The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

Local-environment-guided selection of atomic structures for the development of machine-learning potentials

Cite as: J. Chem. Phys. 160, 074109 (2024); doi: 10.1063/5.0187892
Submitted: 17 November 2023 • Accepted: 26 January 2024 •
Published Online: 21 February 2024

Renzhe Li,[1,2] Chuan Zhou,[1] Akksay Singh,[1,3,4] Yong Pei,[2] Graeme Henkelman,[3,4,a)] and Lei Li[1,a)]

AFFILIATIONS

[1] Shenzhen Key Laboratory of Micro/Nano-Porous Functional Materials (SKLPM), Department of Materials Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, People's Republic of China
[2] College of Chemistry, Xiangtan University, Xiangtan 411105, Hunan Province, People's Republic of China
[3] Department of Chemistry, The University of Texas at Austin, Austin, Texas 78712, USA
[4] Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas 78712, USA

[a)] **Authors to whom correspondence should be addressed:** henkelman@utexas.edu and lil33@sustech.edu.cn

ABSTRACT

Machine learning potentials (MLPs) have attracted significant attention in computational chemistry and materials science due to their high accuracy and computational efficiency. The proper selection of atomic structures is crucial for developing reliable MLPs. Insufficient or redundant atomic structures can impede the training process and potentially result in a poor quality MLP. Here, we propose a local-environment-guided screening algorithm for efficient dataset selection in MLP development. The algorithm utilizes a local environment bank to store unique local environments of atoms. The dissimilarity between a particular local environment and those stored in the bank is evaluated using the Euclidean distance. A new structure is selected only if its local environment is significantly different from those already present in the bank. Consequently, the bank is then updated with all the new local environments found in the selected structure. To demonstrate the effectiveness of our algorithm, we applied it to select structures for a Ge system and a $Pd_{13}H_2$ particle system. The algorithm reduced the training data size by around 80% for both without compromising the performance of the MLP models. We verified that the results were independent of the selection and ordering of the initial structures. We also compared the performance of our method with the farthest point sampling algorithm, and the results show that our algorithm is superior in both robustness and computational efficiency. Furthermore, the generated local environment bank can be continuously updated and can potentially serve as a growing database of feature local environments, aiding in efficient dataset maintenance for constructing accurate MLPs.

I. INTRODUCTION

Atomic simulations are valuable for understanding the behavior of matter at the atomic and molecular levels and have been widely used in physics, chemistry, materials science, and engineering. The key step in atomic simulations is to construct a high-dimensional potential energy surface (PES) of the target system,[1–8] which is usually achieved with either classical force fields or first-principles methods.[9–13] In the past two decades, machine learning has been successfully used to learn PESs from the reference data obtained from first-principles calculations.[14–27] Machine-learning potentials (MLPs) exhibit near first-principle-level accuracy with an affordable computational cost and enable large-scale and long-timescale simulations.

Until now, various types of machine-learning methods have been developed for MLP construction, including atom-centered artificial neural network (ANN) method,[14,17,24,26,28–34] Gaussian approximation potentials (GAPs),[16,35,36] graph neural network (GNN) potential, gradient-domain ML (GDML), spectral neighbor analysis potentials (SNAPs), moment tensor potential (MTP),[37,38] and SGPR-based universal potentials (SGPR: sparse Gaussian process regression),[39,40] among others. These methods have found widespread applications in the fields of materials science, chemistry, and biology. For example, the ANN method has been successful in

fitting potential energy surfaces for Si and water systems, resulting in reasonably accurate predictions of phase behaviors.[41,42] The deep potential approach has also demonstrated a notable performance in reproducing phase diagrams of bulk water and gallium, approaching the accuracy of first-principles methods.[43,44] MLPs developed using GAPs for the elements V, Nb, Mo, Ta, and W have shown encouraging accuracy in computing elastic, thermal, and surface properties of the corresponding bulk materials.[45] The SGPR method is also effective in generating universal potentials for noncyclic hydrocarbons, irrespective of the number of C=C double bonds,[46] as well as its successful application in investigating the stability of high Al-doped $Li_{1.22}Ru_{0.61}Ni_{0.11}Al_{0.06}O_2$ at a high voltage.[47]

MLPs have shown promising applications in atomic simulations, but constructing a reliable MLP for a specific system can be quite challenging. A critical step in developing a MLP is collecting a representative set of reference data that establishes the MLP's reliability and applicability range.[48–53] Typically, *ab initio* molecular dynamic (AIMD) simulations are used to generate these reference data.[54–59] Metadynamics and other techniques may be employed to enhance configuration sampling.[60–63] Alternatively, a more efficient approach involves generating large sets of configurations that cover various distinct regions using less-accurate, cost-effective calculations. These configurations can then be subsampled to produce a smaller dataset for high-level calculations, an approach known as delta learning. However, inefficient reference data can cause model failure. Additionally, generating reference data requires high-level calculations, and an excessive number of redundant atomic structures in the reference configurations can increase computational costs and potentially bias the model.[64–67] On the other hand, regardless of various ML-algorithms, the training data determine the stability and application scope of the generated model. Since MLFF algorithms can only be trained on a limited dataset, a balance between data sampling completeness and computing resources is a key issue in the development of MLPs. Therefore, it is valuable to have an effective prescreening approach for the selection of representative configurations before performing high-level calculations.

Various methods have been proposed to select representative data, aiming to minimize redundancy and effectively capture the diversity of the reference data.[68–71] Random selection (RS) is a widely employed approach due to its convenience and efficiency.[27,70–79] However, it often leads to an imbalanced distribution of data across configuration space. To address the omission of certain structures, active learning-based approaches with uncertainty quantifications have been suggested.[80–83] These methods are typically tailored to the specific simulation of interest and require on-the-fly training of model ensembles with the addition of new data structures. Additionally, their performance can be influenced by the choice of the uncertainty quantification method.

