}{\mathcal{L}} \mathbf{g}(x^*) \mathbf{g}^T(x^*). \tag{10}$$

Its inverse can be easily calculated by the Woodbury matrix identity:

$$(\mathbf{A}^*)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1} \mathbf{g}(x^*) \mathbf{g}^T(x^*) \mathbf{A}^{-1}}{\mathcal{L} + \mathbf{g}^T(x^*) \mathbf{A}^{-1} \mathbf{g}(x^*)}. \tag{11}$$

The output variance of the model, after adding the data point x^*, can be estimated at the reference point x_i without re-training the model as

$$\begin{aligned} \sigma^2_{\hat{y}}(x_i, x^*) &= \mathbf{g}^T(x_i) \left(\mathbf{A}^*\right)^{-1} \mathbf{g}(x_i) \\ &= \mathbf{g}^T(x_i) \mathbf{A}^{-1} \mathbf{g}(x_i) - \frac{\left[\mathbf{g}^T(x_i) \mathbf{A}^{-1} \mathbf{g}(x^*)\right]^2}{\mathcal{L} + \mathbf{g}^T(x^*) \mathbf{A}^{-1} \mathbf{g}(x^*)}, \end{aligned} \tag{12}$$

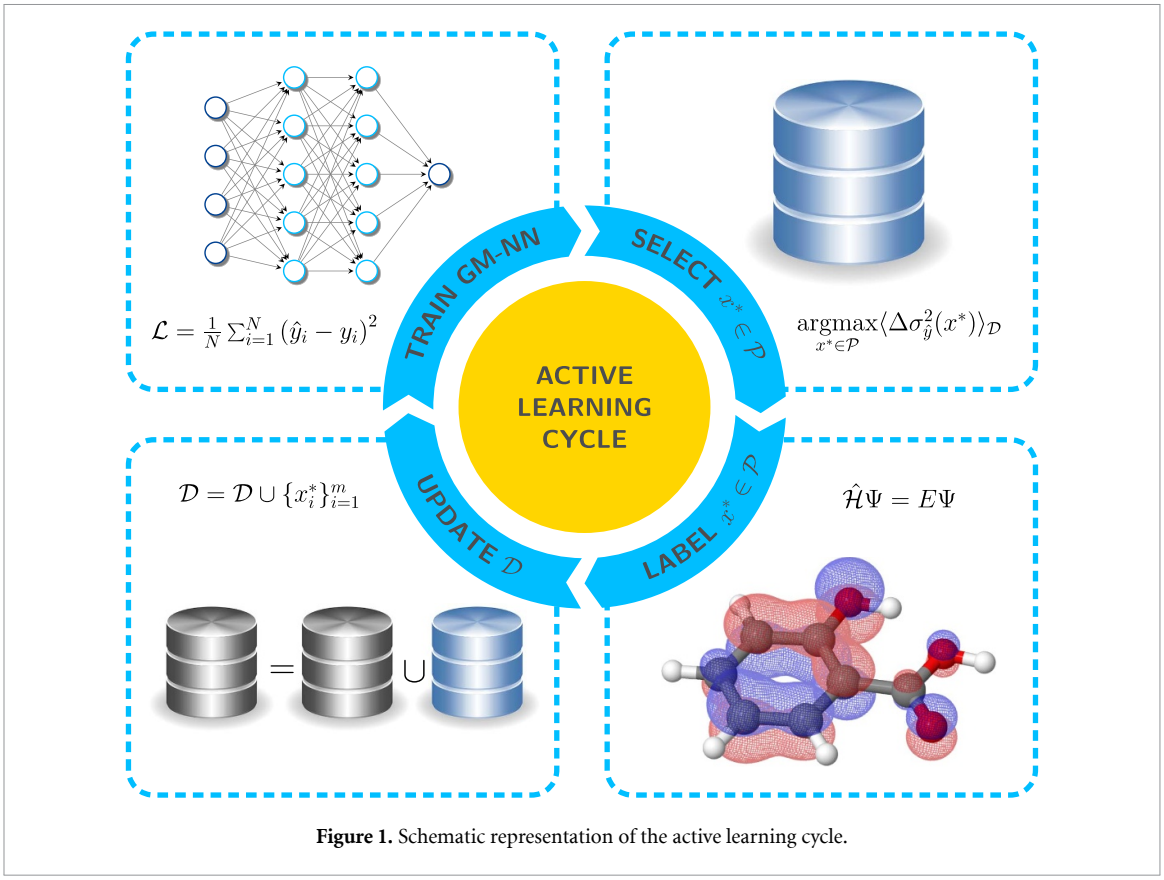

Figure 1. Schematic representation of the active learning cycle.

where the first term, $\mathbf{g}^T(x_i)\mathbf{A}^{-1}\mathbf{g}(x_i) = \sigma^2_{\hat{y}}(x_i)$, is the original output variance of the model. The second term is the expected change in the model's output variance after querying a new data point x^*. Since our interest is the average variance change one has to calculate the average for each instance in the pool. This is inefficient if the pool contains a large number of data. To make this step computationally efficient one can approximate the respective expression by

$$\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}} \approx \frac{\mathbf{g}^T(x^*)\mathbf{A}^{-1}\langle \mathbf{g}(x_i)\mathbf{g}^T(x_i)\rangle_{\mathcal{D}}\mathbf{A}^{-1}\mathbf{g}(x^*)}{\mathcal{L} + \mathbf{g}^T(x^*)\mathbf{A}^{-1}\mathbf{g}(x^*)}. \tag{13}$$

In the above expression, one calculates the average over training samples once when calculating $\langle \mathbf{g}(x_i)\mathbf{g}^T(x_i)\rangle_{\mathcal{D}}$ and can re-use it for all instances in the pool. Given this expression, one solves the problem defined in equation (9), i.e. one can define an instance $x^* \in \mathcal{P}$ which minimizes the model's output variance without either labeling the data or re-training the model. We want to remind the reader that labeling, calculation of *ab initio* energies and atomic forces, is assumed to be computationally expensive.

The expression in equation (13) can be simplified even further using the definition of the Fisher information matrix $\mathbf{A}$ and of the average $\langle \mathbf{g}(x_i)\mathbf{g}^T(x_i)\rangle_{\mathcal{D}}$ and reads

$$\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}} = \frac{\mathcal{L}}{N_{\text{train}}} \cdot \frac{\mathbf{g}^T(x^*)\tilde{\mathbf{A}}^{-1}\mathbf{g}(x^*)}{1 + \mathbf{g}^T(x^*)\tilde{\mathbf{A}}^{-1}\mathbf{g}(x^*)}, \tag{14}$$

where $\tilde{\mathbf{A}} = \mathcal{L}\mathbf{A}$. Note that we neglect the prefactor $\mathcal{L}/N_{\text{train}}$ in the following discussion since it is independent of the new data point x^* and only rescales the expected change in the model's output variance.

2.3. Active learning: atomistic neural networks

An active learning scheme has to be able to select the most informative instances from the unlabeled pool of data. Here, we make use of equation (14) derived in section 2.2 and of the fact that the model's output variance is correlated with the generalization error.

The active learning scenario proposed in this work is schematically shown in figure 1. In the first step, the ML model is initialized, i.e. it is trained on the initial, randomly selected, training data set of size N_{train}. Next, using the expression in equation (14), m structures are selected from the pool. The respective structures are selected such that the expected change in the output variance of the model is maximal:

$$x^* = \underset{x^* \in \mathcal{P}}{\operatorname{argmax}} \langle \Delta\sigma^2_{\hat{y}}(x^*) \rangle_{\mathcal{D}}. \tag{15}$$

The above expression correlates with equation (9). Larger values of the expected change in the output variance imply reduction of the output variance itself, see equation (12). Note that due to the possible correlations of the data in the pool, the active learning algorithm selects a maximum of $m = 0.1 \cdot N_{\text{train}}$ new samples per iteration. The size of the training data set increases after each active learning iteration by m samples, respectively. Finally, after the algorithm has selected m query instances, the respective labels are calculated using, e.g. *ab initio* quantum chemistry (QC) and the training set is updated by these query samples. In this work, we re-train the GM-NN model on the updated training set using re-initialized weights and biases to study the performance of the active learning scheme more thoroughly. However, we should mention that one can start from the pre-trained parameters from the previous iteration which would speed-up the re-training.

The active learning continues until either the maximal size of the training set is reached or the queried instances are not sufficiently informative anymore. We use only the first criterion in section 3, but discuss ways to define the second criterion. Note that the latter is important to perform the so-called learning on-the-fly, where the active learning algorithm queries structures obtained during the simulation. In this work, we use only data sets that are already labeled to test the proposed approach in terms of its applicability to the sampling of configurational and chemical spaces. The maximal size of the training data set depends on the data set. Therefore, the upper limit is defined separately for each data set in section 3.

Before introducing the possible query strategies for atomistic NNs we want to briefly discuss an issue caused by the size of the parameter space of NNs. In general, the parameter space of atomistic NNs is large, which makes the proposed approach, in the first view, intractable on typical computers. For example, for the shallow GM-sNN model with only two hidden layers and 427 invariant molecular descriptors, one obtains more than 142 000 weight parameters. Note that we take weights into account for active learning but no biases because their influence is expected to be negligible. Therefore, one needs to make additional assumptions to make the presented approach applicable to the atomistic NNs.