Alternatively, a more logical and practical approach to reduce data redundancy is to measure the atomic configurational similarity. One common technique for structure selection is to consider non-linear dependencies between structural features, using methods such as clustering[27,70] or farthest point sampling (FPS).[69,84] Clustering involves partitioning the reference data into multiple clusters in a feature space based on the Euclidean distance or the kernel ridge regression (KRR) distances,[85] followed by selecting a portion of data from each cluster. This method is sensitive to various factors, including the number of clusters, the fingerprint scaling method, and the ordering of the data within clusters. In contrast, FPS involves fewer parameters to determine in the initial stage. It begins by selecting an initial data point and then chooses the point with the farthest distance relative to the already selected points until a specific number of data points are obtained. FPS has been widely employed in constructing training data for machine-learning applications. Nevertheless, its performance hinges on a predetermined number of structures for selection and exhibiting instability when applied to the development of the MLPs. Notably, the computational cost escalates with the number of structures. Despite the existence of numerous algorithms for data reduction, there remains a demand for the development of robust methods with high computational efficiency for the selection of representative structures.

In this study, we describe a prescreening approach to select representative atomic structures from pre-generated datasets. We apply our algorithm to a Ge system and a $Pd_{13}H_2$ nanoparticle system and demonstrate that our algorithm effectively reduces the size of the training dataset without a loss of accuracy of the MLP model. We also evaluate the performance variation of the algorithm with a threshold value, which is used to distinguish the local environments of two atoms. Our algorithm demonstrates robustness and a higher computational efficiency when compared to the FPS method.

II. COMPUTATIONAL DETAILS

A. Local-environment-guided screening algorithm for structure selection

The ANN method employs an independent neural network for each element to learn the contributions of each atom and its neighbors and then sums the contributions over all the atoms to obtain the total energy of the system. The representation of the local environment around each central atom due to its neighbors determines the reliability and applicability range of the trained MLP model. Inspired by this, we developed a local-environment-guided screening algorithm for structure selection. The core of this algorithm is to treat each atomic local environment independently, yet appreciate the diversity of atomic environments across different structures. In general, two different structures can have many atoms that share similar local environments within the respective structures, yet there may still be many atoms that have different local environments. We employed the Behler–Parrinello (BP) symmetry function to describe the local environment of each central atom. Specifically, the local environment of atom i is defined as $G_i = [G_{i,1}, G_{i,2}, \ldots, G_{i,n}]$ when n BP symmetry functions are employed for atom i. To evaluate the dissimilarity between two local environments, the Euclidean distance between two fingerprint vectors is defined as follows:

$$d(G_i, G_j) = \sqrt{\sum_{k=1}^{n} \left(G_{i,k} - G_{j,k}\right)^2}. \quad (1)$$

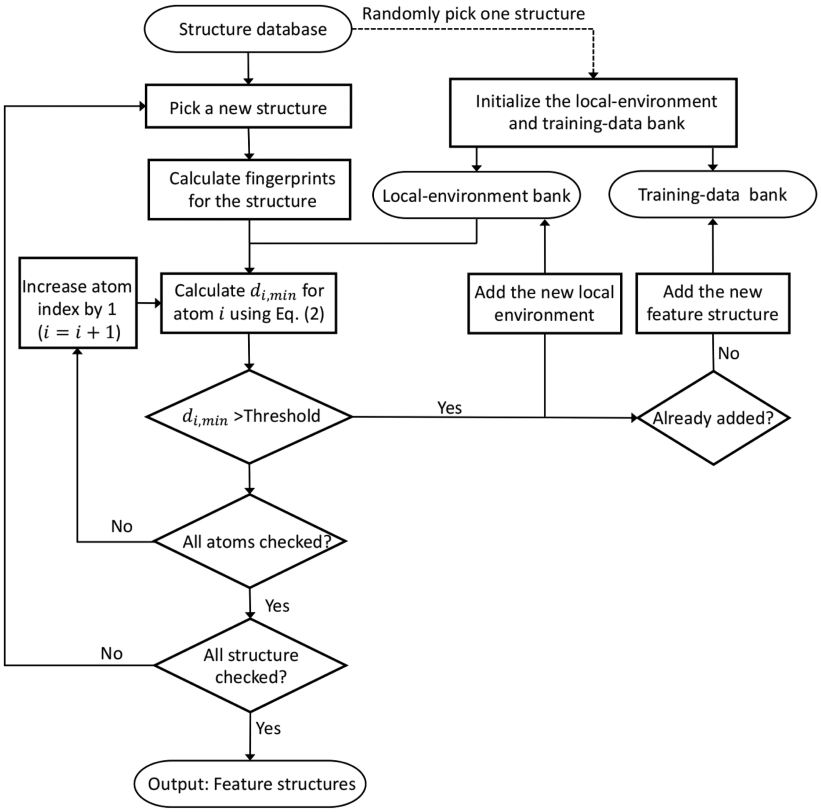

FIG. 1. Flowchart of the local-environment-guided selection algorithm.

Figure 1 shows the flowchart of the local-environment-guided screening algorithm. Here, we assume that a database of atomic structures is generated in advance, which can be achieved via various techniques, such as classical MD simulations, manual construction, and derivation from known chemical databases. In this algorithm, two banks, namely the local environment and training data banks, are defined to collect unique local environments and representative structures, respectively. The local environment bank (*LE*-bank) consists of a set of fingerprint vectors. It can be initialized by randomly selecting one fingerprint vector from the structure database. For each atom (i) in a specific structure, the Euclidean distances between its fingerprint vector and those stored in the local environment bank are calculated. The minimum Euclidean distance [$d_{i,\min}(G_i)$] is determined to describe the similarity between the local environment of this atom and those stored in the bank [see Eq. (2)]. A unique local environment is identified when $d_{i,\min}$ is larger than a predefined threshold value (d_{th}). The structure with a unique local environment is considered as a representative structure to be included in the training data bank,

$$d_{i,\min}(G_i) = \min_{G_k \in LE\_Bank} \left(d(G_i, G_j^{LE-bank}) \right). \quad (2)$$

A step-by-step illustration of the algorithm is given in the following:

Step 1. Select a new structure (S) from the structure database.
Step 2. Calculate fingerprints for structure S.
Step 3. Calculate $d_{i,\min}(G_i)$ for atom i in structure S with Eq. (2).
Step 4. Determine if $d_{i,\min}(G_i) > Threshold$ and if so, update the local environment bank with the fingerprint vector G_i. In addition, update the training data bank if structure S has not been included yet.
Step 5. Check if all atoms in structure S have been considered. If not, repeat Step 3 to 5; if yes, go to Step 6.
Step 6. Check if all structures in the original dataset have been considered. If not, repeat Step 1 to 6. If yes, end the program and output all selected structures in the training data bank.

B. Reference data

We used a Ge system and a $Pd_{13}H_2$ nanoparticle system as application examples to demonstrate the effectiveness of our algorithm for MLP training. The $Pd_{13}H_2$ dataset was collected from our previous study, which contains 19 802 atomic structures.[30] The Ge dataset was derived from the Si dataset reported by Bartók *et al.*[48] We transferred the atomic structures from the Si dataset to Ge atomic structures by expanding the lattice cell at the ratio of Ge–Ge and Si–Si equilibrium bond lengths. To enhance the diversity of local atomic environments, we additionally expanded or compressed the

obtained Ge atomic structures by a factor of 0.8–1.5 from the Ge–Ge equilibrium bond length. This process yielded a total of 7092 atomic structures for the Ge system, including bulk, surface configurations and crystalline and amorphous crystal phases.

The reference energies and forces of the Ge system were calculated by the density functional theory (DFT) method, as implemented in the Vienna *Ab initio* Simulation Package (VASP). The projected augmented wave method and a plane wave basis set with an energy cutoff of 500 eV were employed to represent the core–valence electron interactions. The generalized gradient approximation with the Perdew–Burke–Ernzerhof functional was used to describe electronic exchange–correlation interactions. An automatic k-point generation scheme was employed, and the number of k-points was set to ensure that its multiplication with the lattice constant was greater than 20 Å.

C. Atom-centered neural network model

We used the ANN algorithm as implemented in the Python-based atom-centered machine-learning force field (PyAMFF) package[86] to fit machine-learning potentials for the Ge and $Pd_{13}H_2$ systems. The ANN algorithm interprets atomic structures using atom-centered symmetry functions to represent the local atomic environments, commonly referred as to structural fingerprints $G_{i,j}$. Neural network models were developed to describe the relation between the structural fingerprints and the reference energy of the structure. The total energy of the structure is the sum of atomic energies (E_i), which has the following functional form (assuming a NN model with one hidden layer):

$$E_i = \sum_k w_{k1}^{12} \cdot f_k^1\left(\sum_j w_{jk}^{01} \cdot G_{i,j} + b_k^1\right) + b_1^2, \tag{3}$$

where w_{jk}^{01} represents the weight connecting node j in layer 0 and node k in layer 1. In addition, b_k^1 and f_k^1 are the bias and activation function of node k in layer 1, respectively. Note that layer 0 represents the input layer and layer 2 is the output layer.

In this study, we used modified BP symmetry functions to describe the atomic local environments. The radial (G_i^{II}) and angular (G_i^I) terms were constructed with the following equations:

$$G_i^I = \sum_{j\neq i}^{all} e^{-\eta\frac{(R_{ij}-R_s)^2}{R_c^2}} f_c(R_{ij}), \tag{4}$$

$$G_i^{II} = 2^{1-\zeta} \sum_{\substack{j,k\neq i \\ j\neq k}}^{all} (1+\lambda\cos(\theta_{ijk}-\theta_s))^\zeta \times e^{-\eta\frac{\left(R_{ij}^2+R_{ik}^2+R_{jk}^2\right)}{R_c^2}} f_c(R_{ij}) f_c(R_{ik}) f_c\left(R_{jk}\right), \tag{5}$$

where R_{ij} is the distance between the central atom i and the neighboring atom j. Here, η and ζ are the widths of the Gaussian basis and the angular resolution, respectively, whereas R_s and θ_s are the position offsets of the radial and angular distributions of all neighboring atoms within the cutoff radius R_c, respectively. The value of λ is ±1 and $f_c(R_{ij})$ is a switching function [Eq. (6)] that transitions smoothly to zero at a cutoff radius, R_c,

$$f_c(R_{ij}) = \begin{cases} 0.5\cdot\left[\cos\left(\frac{\pi R_{ij}}{R_c}\right)+1\right] & for\ R_{ij}\le R_c, \\ 0 & for\ R_{ij} > R_c. \end{cases} \tag{6}$$

For the Ge system, we adopted 24 G_i^I functions and 16 G_i^{II} functions. For the $Pd_{13}H_2$ nanoparticle system, we adopted 36 G_i^I functions and 12 G_i^{II} functions (see details in the supplementary material). Note that our focus is not to develop general MLP models for these systems. A larger training dataset will be necessary to develop MLP models that are accurate beyond the limited training set that we have used here.

III. RESULTS AND DISCUSSION

A. Performance test on the Ge system

First, we chose a single-element Ge system to evaluate the efficiency of our algorithm for structure selection. We systematically varied the threshold value (T_{Ge}) from 0.1 to 0.45 with an interval of 0.025 and monitored the corresponding changes in the number of structures and atoms within the selected structures. During initialization, we randomly selected one feature vector from the dataset and added its corresponding structure to the training data bank. The screening process of representative structures may vary depending on the initialization procedure. We performed a series of experiments to assess the impact of this factor on the algorithm's performance. Specifically, we conducted 50 independent screening processes by randomly shuffling the original dataset to generate 50 sets of training images and calculated the average number of the selected images and atoms at each T_{Ge}.

Figure 2(a) shows the variation of the average number of selected images and atoms in the selected training data bank with T_{Ge}. The original dataset consists of 7506 Ge structures (corresponding to T_{Ge} = 0.0), including a total of 707 440 atoms in bulk, slabs, nanowires, and the liquid phase. The number of images and the number of atoms exhibit similar dependencies on the threshold value. As T_{Ge} increased, the number of images and the total number of atoms included in the training data bank decreased monotonically. For example, at the threshold value of 0.1, we screened out 4238 representative structures, including 465 108 atoms. When the threshold value was 0.275, the number of representative structures decreased to 1,264, including 134 882 atoms. Figures 2(c) and 2(d) show the distribution of percentage errors of the structures and atoms in the training data bank obtained from 50 independent selections at T_{Ge} = 0.275, conforming to a normal distribution with a small variance.