To tackle the size problem we assume that the weight parameters of the output layer contribute most to the estimation of the model's output variance. One can argue that the preceding layers of the NN contain some amount of redundant information that is filtered when passing through the network. This makes the input of the output layer and, thus, the respective weights more sensitive to the relevant changes in the queried structures. A similar assumption was made for the latent distances in [20]. The performed experiments prove the above premise, see section 3. Using only the weight parameters of the output layer, the Fisher information matrix is only 128×128-dimensional and can be easily inverted.

Now we want to focus on particular query strategies that can be used to select the most informative structures for atomistic NNs. In the following, three different possibilities are presented, based on the energy, force, and the total squared loss.

2.3.1. Query strategy QS_1: energy squared loss

The first query strategy, labeled as QS_1, uses only the energy squared loss:

$$\mathcal{L}_{\hat{E}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left\| E_i - \hat{E}_i \right\|^2. \tag{16}$$

Given the energy loss, one can easily define the network sensitivity, $\mathbf{g}_{\hat{E}}(x_i) = \partial \hat{E}_i / \partial \mathbf{w}$, and the corresponding Fisher information matrix reads

$$\mathbf{A}_{\hat{E}} = \frac{1}{\mathcal{L}_{\hat{E}}} \sum_{i=1}^{N_{\text{train}}} \mathbf{g}_{\hat{E}}(x_i)\, \mathbf{g}_{\hat{E}}(x_i)^T. \tag{17}$$

Configurations are selected from the pool of unlabeled structures using equations (14) and (15) with the respective network sensitivity, Fisher information matrix, and squared loss.

2.3.2. Query strategy QS_2: force squared loss

The second query strategy, labeled as QS_2, uses only the force squared loss:

$$\mathcal{L}_{\hat{F}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \frac{1}{3N_{\text{at}}} \sum_{j=1}^{N_{\text{at}}} \sum_{k=1}^{3} \left\| F_{ijk} - \hat{F}_{ijk} \right\|^2. \tag{18}$$

In this case, the network sensitivity is calculated as the gradient of the atomic force element with respect to the model parameters:

$$\mathbf{g}_{\hat{F}_{jk}}(x_i) = \frac{\partial \hat{F}_{ijk}}{\partial \mathbf{w}}. \tag{19}$$

This implies that one obtains a tensor of rank 3 for the whole molecular structure instead of a vector as in QS_1. This makes the query strategy QS_2 less efficient. However, it can be advantageous if the most informative local atomic environments have to be found.

Using the corresponding network sensitivity and the force squared loss one can write an expression for the Fisher information matrix:

$$\mathbf{A}_{\hat{F}} = \frac{1}{\mathcal{L}_{\hat{F}}} \sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{N_{\text{at}}} \sum_{k=1}^{3} \mathbf{g}_{\hat{F}_{jk}}(x_i)\, \mathbf{g}_{\hat{F}_{jk}}(x_i)^T. \tag{20}$$

Similar to QS_1, configurations are selected from the pool of unlabeled structures using equations (14) and (15) with the respective network sensitivity, Fisher information matrix, and squared loss.

In contrary to QS_1 one obtains a matrix, $(\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}})_{ij}$, of size $N_{\text{at}} \times 3$ for each structure x^* in the pool. We propose to use the mean of that matrix as the final value employed to select the most informative structures, i.e.

$$\langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}} = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} (\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}})_{ij}. \tag{21}$$

Alternatively, one could use the maximal value of the matrix, $(\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}})_{ij}$, selecting structures according to the most informative local environments. For the data sets used in section 3, we have found that using the mean gives the best correlation between the calculated metric, $\langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}}$, and the actual absolute error in predicted force. Therefore, we employed only this approach in section 3.

2.3.3. Query strategy QS_3: total squared loss

It is also possible to calculate the expected change in the estimated output variance of the model using the total loss function presented in equation (4). The uncertainty of the model evaluated for an instance $x^* \in \mathcal{P}$ can be written as a weighted sum of results obtained in sections 2.3.1 and 2.3.2:

$$\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle^{\text{total}}_{\mathcal{D}} = \langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle^{\text{energy}}_{\mathcal{D}} + \beta \langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle^{\text{force}}_{\mathcal{D}}, \tag{22}$$

where $\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle^{\text{energy}}_{\mathcal{D}}$ is the expected change in the model's output variance obtained using the energy loss and $\langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle^{\text{force}}_{\mathcal{D}}$ is the uncertainty obtained from the force loss. The force contribution is scaled by a factor β, similar to equation (4). Note that in section 3.1 and section 3.3 we have found that this approach is of no practical use for the respective systems. This is due to the high correlation between QS_1 and QS_2 metric, high computational cost of the latter, and only minor improvement in terms of the correlation with the actual error.

3. Results

In this section, we test the proposed active learning (AL) scheme on three different molecular data sets. In section 3.1, we confirm the applicability of our approach to atomistic NNs in practice using the ethanol molecular dynamics (MD) data set [6], which samples the configurational space. In section 3.2, we simulate the chemical space sampling employing the well-established QM9 data set [28, 29]. Lastly, in section 3.3, the OED uncertainty metric is used to construct a transferable and uniformly accurate NN potential using the N-ASW data set [7]. Additionally, we discuss the possibility of using the proposed approach on-the-fly.

All experiments described in this section were run using the GM-NN approach [6], which uses feed-forward NN with two hidden layers containing [256, 128] nodes each. For representing molecular structures the selected approach uses 427 rotationally invariant scalars referred to as GMs. The only hyper-parameter needed to be defined is the cutoff radius. It is set up for each experiment separately.

3.1. Ethanol: molecular dynamics data

We start by applying the proposed AL approach to the ethanol data set [6] to confirm its general applicability to molecular systems. The ethanol data set contains Cartesian coordinates, total energies, and atomic forces of 5000 conformations obtained from an *ab initio* MD at 1000 K. Energies and atomic forces were calculated at the PBE-D3(BJ)/6-31G* [33–36] level of theory. In this section, we set the cutoff radius to 4 Å, i.e. the whole molecule is within the cutoff sphere.

Before discussing the obtained results we want to define all hyper-parameters of AL scheme used in this section. Each AL cycle is initialized drawing randomly 100 and 200 structures from the data set. The GM-sNN is trained on 100 structures, the other 200 structures were used for early stopping [37]. Then, the parameters of the trained model are used to calculate the OED metric defined in equation (14) for all conformations in the pool. Note that the pool comprises the structures remaining after the selection of the training samples and includes structures used for validation. $0.1 \cdot N_{\text{train}}$ structures with the maximal OED metric are selected and added to the training set. Finally, 200 new conformations are drawn randomly from the pool for validation and the model is re-trained. In total, we performed 32 active learning iterations including the initialization, which results in a maximal training set size of 1863. All structures which were not employed during training (e.g. for the last AL iteration 2937 structures) were used to test the performance of the model.

It must be emphasized that the gains predicted in the OED framework are expected gains and depend on several approximations discussed in section 2.2. Therefore, we want to confirm that the OED uncertainty metric correlates with the absolute error in energies and atomic forces. Figure 2 shows the correlation of the QS_1 (top left) and QS_2 (top right) metric with the actual force errors. Each dot in figure 2 corresponds to one structure in the pool. Note that forces alone determine the dynamics of a chemical system. Therefore, we want to confirm that the AL approach can select those structures which would improve force prediction on extrapolative conformations. All metrics presented in figure 2 are calculated according to equation (14), if not stated otherwise, and normalized to $[0, 1]$ for comparison.

From figure 2 one can notice that the correlation between the actual error and the estimated uncertainty of the model is not ideal. The imperfect correlation between uncertainties and actual errors can originate from inductive biases of the model and from the fact that a large uncertainty does not necessarily imply a large error. To estimate the influence of the latter contribution, we compute the correlation between the uncertainty and random errors sampled from the posterior predictive distribution as

$$\Delta F = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right| , \tag{23}$$

with $\mathbf{F_i} \sim \mathcal{N}\left(\hat{\mathbf{F}}_i, \langle \Delta\sigma^2_{\hat{y}}(x^*) \rangle_{\mathcal{D}}\right)$. Using this method, we find a correlation of 0.86 for QS_2 if the posterior predictive distribution was accurate. The fact that the real correlation is lower can be attributed to the inductive biases of the ML method.

Figure 2 shows a slightly better correlation of the QS_2 metric with the actual error in the forces. The correlation coefficients obtained for QS_1 and QS_2 are 0.53 and 0.60, respectively. Note that for the QS_1 uncertainty we used only the numerator of equation (14) for the plot since we have found that the whole expression shows a non-linear relationship with the actual error. This can be done as long as the expression in equation (14) and its numerator are monotonically increasing functions, which is the case for the current study. Regarding the computational cost, the QS_2 metric needs 80 times more CPU time (6 s for the QS_1 metric on a single Intel Xeon CPU E5-2640 4) and 20 times more memory (140 MB for the QS_1 metric) to be evaluated for all structures in the pool. Both, the CPU time and the memory usage, depend only slightly on the active learning iteration.

Additionally, we have studied the correlation between QS_1 and QS_2 uncertainties. In figure 2 (bottom left) one can see that the uncertainty metrics are strongly correlated with a correlation coefficient of 0.75. The combination of QS_1 and QS_2 uncertainties comprising the QS_3 query strategy, see section 2.3.3, have not shown any considerable improvement. We could reach only a correlation coefficient of 0.61 with $\beta = 1.5$, see figure 2 (bottom right). For that reason, in the following, we consider only two other query strategies described in section 2.3.1 and section 2.3.2.