To assess the effectiveness of our algorithm in selecting local environments, we analyzed the distribution of symmetry functions at specific values of T_{Ge} (0.0, 0.275, and 0.45). The results of this analysis are presented in Figs. 2(b) and S1. To illustrate this analysis, we take the distribution of the second radial symmetry function (G_2^I) as an example, as shown in Fig. 2(b). Initially, when considering the dataset selected at T_{Ge} = 0.0, the distribution of G_2^I was found to be highly concentrated between 0.0 and 0.2; only a small portion of the distribution extended beyond 0.2. As T_{Ge} increased,

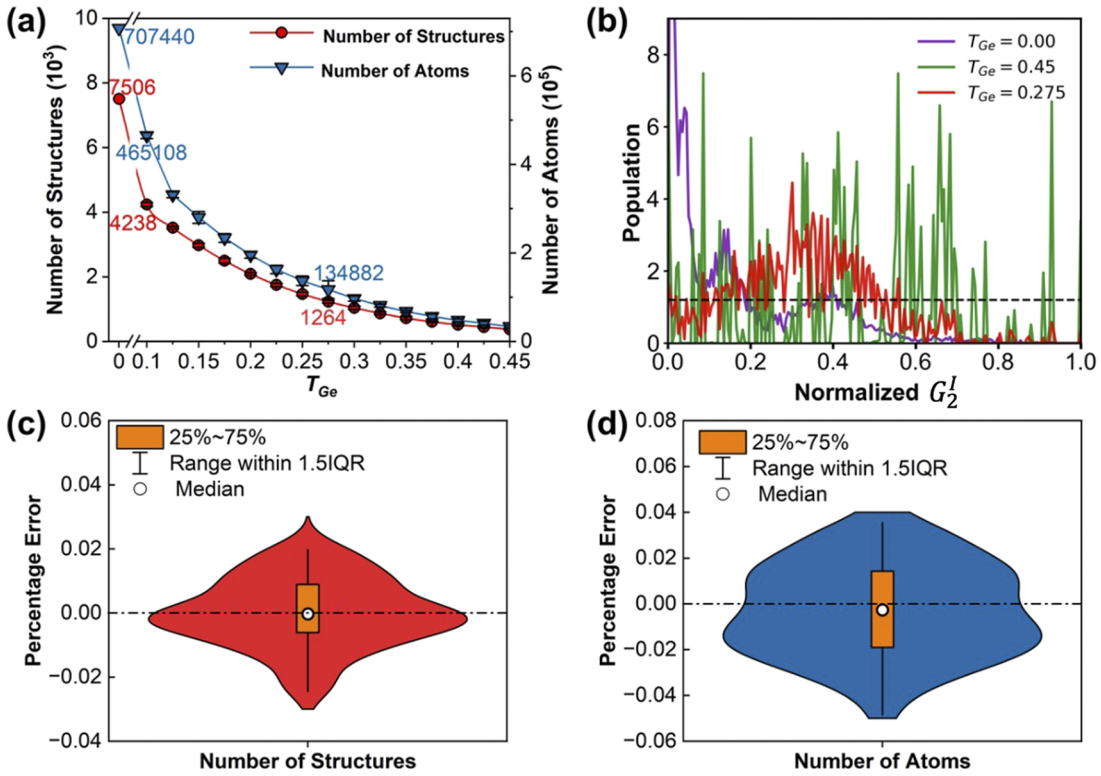

FIG. 2. (a) The number of structures and atoms in the training data bank as a function of the threshold value (T_{Ge}) employed in the structure screening process. (b) Distribution of the second radial symmetry function (G_2^I) for training images selected at T_{Ge} = 0.0, 0.275, 0.45. (c) Violin plot showing the percentage error distribution in the number of structures and atoms in the training data bank obtained from 50 independent runs at T_{Ge} = 0.275.

the algorithm screened out structures with similar local environments. This screening process led to a decrease in the population of G_2^I within the range of 0.0 to 0.2. Simultaneously, the population of local environments with $G_2^I > 0.2$ increased, resulting in a more uniform distribution of G_2^I from 0 to 1.0 for the selected dataset. Notably, T_{Ge} = 0.275 exhibits a Gaussian-like distribution with a peak value of G_2^I = 0.3. However, when T_{Ge} was further increased, the algorithm gradually overly screened out local environments, leading to many discontinuous breakpoints between 0.0 and 1.0. This discontinuity indicates that certain local environments were excessively filtered out. Nonetheless, our algorithm effectively modulates the distribution of local environments by adjusting the threshold value used to determine the dissimilarity between two local environments.

After selecting structures at different threshold values, we proceeded to assess the performance of the trained force field. For each threshold value, we used the selected structures as the training set to develop a machine-learning force field. We subsequently utilized this force field model to predict the energy and force of the original dataset and calculated the RMSE (denoted as the prediction set). To ensure the robustness of our findings, we repeated the entire process for each threshold value using 50 sets of training data obtained from independent screening processes. Figures 3(a) and 3(b) show the variation of average RMSEs of the energy and force as a function of T_{Ge}. For reference, the training RMSEs obtained from the model developed on all structures within the original dataset are represented by the dashed line.