Considering the efficiency of the QS_1 metric and the sufficient correlation with the actual force error it seems superior over the QS_2 and QS_3 metric. The QS_2 metric can directly provide information on the most informative local atomic environment. However, because the total energy is decomposed into 'atomic' energies, it is also possible to identify the most informative local environment by using the QS_1 metric.

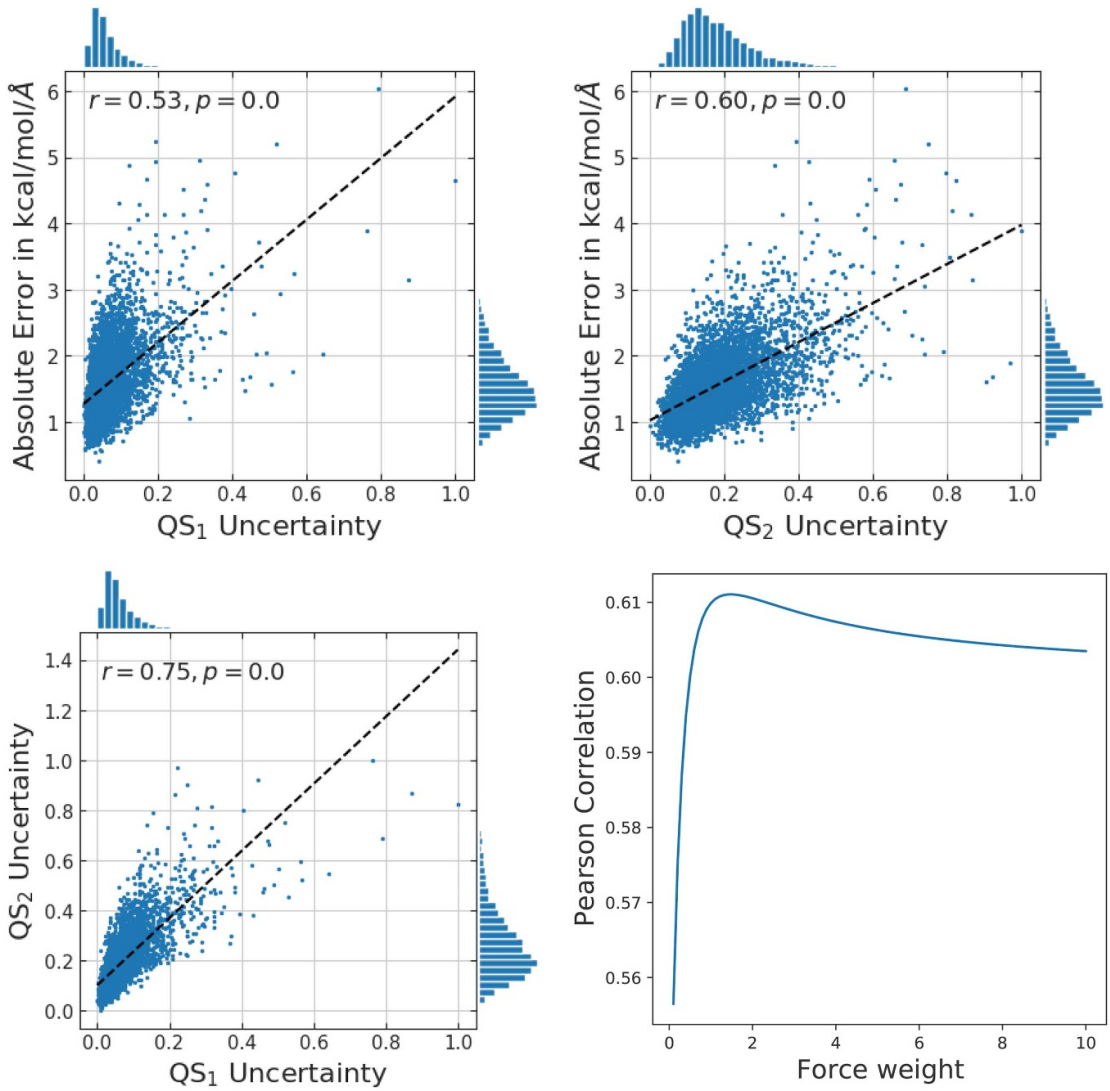

Figure 2. Correlation of the (top left) QS_1 and (top right) QS_2 metric with the absolute force error $\Delta F = \frac{1}{3N_{at}} \sum_{i=1}^{N_{at}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right|$ on the ethanol data set. For both query strategies, the correlation coefficient is shown as an inset along with the best fit line. (Bottom left) Correlation between QS_1 and QS_2 uncertainty metric. The correlation coefficient is shown as an inset along with the best fit line. (Bottom right) The dependence of the correlation coefficient obtained for the QS_3 metric and the absolute force error on the force weight β in equation (22).

The success of the AL algorithms can be measured by comparing to randomly chosen training sets. Figure 3 shows three different error measures obtained for the force predictions of the GM-sNN model trained on actively and randomly selected data. The respective error measures are the mean absolute error (MAE), L_1, the root-mean-squared error (RMSE), L_2, and the maximal error (MAXE), L_∞. All results are averaged over three independent runs. While the MAE and RMSE of the model trained on the structures selected by the AL algorithm are improved only by factors of 1.15 to 1.23, the MAXE is reduced by factors larger than 2.0, compared to the results obtained with randomly selected training data. Strong improvement of the MAXE, i.e. the identification of extrapolative or unusual configurations, shows that our AL scheme leads to the generation of uniformly accurate machine-learned potentials. Note that only about 1150 structures were needed for a maximal error of around 1 kcal mol^{-1} $Å^{-1}$, the accuracy required for molecular simulations.

All models for the ethanol data set were trained on an NVIDIA Tesla V100-SXM2-32GB GPU. The training of 5000 epochs took from 13 min (100 structures) to about 2 h (1863 structures).

3.2. QM9

In this section, we use the QM9 data set [28, 29] to assess the performance of our AL scheme when sampling the chemical space. QM9 is a widely used benchmark for the prediction of several properties of molecules in

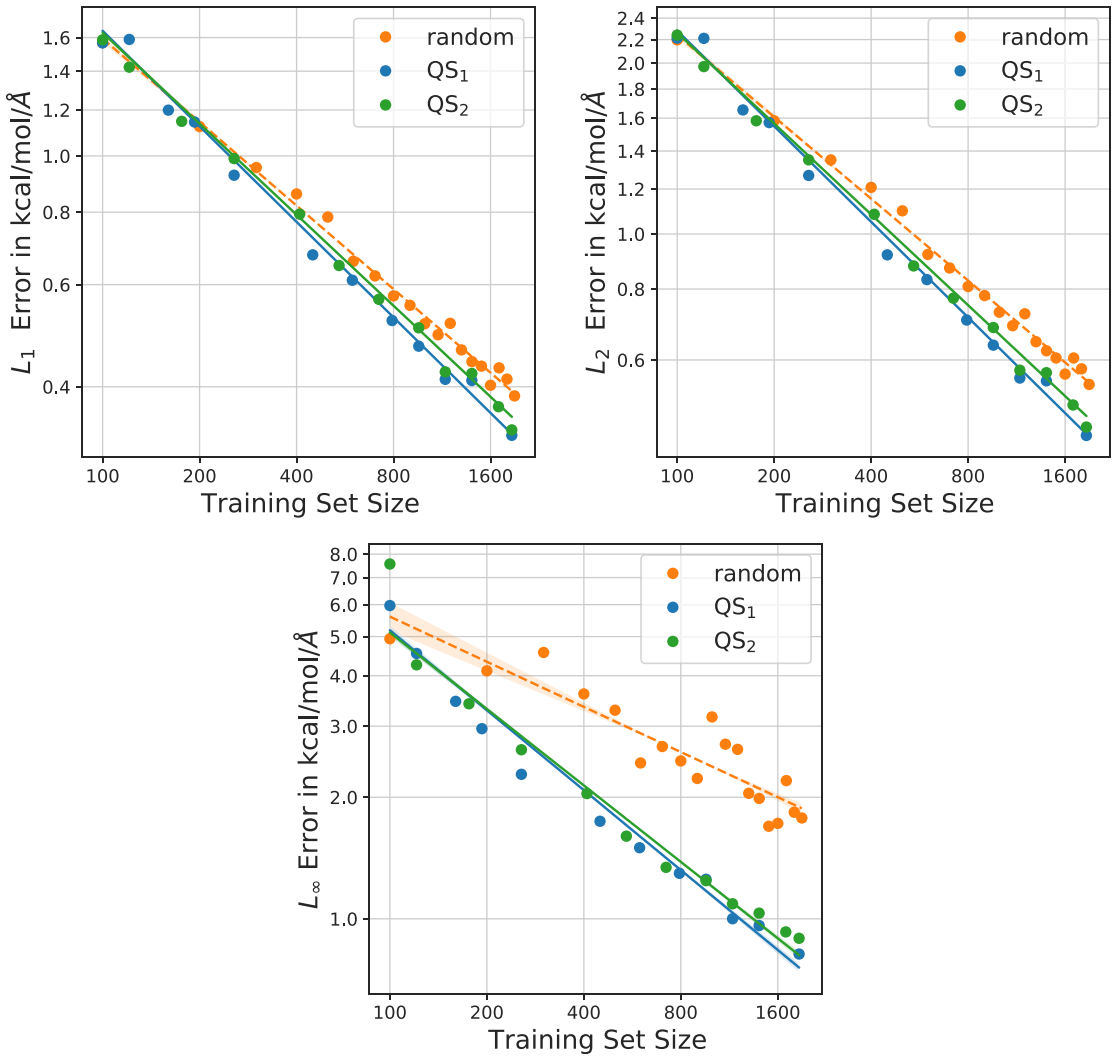

Figure 3. Log–log plot of (top left) the mean absolute, (top right) the root-mean-squared, and (bottom) the maximal error of the predicted forces on the ethanol data set depending on the size of the training set. Structures are selected randomly or using query strategies QS_1 and QS_2. For each learning curve a linear fit is plotted in addition.

equilibrium. Thus, all forces vanish and the model is trained using only the energy squared loss. For AL we use only the query strategy QS_1.