As shown in Figs. 3(a) and 3(b), the energy and force RMSEs exhibit a similar trend in relation to the threshold value. The energy RMSE for the training set consistently decreases as the threshold value increases due to a reduction in training data. Conversely, the energy RMSE for the prediction set gradually increases with higher threshold values, which can be attributed to the sparsity of the training data and negatively impacts the predictive performance of the trained model. For example, the observed discontinuity in the distribution of G_2^I for the dataset selected at T_{Ge} = 0.45 suggests the loss of certain local environments. The model trained on such datasets exhibits an inferior prediction performance on the structures that contain missing local environments. Regarding the force RMSEs, a deviation is observed at $T_{Ge} > 0.275$. As T_{Ge} increases from 0.275, on average, the prediction performance of the MLP starts to quickly deteriorate [Figs. 3(a) and 3(b)]. Furthermore, the variance in the prediction performance, for both energy and forces, also rises. There is no significant performance sacrifice for the MLP model devel-

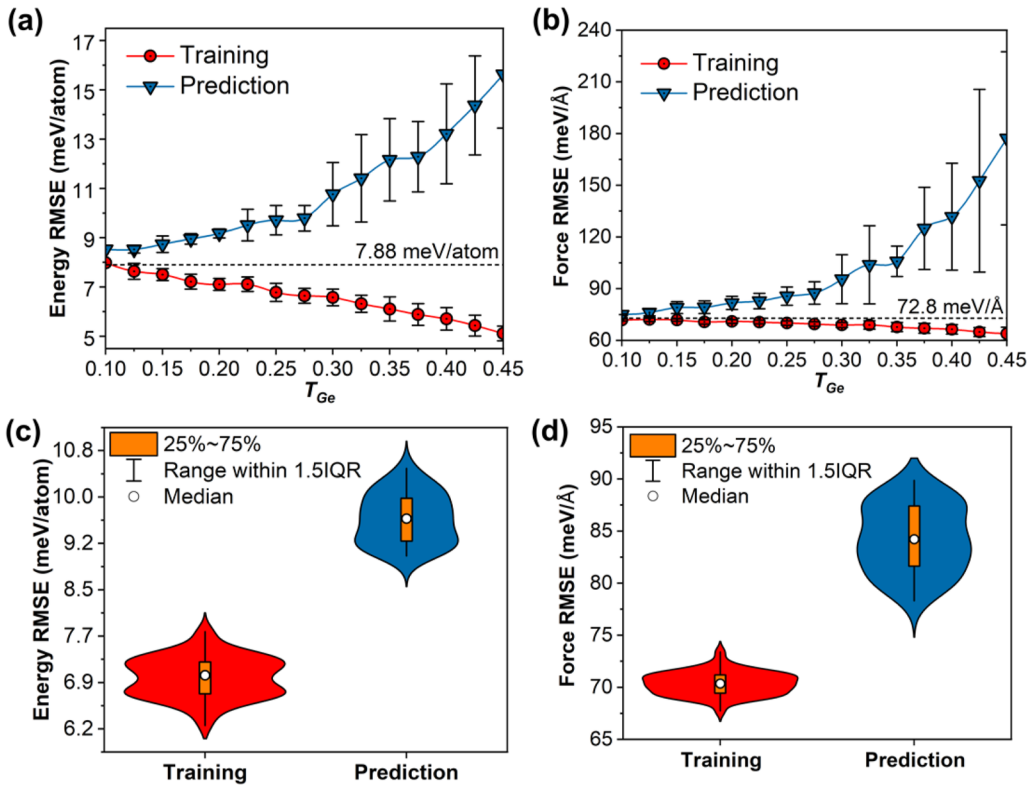

FIG. 3. Variation of (a) energy and (b) force RMSEs for training (red) and prediction (blue) datasets with the threshold value (T_{Ge}) employed in the structure screening process. Violin plots illustrating the distribution of (c) energy and (d) force RMSEs obtained from 50 independent runs with training images selected at T_{Ge} = 0.275. The red and blue colors represent the corresponding distribution for the training and prediction datasets, respectively.

oped when $T_{Ge} \leq 0.275$ (Fig. S2). We also note that the variance in both the energy and force RMSEs, in this region ($T_{Ge} \leq 0.275$), is smaller.

Based on these findings, we identified a threshold T_{Ge} value of 0.275. At this threshold, 1264 structures with 134 882 atoms were selected, which is less than 20% of the original dataset. The MLP at this threshold value has energy and force RMSEs of 9.78 meV/atom and 87.4 meV/Å, respectively, comparable to 7.88 meV/atom and 72.8 meV/Å obtained by the MLP model trained with the original dataset. Figures 3(c) and 3(d) show the distribution of the energy and force RMSEs obtained from the model developed with T_{Ge} = 0.275. Both energy and force RMSEs for the training and prediction follow normal distributions with a small variance. The significant reduction in the number of atomic structures in the training data improves the training process.

B. Performance test with the $Pd_{13}H_2$ nanoparticle system

We have successfully demonstrated the effectiveness of our algorithm in selecting representative structures for the Ge system. To further validate the universality of the algorithm, we conducted a performance test using a two-element system: a $Pd_{13}H_2$ nanoparticle system, for which we have reported a well-established MLP model with optimized fingerprints. Here, we employed two threshold values, namely T_{Pd} and T_H, to treat the local environments of Pd and H separately. Utilizing a grid search method, we explored the variation in the number of selected structures with different T_{Pd} and T_H. The results presented in Fig. 4 clearly show the dependence of the total number of structures selected on both T_{Pd} and T_H. We observed a continuous decrease in the total number of selected structures as both T_{Pd} and T_H increased, which shows that the algorithm effectively modulates the number of selected structures.

We employed the same procedure used for the Ge system to develop an MLP model with selected structures at each T_{Pd} and T_H values and calculated the energy and force RMSEs for both the training and prediction datasets. Figure 5 shows contour maps illustrating the variations of energy and force RMSEs with T_{Pd} and T_H for both training and prediction datasets. For reference, an energy RMSE of 8.6 meV/atom and a force RMSE of 100 meV/Å from the MLP model trained with the original dataset (no screening process was performed) are labeled in the contour map. The energy RMSE follows a similar trend as observed in the Ge system. In the train-

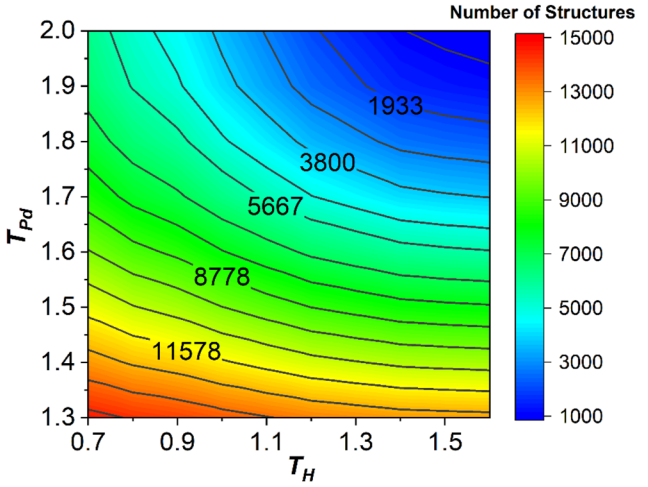

FIG. 4. Number of structures and atoms in the training data bank as a function of the threshold values (T_{Pd} and T_H) employed in the structure screening process.