The QM9 data set consists of 133 885 neutral, closed-shell organic molecules with up to 9 heavy atoms (C, O, N, F) and a varying number of hydrogen (H) atoms. The largest structure in the data set contains 29 atoms in total. Since 3054 molecules from the original QM9 data set failed a consistency test [29], we used only the remaining 130 831 structures in the following experiments. Similar to the previous work we used a cutoff radius of 3.0 Å [6].

For initializing AL cycles we selected randomly 5000 samples to train the model and another 2000 structures to validate its performance during the training procedure. The parameters of the converged model were employed to select new training samples from the pool comprising 125 831 structures using equation (14). Note that the structures used for early stopping (validation set) are also added to the pool. In every iteration the AL algorithm selects $0.1 \cdot N_{\text{train}}$ new structures and adds them to the training set. The AL cycle was stopped when the training set size reached a value of 25 261, i.e. after 18 iterations including the initialization.

We want to make an additional remark on the computational cost of the QS_1 strategy. To select 500 samples out of 125 831 structures in the pool we needed about 135 s on the single Intel Xeon CPU E5-2640 4. The memory used is about 850 MB. Both values are almost independent of the AL step.

Similar to the previous section, figure 4 (top left) shows the correlation of the OED metric with the absolute energy error. We have found a correlation coefficient of 0.44 for the QS_1 uncertainty metric. Note that in the figure we used the square root of the expression in equation (14) to be consistent with the results

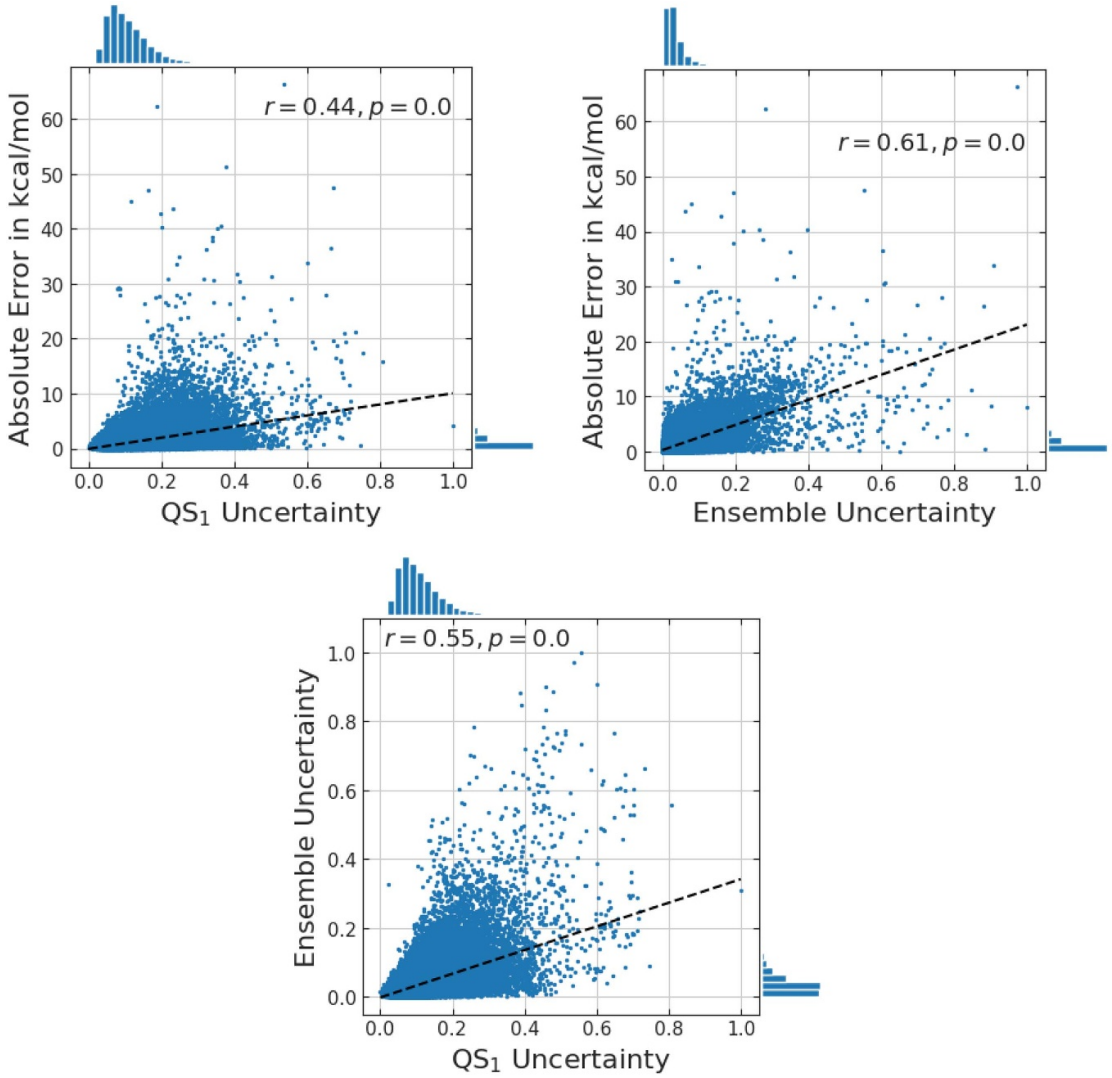

Figure 4. Correlation of (top left) the QS_1 metric and (top right) the ensemble variance with the absolute energy error, $\Delta E = \left|E - \hat{E}\right|$, for QM9 test data. (Bottom) Correlation between the variance estimated by the ensemble and the QS_1 uncertainty metric. The correlation coefficients are shown as insets along with the best fit lines.

obtained below for ensembling. Structures predicted with higher energy error obtain a higher uncertainty by the OED approach. To demonstrate that the performance of the OED approach is comparable to well-established AL techniques we trained the committee of three models. The uncertainty in the Query-by-Committee (QbC) approach can be measured as

$$\sigma_{\text{ens}}(x^*) = \sqrt{\frac{1}{N_{\text{ens}}} \sum_{i=1}^{N_{\text{ens}}} \left(\hat{y}_i(x^*) - \hat{\bar{y}}(x^*)\right)^2}, \tag{24}$$

where N_{ens} is the number of models in the committee, i.e. $N_{\text{ens}} = 3$. $\hat{\bar{y}}(x^*) = 1/N_{\text{ens}} \sum_{i=1}^{N_{\text{ens}}} \hat{y}_i(x^*)$ is the mean of the energy prediction over the committee.

Figure 4 (top right) shows that the correlation of the absolute energy error with the uncertainty metric obtained employing a committee is comparable to the correlation obtained using our method. The correlation coefficient has a value of 0.61. The difference is negligible when one takes into account that the OED approach does not need to train multiple models. In general, the proposed approach is N_{ens}-times more efficient than the QbC method.

For the sake of completeness, we studied the correlation between uncertainty estimates. Figure 4 (bottom) shows a strong correlation between them with a correlation coefficient of 0.55. For the comparison of the QbC approach with a few other approaches, e.g. Monte Carlo dropout [19, 20], feature space distances [18, 21, 22] and latent space distances [20], see elsewhere [20].

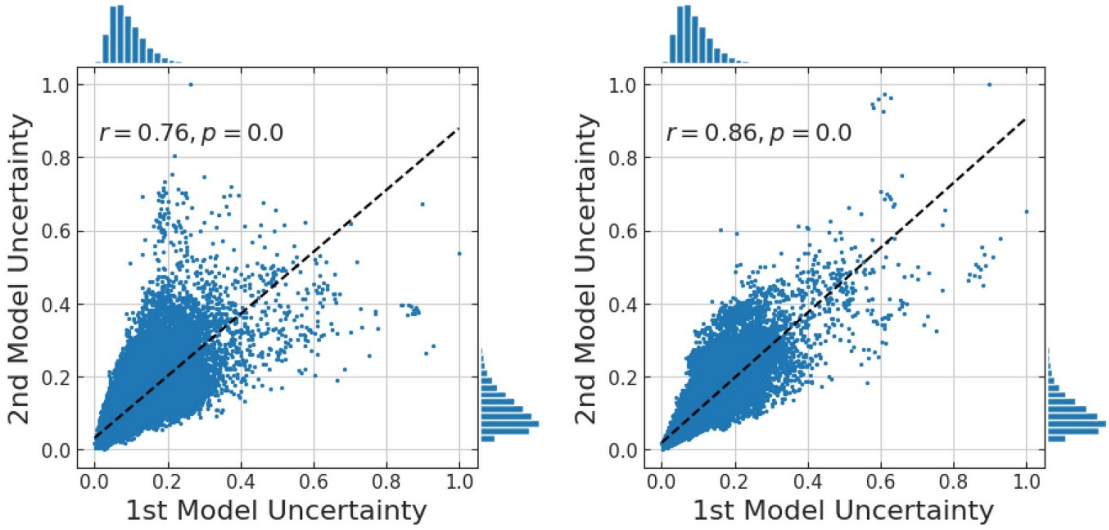

Figure 5. (Left) Correlation of the OED uncertainty metrics calculated using parameters of two independently initialized and trained GM-NN models. Both models have been trained to the same degree. (Right) Correlation of the OED uncertainty metrics calculated using the same model but trained to different degrees. The first model has been trained for 5000 epochs. The second model has been trained for 112 epochs.