ing set, the largest RMSE is found in the left-bottom corner with small T_{Pd} and T_H values, and the smallest RMSE is observed in the right-top corner associated with large T_{Pd} and T_H values [Fig. 5(a)], indicating that a reduction in training data leads to better convergence of the model. This situation is reversed in the prediction set due to the loss of certain local environments at high T_{Pd} and T_H values [Fig. 5(b)]. Note that the performance of the ML model is less sensitive to the T_H parameter when T_{Pd} is small, which is attributed to the fact that the H–H interaction becomes negligible beyond a distance of 1.5 Å.

The force RMSE in the training set exhibits a small variation with threshold values, typically in a range of 10 meV/Å [Fig. 5(c)]. However, in the prediction set, we observe an increase in the force RMSE from the left-bottom to the right-top corner, following the diagonal line [Fig. 5(d)]. This trend follows what we observed in the Ge system. Based on this variation trend, we identified a critical line (yellow dotted line) in Fig. 5(d) that denotes the suitable T_{Pd} and T_H values for filtering representative structures in the development of the MLP model for the $Pd_{13}H_2$ system. The MLP model developed using structures selected along the critical line demonstrates a performance comparable to that achieved with the model obtained from the original dataset (Fig. S3). A further increase in the T_{Pd} and T_H values result in a significantly inferior performance of the MLP model in terms of prediction accuracy. These findings show that a selection of 4000 representative structures along the critical line is sufficient to capture the characteristic information contained in the original dataset of 19 802 structures, underscoring the importance of preprocessing the dataset.

C. Comparison with FPS and clustering algorithms

To evaluate the efficiency of our algorithm in selecting representative structures, we compared our method with the furthest point sampling (FPS) method and the clustering algorithm, both of which are widely used methods for extracting representative data from a database. FPS relies on the definition of the similarity between atomic structures. In this study, we utilized both the average structural kernel (Avg-kernel) and the regularized entropy match kernel (REMatch-kernel) as defined in Ref. 68 to assess structural similarity. We computed the structural similarity matrix (K) using the DScribe package,[87] which is based on SOAP descriptors. For Avg-kernel and REMatch-kernel, we set the number of radial basis functions (RBFs) and the maximum degree of spherical harmonic functions to 4 and 3, respectively, (denoted as 4,3-SOAP). This choice ensures that the number of SOAP and the number of BP symmetric functions are equal. It is worth noting that a further increase in the number of RBFs and the maximum degree of spherical harmonics does not yield improvement in the performance (Fig. S4). Other input settings were left at their default values. Subsequently, the similarity matrix was used to determine the distance between each pair of structures ($\tilde{d}(A,B)$),

$$\tilde{d}(A,B) = \sqrt{2 - 2 \times K(A,B)}. \tag{7}$$

The agglomerative clustering algorithm implemented in the Scikit-learn package[88] was used to divide the dataset into a given number of clusters based on structure similarity. A random structure was selected from each cluster as a representative structure. We used the same BP symmetry functions as described in Sec. III A and utilized the structure global descriptor [$\overline{\phi}$ (A)] defined in Eq. (8) to represent the global features of structures.[69] The Euclidean distance between two structures was used to evaluate the similarity of two structures,

$$\overline{\phi}(\mathrm{A}) = \frac{\sum_i^n G_i}{n} \quad (n \text{ is the number of atoms in Structure A}). \tag{8}$$

In the FPS, the first two data points were determined using the algorithm detailed in Ref. 84. The FPS process was iterated until a predetermined number of structures had been chosen. The machine-learning force field trained with the selected structures was employed to compute the RMSEs for energy and force within the prediction datasets. Figure 6 shows the variation of the RMSE values for energy and force with the number of structures selected for the Ge system. For the PdH system, a comparison between the cluster algorithm and our approach is presented in Fig. S5. Additionally, we also conducted a comparison of our algorithm with the CUR decomposition method, which exhibits an inferior performance for both the Ge and PdH systems (see details in Fig. S6 and Part II of the supplementary material).

As shown in Fig. 6, our algorithm demonstrates a superior performance in both accuracy and robustness compared to the FPS method. When selecting the same number of structures, the machine-learning model generated with our method exhibits comparable RMSE values in energy and lower RMSE values in force [Figs. 6(a) and 6(b)]. Moreover, the FPS method exhibits a noticeable fluctuation in RMSE values for both energy and force [blue and orange lines in Figs. 6(a) and 6(b)], indicating its instability. The clustering algorithm [green lines in Figs. 6(a) and 6(b)] exhibits an improved performance in both accuracy and robustness compared to FPS methods, yet slightly inferior than our algorithm for the Ge system. However, a larger variation is observed in the PdH system (Fig. S5). These findings indicate that our algorithm

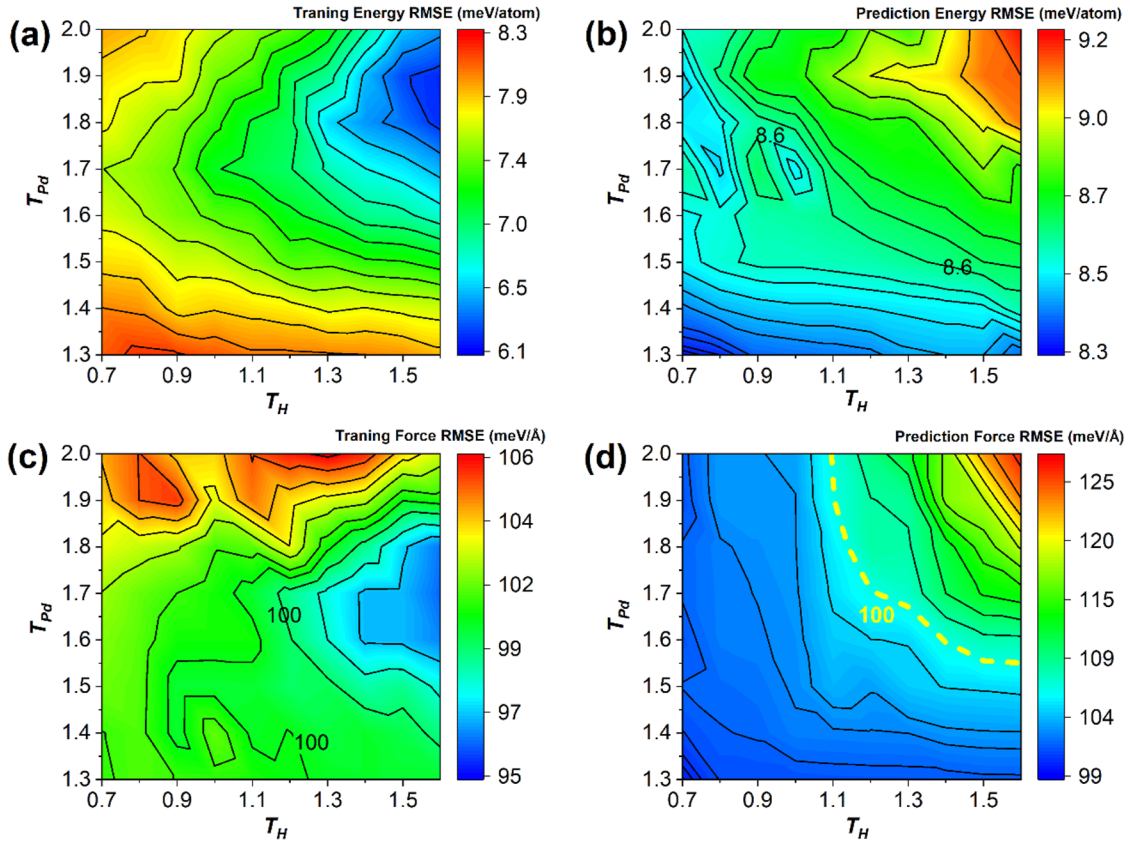

FIG. 5. (a) and (b) Contour plots of energy RMSE as a function of the threshold values T_{Pd} and T_H for (a) training and (b) prediction datasets. (c) and (d) Contour plots of force RMSE as a function of T_{Pd} and T_H for (c) training and (d) prediction datasets.

is more effective in capturing the correlations within the high-dimensional space, whereas the FPS and clustering algorithms, relying on the global features, shows limitation in handling the intricacies of the data. This limitation can be attributed to the loss of local details during the construction of the global similarity matrix.

We compared the computational efficiency of our algorithm with the Avg-kernel FPS, the REMatch-kernel FPS, and the clustering algorithm. Figure 6(c) shows a correlation between the computation cost and the number of representative structures selected using the Avg-kernel FPS, the REMatch-kernel FPS, the clustering algorithm, and our algorithm. Our algorithm offers a substantial reduction in computational cost when compared to both FPS methods. This efficiency can be further enhanced through the utilization of matrix operations with PyTorch tensors, making it comparable with the clustering algorithm. For example, when the number of sampling structures is set to 1264 (the critical point identified by our algorithm), the Avg-kernel FPS and the REMatch-kernel FPS methods require 14 648 and 17 800 s, respectively, with 97% of the cost attributed to the construction of the similarity matrix. In contrast, our algorithm using NumPy takes only 1859 s. By using PyTorch for matrix operations, the computational cost can be further lowered to 55 s, and it exhibits a near-linear dependency on the number of sampling structures, making it suitable for large datasets.

The efficiency of our algorithm stems from its focus on maintaining a local environment bank during the selection process, effectively excluding non-representative local environments from subsequent calculations. The computational complexity of our algorithm is expressed as $O(mnN)$, where "m," "n," and "N" represent the number of local environments in the bank, the number of atoms in a structure, and the number of structures in the dataset, respectively. It is important to note that if no local environment were excluded, m is equal to nN and it decreases when more local environments are excluded. This implies that the computational cost decreases as more structures are excluded, as observed in Fig. 6(c). In contrast, the construction of a global similarity matrix for the FPS algorithm requires computing similarities between every pair of two atoms in two structures, leading to a significantly higher complexity of $O(N^2n^2)$ compared to the $O(mnN)$ complexity of our approach.

While our algorithm significantly reduces computational complexity, it currently does not demonstrate linear scaling with the number of sampled structures and it could face computational

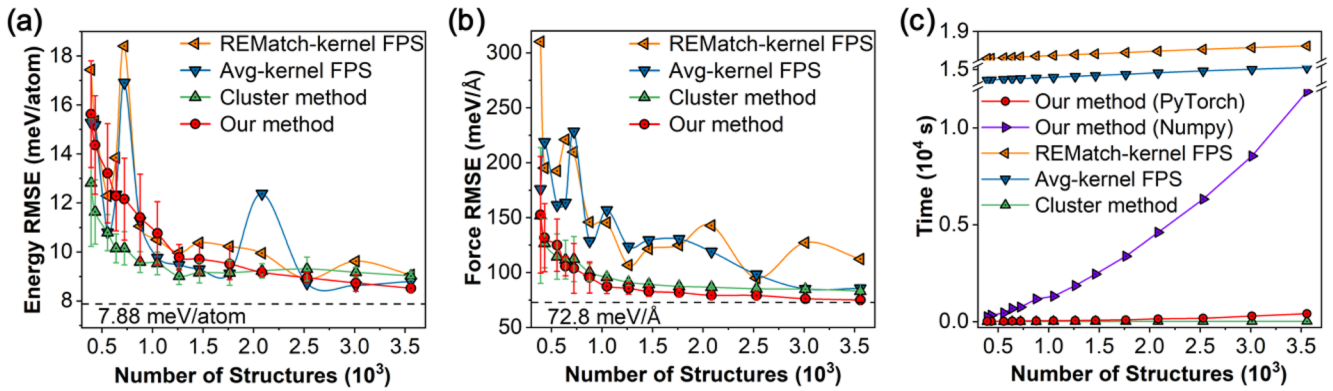

FIG. 6. Variation of (a) energy and (b) force RMSEs in prediction datasets with the number of selected structures, obtained through our algorithm (red), the Avg-kernel FPS (blue), the clustering algorithm (green), and the REMatch-kernel FPS (orange). (c) Variation of execution time with the number of selected structures for all four algorithms (tested on a Xeon Platinum 8160 CPU using 1 CPU core). Note that the time used for the computation of SOAP and BP symmetry function was not included.