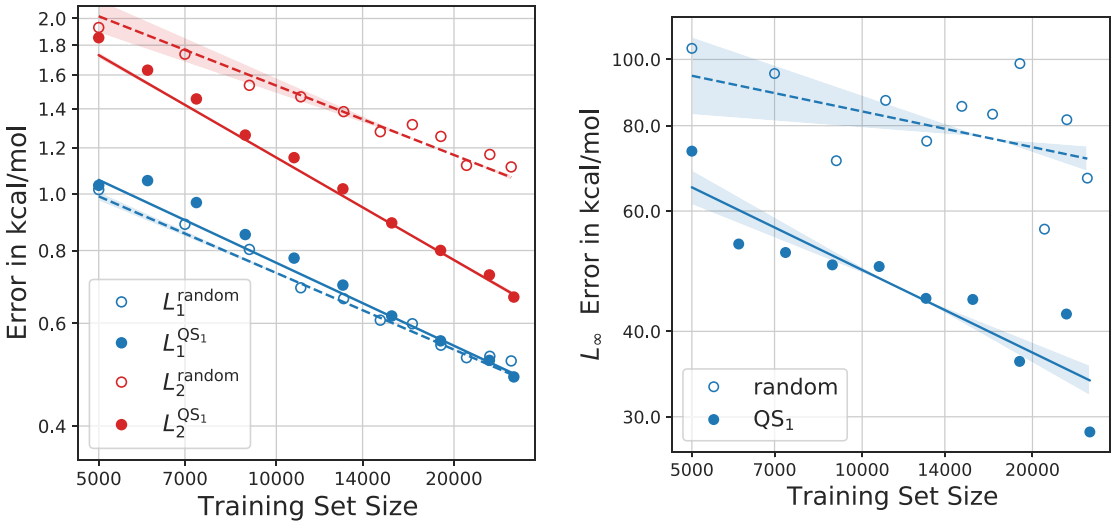

Figure 6. Log–log plot of (left) the mean absolute (L_1) and the root-mean-squared errors (L_2), and (right) the maximal error (L_∞) in the predicted energies on the QM9 data set. Structures are selected randomly or using query strategy QS_1. For each learning curve a linear fit is plotted in addition.

Besides the correlation of the OED uncertainty with the actual error, the correlation between uncertainties obtained for different local minima of the model has been studied. For that purpose, on the one hand, two models were trained using the same training data but independent randomly initialized NN parameters. Figure 5 (left) shows the correlation between uncertainties of two converged, i.e. trained to the same degree, models with a linear correlation coefficient of 0.76. On the other hand, the correlation between uncertainties of the same model but trained to a different degree is shown in figure 5 (right). In this case, we have found that the uncertainty obtained after training for 112 epochs correlates strongly with the uncertainty estimated for the converged model with a correlation coefficient of 0.86. These findings show that different local minima produce similar results. In particular, these results show that it is enough to train the model only shortly to obtain the desired uncertainty estimate, which improves the computational efficiency of the proposed approach considerably. Equivalent results can be obtained for other training data sets and, for the sake of brevity, will be left out in other sections. Given the deviation between the estimated NN uncertainties, an interesting question arises about the ensembling of them as well as about a combination with the QbC variance to achieve an even better correlation with the actual error. Unfortunately, this is out of the scope of this paper and will be studied in our future works.

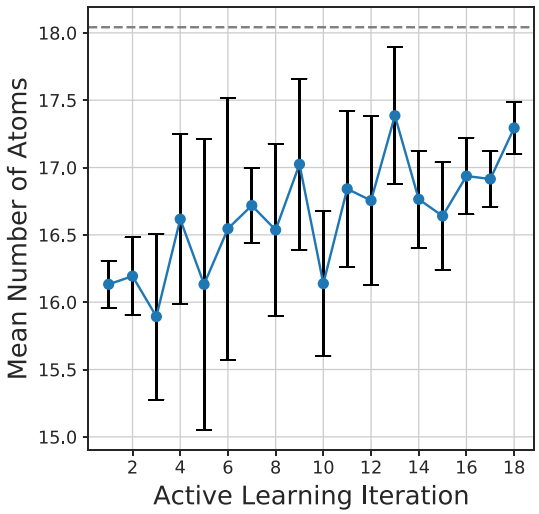

Figure 7. The mean number of atoms in molecules selected from the QM9 data set using the QS_1 metric. Error bars represent the standard deviation. The dashed gray line represents the mean number of atoms obtained using the random selection.

Figure 6 compares the mean absolute error (MAE, L_1), the root-mean-squared error (RMSE, L_2), and the maximal error (MAXE, L_∞) in predicted energies obtained using the models trained on actively and randomly selected structures. All results are obtained averaging over three independent runs. The GM-NN model results is quite low values for MAE already when trained on randomly selected data points. AL can not improve on that. However, the RMSE does not reach the desired accuracy of 1 kcal mol^{-1} for randomly chosen training data even after training on 25 000 structures. The MAXE can have values of about 100 kcal mol^{-1} which makes the model too inaccurate for the energy prediction of these unusual or even extrapolative structures.

Applying the AL scheme we were able to reduce the maximal error by a factor of 2.3, measured for the models trained on 25 000 structures. This confirms that the AL algorithm selects molecules that better represent unusual molecules and, therefore, reduces the overall maximal error. The RMSE is reduced by a factor of about 1.7, again for the models trained on 25 000 structures. We were able to reach the accuracy of 1 kcal mol^{-1} using only about 13 000 structures. In reference [6], we had obtained an RMSE of 0.63 kcal mol^{-1} when training on 110 426 randomly selected structures. Using our AL approach it was now possible to reach an RMSE value of 0.67 using less than a quarter of the number of structures used previously.

To further test the proposed AL approach we have built a training data set containing 105 508 structures. The uncertainty of the GM-NN model at each AL iteration was estimated after training for 250 epochs. In total, 32 AL iteration were performed. Training the NN on the final training data for 5000 epochs we obtained an RMSE value of 0.28 kcal mol^{-1} and an MAE value of 0.21 kcal mol^{-1}. Compared to the results obtained in reference [6] for 110 426 randomly selected structures there is an improvement of a factor 2.25 and 1.29, respectively. Additionally, using the QS_1 strategy we could reduce the maximal error from 62.06 kcal mol^{-1} to a value of 2.24 kcal mol^{-1} and lower the number of structures with absolute error larger than 1 kcal mol^{-1} by a factor of 6 (about 90 structures for AL). With that we can conclude that the presented AL approach allows us to construct transferable and uniformly accurate machine-learned potentials, which used to be impossible using random selection.

Finally, we investigated the sizes of the molecules selected by the active learning approach at each iteration. Figure 7 shows that the QS_1 approach selects on the average smaller structures than the ones drawn randomly. This implies that the smaller structures in the QM9 data set contain local environments relevant to the larger structures. Note that a similar trend was obtained in [26]. For example, authors in [26] obtained the mean number of atoms of around 16 for the training data size of 6000 which is similar to our result. However, in contrast to their results we see that the model tries to select larger structures with increasing iteration step. This difference most probably comes from the different sizes of the training sets.

All GM-sNN models were trained on one NVIDIA Tesla V100-SXM2-32GB GPU each for 5000 training epochs. The training took from about 3 h (5000 structures) to 14 h (25 261 structures).

3.3. N-ASW: molecular dynamics data

As a final test, we apply our AL approach to the data set recently used to study the adsorption and desorption dynamics of nitrogen atoms on top of amorphous solid water (ASW), which is relevant in astrochemical

processes [7]. The N-ASW data set is available directly from reference [38]. The goal of this section is to show to which extent the AL algorithm can be useful for real-time chemical simulations.

The purpose of the N-ASW data set was to describe the interaction of a nitrogen atom with an ASW surface. The model used in reference [7] contained 1498 atoms. To train a NN for this system highly heterogeneous data was needed. Therefore, the N-ASW data set contains structures with 3 to 378 atoms which result in 28 715 structures in total. Energies and atomic forces are calculated at the PBEh-3c/def2-mSVP [39] level of theory.

In this work, we split the data into five classes. The first class, C1, contains structures with $3 \leqslant N_{\text{at}} \leqslant 9$. The second class, C2, contains structures with $N_{\text{at}} = 22$ atoms and is the largest one with 18 735 structures in total. Other classes, C3, C4, and C5, contain structures with $N_{\text{at}} = 37$, $N_{\text{at}} = \{90, 91\}$, and $N_{\text{at}} \geqslant 100$, respectively. For this data set, a cutoff radius of 5.5 Å was used to include the long-range interactions of the nitrogen with the water ice surface.

Each AL cycle was initialized by randomly drawing 1000 structures for training and another 1000 for validation. Using the parameters of the trained models the uncertainty metric given in equation (14) was calculated and new structures were selected from the pool. Similar to the previous sections the pool contained the structures used for validation. In each iteration the model was allowed to select $0.1 \cdot N_{\text{train}}$ new structures and add them to the training set until the latter reached the size of 6105. This took 20 AL iterations in total, including the initialization.

The computational cost of the active learning algorithms was almost independent of the active learning iteration. The selection of m new training samples using the query strategy QS_1 required about 9 min on a single Intel Xeon CPU E5-2640 4. The query strategy QS_2 was about 80 times slower.

Figure 8 shows the correlation of the QS_1 metric (top left) and the QS_2 metric (top right) with the absolute force error, $\Delta F = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right|$. From the figure one can see that both OED metrics are strongly correlated with the absolute force error. We obtained correlation coefficients of 0.88 and 0.79 for the QS_1 and QS_2 metrics, respectively. Here we have found that the QS_1 metric correlates better with the force error than QS_2. This appears to be caused mainly by the class C1 of the training data. Removing it from the data set results in correlation coefficients of 0.94 for QS_2 and 0.92 for QS_1. This can be explained as follows. The AL algorithm recognizes that the smallest structures are underrepresented in the data set since the model can barely transfer the knowledge obtained from the bigger structures using a cutoff of 5.5 Å. Therefore, it labels them with larger values of the uncertainty metric, but the force error is rather small for these simple structures. Interestingly, the QS_1 metric is less sensitive to it and can recognize the minor importance of these structures. Figure 8 shows that both AL algorithms recognize correctly that the last two data classes, C4 and C5, are the most relevant ones for a better generalization of the model. The most abundant data class, C2, is already well represented using the initial 1000 training points. Both algorithms assign small uncertainty metric values to C2 structures.

Similar to section 3.1 we calculated the correlation between both uncertainty metrics, QS_1 and QS_2. It should be mentioned that we have found a good correlation for all data classes except for C1 which contains structures with $3 \leqslant N_{\text{at}} \leqslant 9$. Therefore, it was left out in the correlation plot, see figure 8 (bottom left), as well as in figure 8 (bottom right). We have obtained a correlation coefficient of 0.94 between QS_1 and QS_2 metric. Any considerable improvement could not be found when using combined uncertainty estimation QS_3. The linear correlation coefficient reached only a value of 0.95 for $\beta = 4.6$.

Figure 9 compares the root-mean-squared error (RMSE) and the maximal error (MAXE) in predicted force and energy errors depending on the training set size. The structures used for the training of the respective models are selected randomly, or employing the QS_1 and QS_2 approaches. All results are obtained averaging over three independent runs. It can be seen that employing AL algorithms improves both the RMSE and MAXE in predicted energies and forces.

The RMSE in predicted forces is reduced by a factor of 1.2 for the training set size of 6105 when using the QS_1 metric. The MAXE is reduced by a factor of 2.4 for the training set size of 6105, and the desired accuracy of 1 kcal mol^{-1} Å^{-1} is reached already after training on about 2500 structures. Using random selection even after training on 6105 structures the maximal error is about 1.29 kcal mol^{-1} Å^{-1} and around 22 structures have an error in predicted forces larger than 1 kcal mol^{-1} Å^{-1}. After training on 3000 randomly selected structures we obtained around 120 structures that had force errors larger than 1 kcal mol^{-1} Å^{-1}. The MAXE obtained using around 3000 actively selected structures is 0.70 kcal mol^{-1} Å^{-1}. These values were obtained with the QS_1 metric, the results for QS_2 are similar.

In figure 9 one can see an even stronger improvement in RMSE and MAXE for the total energies when employing the AL scheme. The RMSE is lower by a factor of 1.8 than the one obtained using the model trained on randomly selected data points and equals 0.25 kcal mol^{-1} (6105 training points). The MAXE is reduced from 8.34 to 1.14 kcal mol^{-1}. Only around 9 structures show an energy error larger than 1 kcal mol^{-1} employing the model which was trained on 6105 actively selected samples (1000 of them were drawn

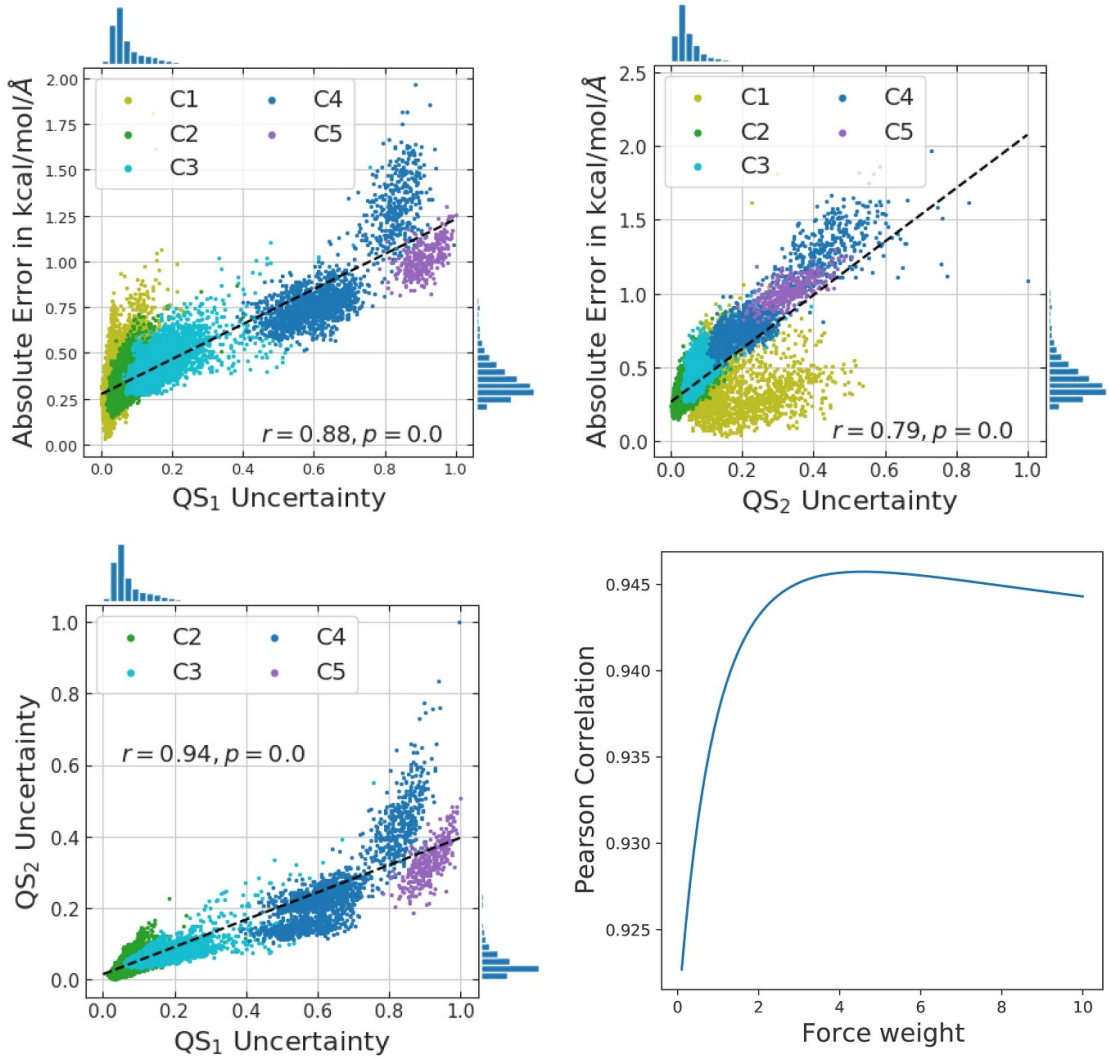

Figure 8. Correlation of the (top left) QS_1 metric and the (top right) QS_2 metric with the absolute force error $\Delta F = \frac{1}{3N_{at}} \sum_{i=1}^{N_{at}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right|$ on the N-ASW data set. For both graphs, the correlation coefficient is shown as an inset along with the best fit line. (Bottom left) Correlation between QS_1 and QS_2 uncertainty metric. The correlation coefficient is shown as an inset along with the best fit line. (Bottom right) The dependence of the correlation coefficient obtained for the QS_3 metric and the absolute force error on the force weight β in equation (22). For both graphs, the first data class C1 was left out, see text.

randomly from the data set in the initialization step), while for 6105 randomly selected samples around 570 show an energy error larger than 1 kcal mol^{-1}. Similar to the discussion of the force errors all values were obtained with the QS_1 metric. The QS_2 approach leads to a similar outcome for the RMSE in predicted energies. However, the performance in MAXE is deteriorated by a factor of 2.4 compared to the QS_1 approach. This shows that the QS_1 is superior to QS_2 since it leads to a comparable improvement in force errors but much better performance in predicting total energies.

The above observations concerning the RMSE and MAXE in predicted forces and energies let us draw the following conclusion. Employing the proposed AL scheme allows us to create a uniformly accurate and transferable potential trained on 6105 structures, which can describe the nitrogen adsorption and desorption with the desired accuracy. To generate an equally accurate potential using the randomly selected structures would require much more data. Note that the potential obtained using only around 2500 data points can already be used for MD simulations. However, reference [7] aimed at binding energies, which required a high accuracy in the energies as well.

Finally, we want to study different approaches to define the threshold value of the OED uncertainty metric which is important for the application of the AL algorithm on-the-fly. We propose two approaches: (1) one can define the threshold value as the mean of the model's uncertainty over the training data, $\sigma_{\text{th}}^{(1)} = 1/N_{\text{train}} \sum_{i=1}^{N_{\text{train}}} \langle \Delta\sigma_{\hat{y}}^2(x_i) \rangle_{\mathcal{D}}$; (2) one can define the threshold value as the median of the model's uncertainty over the training data, $\sigma_{\text{th}}^{(2)} = \underset{x_i \in \mathcal{D}}{\text{median}} \langle \Delta\sigma_{\hat{y}}^2(x_i) \rangle_{\mathcal{D}}$. As a measure of informativeness of the

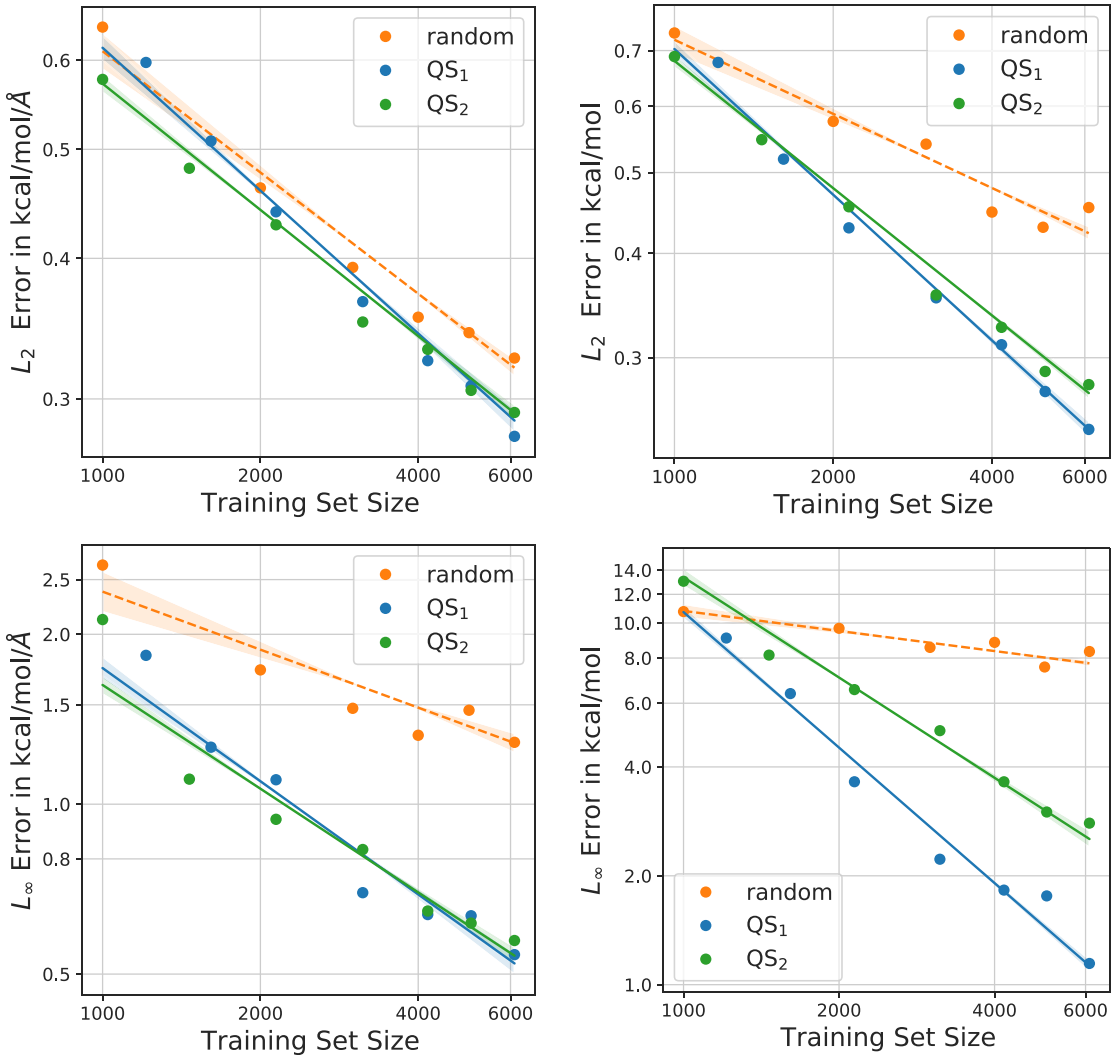

Figure 9. Log–log plot of the root-mean-squared error in predicted (top left) atomic forces and (top right) total energies on the N-ASW data set depending on the training set size. Log–log plot of the maximal error in predicted (bottom left) atomic forces and (bottom right) total energies depending on the training set size. Structures are selected randomly or using query strategies QS_1 and QS_2. For each learning curve a liner fit is plotted in addition.

selected structures we define the fraction $I = \langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}}/\sigma_{\text{th}}$. Structures with $I > 1$ are considered to be informative.

Figure 10 (left) shows the relative number of structures with $I > 1$ selected by both query strategies. Figure 10 (right) shows the ratio of the averaged informativeness of all structures selected in one AL iteration, i.e.

$$\bar{I}^{(1)} = \frac{1}{\#x^*}\sum_{x^*}\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}}/\sigma^{(1)}_{\text{th}}$$

or

$$\bar{I}^{(2)} = \underset{x^*\in\mathcal{P}}{\text{median}}\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}}/\sigma^{(2)}_{\text{th}}.$$

From figure 10 one can see that strongest improvement is achieved at around $N_{\text{train}} = 3000$, in accordance with the MAXE shown in figure 9. The relative number of informative structures decreases strongly for all query strategies except for $\sigma^{(2)}_{\text{th}}(QS_2)$. Additionally, $\sigma^{(2)}_{\text{th}}(QS_1)$ shows an increase in figure 10 (top left) after reaching $N_{\text{train}} = 4500$. This can be explained as follows. Many of the selected structures have I close to 1, in case of QS_2, or slightly less than 1, in case of QS_1. Therefore, we see only a small averaged informativeness in figure 10 (top right). However, some amount of the selected structures are nevertheless able to reduce the model's output variance significantly and, thus, have high values of OED uncertainty. Note that one would include only the most informative structures when learning on-the-fly and, therefore, reduce

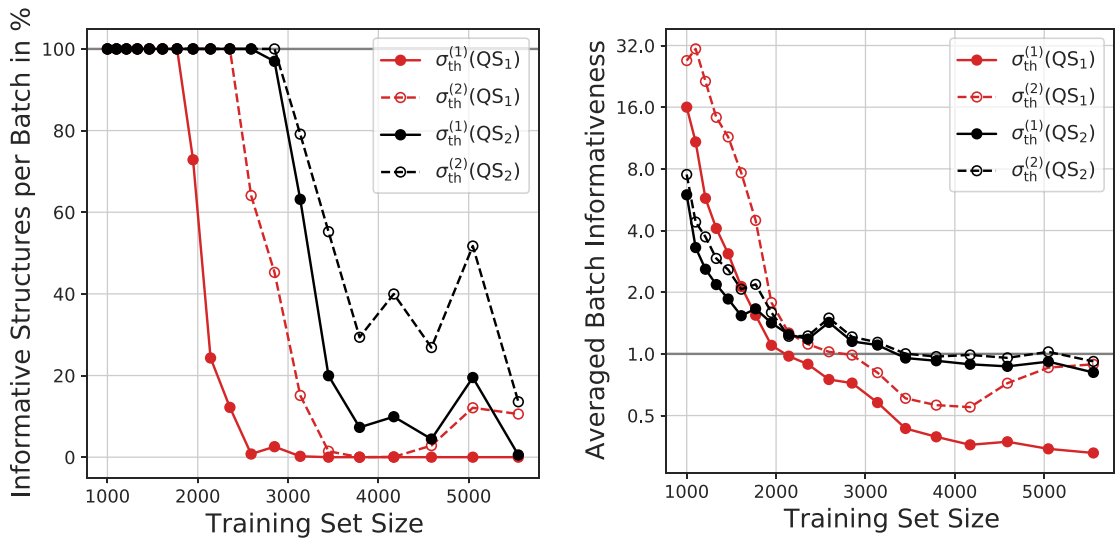

Figure 10. Comparison of different threshold values $\sigma_{th}^{(1)}$ and $\sigma_{th}^{(2)}$. (Left) The relative number of informative structures and (right) the log plot of the averaged informativeness of all structures selected in each AL iteration.

the number of structures necessary for the desired performance even further. In this work, we included $0.1 \cdot N_{train}$ new structures at each AL iteration which is less efficient, however allows us to test the performance of the AL algorithm and define the threshold values, $\sigma_{th}^{(1)}$ and $\sigma_{th}^{(2)}$.

Taking a more thorough look at the structures in the training data we have found that some of them were still underrepresented. These structures built the minority of the training data and have high values of the QS_1 and QS_2 metric. Therefore, they bias the threshold $\sigma_{th}^{(1)}$ to higher values. Taking this into account one can argue that $\sigma_{th}^{(2)}$ is superior to the latter giving the possibility to include the structures underrepresented but already included in the training set. This relationship can be easily seen comparing figure 9 and figure 10.

All GM-NN models for the N-ASW data set were trained on one NVIDIA Tesla V100-SXM2-32GB GPU each. The training of the model on 1000 structures for 5000 epochs took less than 4 h, and the training on 6105 structures for 5000 epochs was carried out during 17 h.

4. Conclusion

Machine learned potentials have been proven to have the potential to bridge the accuracy of *ab initio* methods and efficiency of empirical potentials. To construct potentials that are transferable and uniformly accurate for the chemical and conformational spaces of interest the model has to be able to define and select the extrapolative and most unusual structures for which, subsequently, the *ab initio* atomic forces and energies are calculated. Unfortunately, neural networks, the most frequently used machine learning approach in computational chemistry, have no inherent uncertainty estimators, as, for example, Gaussian processes have. For this reason it is not *a priori* clear, which data has to be included in the training set. The data sets employed are usually way larger than required or miss important regions of the configurational and chemical space.

In this paper, we proposed a novel active learning scheme for atomistic neural network potentials defined in the framework of the optimal experimental design. This approach uses the expected change in the estimated model's output variance to select structures that can be expected to reduce the generalization error of the model. The output variance is derived using the squared loss. Therefore, we were able to define three different query strategies based on the energy, the force, and the total losses.

To test the active learning approach we employed three different data sets. First, query strategies based on the energy and force squared losses were applied to an MD data set sampling the conformational space. Here, we have confirmed that the estimated gains are strongly correlated with the actual errors in predicted forces. We have seen that the active learning approach leads to a considerable reduction of the maximal error indicating that the most extrapolative and unusual structures are selected by the algorithm.

Next, the query strategy based on the energy squared loss was applied to a chemically diverse data set, the QM9 set [28, 29]. We have seen that employing the active learning scheme leads to a more accurate potential over the whole data set. We could achieve an RMSE close to the one obtained previously in reference [6] after training on randomly selected 110 426 structures, using only a fraction of the data set.

Finally, the AL approach was tested on the data set recently used to study the adsorption and desorption dynamics of nitrogen atoms on top of amorphous solid water. As a result, we obtained a model which predicts atomic forces with a maximal error of 0.54 kcal mol^{-1} $Å^{-1}$ (0 structures with force error $\geqslant$1 kcal mol^{-1} $Å^{-1}$) and total energies with a maximal error of 1.14 kcal mol^{-1} (around 9 structures with energy error of $\geqslant$1 kcal mol^{-1}). The model was trained on only 6105 structures. This is a great improvement over the model trained on 6105 randomly selected structures, which resulted in a maximal error of the predicted total energies of about 8.34 kcal mol^{-1} and failed to predict the total energies with the desired accuracy of 1 kcal mol^{-1} for around 570 structures. The maximal error in predicted forces is around 1.29 kcal mol^{-1} $Å^{-1}$.

We also studied the possible threshold values for the OED uncertainty metrics which is necessary to perform learning on-the-fly. We have found that the median of the expected change in the estimated output variance over the training set is the better choice compared to the mean. This is due to the fact that the mean value is more sensitive to the structures already present but underrepresented in the training set. In general, we observed a good correlation of the estimated informativeness of selected structures with the maximal error reduction. This indicates that the threshold value can be defined naturally using both the mean and the median over the data used for training. Therefore, it is possible to use the proposed algorithm on-the-fly.

In summary, we presented an efficient approach for the active selection of the most informative structures from molecular data sets. We have shown that it leads to a considerable reduction of the training set sizes and at the same time to a reduction of the generalization error. Additionally, we have shown that it is possible to naturally define a threshold value for the OED uncertainty metric. This allows the application of the proposed method to the generation of transferable and uniformly accurate potentials on-the-fly.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The authors acknowledge financial support received in the form of a Ph.D. scholarship from the Studienstiftung des deutschen Volkes (German National Academic Foundation). We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding EXC 2075 – 390740016 under Germany's Excellence Strategy. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech). We also like to acknowledge the support by the Institute for Parallel and Distributed Systems (IPVS) of the University of Stuttgart for providing computer time.

ORCID iDs

Viktor Zaverkin https://orcid.org/0000-0001-9940-8548
Johannes Kästner https://orcid.org/0000-0001-6178-7669

References

[1] Hornak V, Abel R, Okur A, Strockbine B, Roitberg A and Simmerling C 2006 *Proteins* **65** 712–25
[2] Vanommeslaeghe K *et al* 2010 *J. Comput. Chem.* **31** 671–90
[3] Halgren T A 1996 *J. Comput. Chem.* **17** 490–519
[4] Mackerell Jr A D 2004 *J. Comput. Chem.* **25** 1584–604
[5] Dral P O 2020 *J. Phys. Chem. Lett.* **11** 2336–47
[6] Zaverkin V and Kästner J 2020 *J. Chem. Theory Comput.* **16** 5410–21
[7] Molpeceres G, Zaverkin V and Kästner J 2020 *Mon. Not. R. Astron. Soc.* **499** 1373–84
[8] Settles B 2009 Active learning literature survey Computer Sciences *Technical Report* 1648 (University of Wisconsin–Madison)
[9] Vandermause J, Torrisi S B, Batzner S, Xie Y, Sun L, Kolpak A M and Kozinsky B 2020 *npj Comput. Mater.* **6** 20
[10] Guan Y, Yang S and Zhang D H 2018 *Mol. Phys.* **116** 823–34
[11] Li Z, Kermode J R and De Vita A 2015 *Phys. Rev. Lett.* **114** 096405
[12] Li Z 2014 On-the-fly machine learning of quantum mechanical forces and its potential applications for large scale molecular dynamics PhD Thesis King's College, London
[13] Browning N J, Ramakrishnan R, von Lilienfeld O A and Roethlisberger U 2017 *J. Phys. Chem. Lett.* **8** 1351–9
[14] Huang B and von Lilienfeld O A 2020 *Nat. Chem.* **12** 945–51
[15] Smith J S, Nebgen B, Lubbers N, Isayev O and Roitberg A E 2018 *J. Chem. Phys.* **148** 241733
[16] Gastegger M, Behler J and Marquetand P 2017 *Chem. Sci.* **8** 6924–35
[17] Zhang L, Lin D Y, Wang H, Car R and E W 2019 *Phys. Rev. Mater.* **3** 023804
[18] Schran C, Behler J and Marx D 2020 *J. Chem. Theory Comput.* **16** 88–99

[19] Gal Y and Ghahramani Z 2016 Dropout as a Bayesian approximation: representing model uncertainty in deep learning *Proc. 33rd Int. Conf. Machine Learning* (*Proc. Machine Learning Research* vol 48), eds M F Balcan and K Q Weinberger (New York, USA: PMLR) pp 1050–9
[20] Janet J P, Duan C, Yang T, Nandy A and Kulik H J 2019 *Chem. Sci.* **10** 7913–22
[21] Janet J P and Kulik H J 2017 *J. Phys. Chem.* A **121** 8939–54
[22] Nandy A, Duan C, Janet J P, Gugler S and Kulik H J 2018 *Ind. Eng. Chem. Res.* **57** 13973–86
[23] Cohn D A 1996 *Neural Netw.* **9** 1071–83
[24] MacKay D J C 1992 *Neural Comput.* **4** 590–604
[25] Fedorov V 1972 *Theory of Optimal Experiments* (New York: Academic)
[26] Gubaev K, Podryabinkin E V and Shapeev A V 2018 *J. Chem. Phys.* **148** 241727
[27] Podryabinkin E V and Shapeev A V 2017 *Comput. Mater. Sci.* **140** 171–80
[28] Ruddigkeit L, van Deursen R, Blum L C and Reymond J L 2012 *J. Chem. Inf. Model.* **52** 2864
[29] Ramakrishnan R, Dral P O, Rupp M and von Lilienfeld O A 2014 *Sci. Data* **1** 140022
[30] Reddi S J, Kale S and Kumar S 2019 (arXiv:1904.09237) [cs.LG]
[31] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems software available from tensorflow.org (available at: https://www.tensorflow.org/)
[32] Geman S, Bienenstock E and Doursat R 1992 *Neural Comput.* **4** 1–58
[33] Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
[34] Grimme S, Antony J, Ehrlich S and Krieg H 2010 *J. Chem. Phys.* **132** 154104
[35] Grimme S, Ehrlich S and Goerigk L 2011 *J. Comput. Chem.* **32** 1456
[36] Rassolov V A, Pople J A, Ratner M A and Windus T L 1998 *J. Chem. Phys.* **109** 1223
[37] Prechelt H 2012 *Neural Networks: Tricks of the Trade* (Berlin: Springer)
[38] Molpeceres G, Zaverkin V and Kästner J 2020 N-ASW: molecular dynamics data (v1) (available at: http://doi.org/10.5281/zenodo.4013889)
[39] Grimme S, Brandenburg J G, Bannwarth C and Hansen A 2015 *J. Chem. Phys.* **143** 054107