challenges when dealing with a massive number of selected local environments (fingerprint vectors in the local environment bank). Implementing a grouping mechanism for these fingerprint vectors could be a viable strategy to further improve the computational efficiency. Furthermore, considering a sequential combination of the clustering algorithm with our approach might offer an alternative to improve the efficiency, while upholding high accuracy and robustness.

IV. CONCLUSIONS

In this study, we developed a training data selection algorithm based on a local-environment bank, which stores high-dimensional vectors representing the local environments of atoms. The algorithm utilizes a threshold value to assess the similarity between two local environments. A new local environment is identified when it has a larger-than-threshold Euclidean distance relative to all those present in the bank. The corresponding structure is then selected as the new structure. We applied this algorithm recursively to select structures from a database of atomic structures, for the Ge and $Pd_{13}H_2$ systems, respectively. Our results show that the performance of the MLP model highly depends on the threshold value employed in the screening process. By identifying a suitable threshold value, we significantly reduced the training data size, while maintaining the performance of the MLP model. Specifically, the training data were reduced from 7506 to 1264 structures for the Ge system and from 19 802 to 4000 structures for the $Pd_{13}H_2$ system. This reduction significantly improves the training efficiency of the MLP without sacrificing the accuracy. Furthermore, we verified the generality of this algorithm, demonstrating that the results are independent of the selection and ordering of the initial structures. Our algorithm also exhibits a superior performance when compared to the performances of FPS and the clustering method. These results enhance our understanding of how the performance of the MLP model relies on the training data and potentially expedite the development of ML potentials.

In our algorithm, the local environment bank can be dynamically updated without re-evaluating the training data bank in real time. This capability holds promise as an efficient data storage mechanism for constructing accurate machine learning potentials. The effective threshold measures distances within the high-dimensional vector space of the structural configurations. Further investigation into system dependency could enhance the generalizability of our algorithm and aid in uncertainty estimation of the MLP model. Nevertheless, in its current version, users will need a pre-determined dataset and an appropriate threshold, which may pose a challenge to applying our algorithm. This challenge is also a common issue in this field and requires resolution. Additionally, our algorithm solely focuses on representatives of local environments within a specific cutoff, excluding global features. As a result, it may not be suitable for cases where global features significantly impact the MLP accuracy.

SUPPLEMENTARY MATERIAL

See the supplementary material for additional details about parameters used for MLP development, distributions of symmetry functions, and a comparison of predicted energies and forces with reference DFT values. Codes for the FPS, CUR decomposition, clustering algorithms, and the local-environment-guided selected algorithm can be found at GitHub (https://gitlab.com/Li-Renzhe/pyimg).

ACKNOWLEDGMENTS

This work was supported by the National Key R & D Program of China (Grant No. 2022YFA1503102), the National Natural Science Foundation of China (Grant No. 92270103), the Shenzhen Key Laboratory of Micro/Nano-Porous Functional Materials (SKLPM) (Grant No. ZDSYS20210709112802010), and Shenzhen fundamental research funding (Grant No. JCYJ20210324115809026). The

theoretical calculations were supported by the Center for Computational Science and Engineering of Southern University of Science and Technology. Work at the University of Texas was supported by the National Science Foundation (Grant No. CHE-2102317) and the Texas Advanced Computing Center.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

R.L. and C.Z. contributed equally to this paper.

Renzhe Li: Formal analysis (equal); Investigation (equal); Methodology (equal); Writing – original draft (equal). **Chuan Zhou**: Formal analysis (equal); Investigation (equal); Methodology (equal); Writing – review & editing (equal). **Akksay Singh**: Investigation (supporting); Writing – review & editing (supporting). **Yong Pei**: Writing – review & editing (supporting). **Graeme Henkelman**: Conceptualization (equal); Funding acquisition (equal); Project administration (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal). **Lei Li**: Conceptualization (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

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

REFERENCES

[1]J. Behler, "Four generations of high-dimensional neural network potentials," Chem. Rev. **121**, 10037 (2021).
[2]R. C. Bernardi, M. C. Melo, and K. Schulten, "Enhanced sampling techniques in molecular dynamics simulations of biological systems," Biochim. Biophys. Acta, Gen. Subj. **1850**, 872 (2015).
[3]F. Musil *et al.*, "Physics-inspired structural representations for molecules and materials," Chem. Rev. **121**, 9759 (2021).
[4]O. T. Unke *et al.*, "Machine learning force fields," Chem. Rev. **121**, 10142 (2021).
[5]C. M. Handley and P. L. Popelier, "Potential energy surfaces fitted by artificial neural networks," J. Phys. Chem. A **114**, 3371 (2010).
[6]S. Manzhos and T. Carrington, Jr., "Neural network potential energy surfaces for small molecules and reactions," Chem. Rev. **121**, 10187 (2020).
[7]J. Behler, "Constructing high-dimensional neural network potentials: A tutorial review," Int. J. Quantum Chem. **115**, 1032 (2015).
[8]T. W. Ko and S. P. Ong, "Recent advances and outstanding challenges for machine learning interatomic potentials," Nat Comput Sci **3**, 998 (2023).
[9]N. L. Allinger, Y. H. Yuh, and J. H. Lii, "Molecular mechanics. The MM3 force field for hydrocarbons. 1," J. Am. Chem. Soc. **111**, 8551 (1989).
[10]W. D. Cornell *et al.*, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," J. Am. Chem. Soc. **117**, 5179 (1995).
[11]T. J. Lenosky *et al.*, "Highly optimized empirical potential model of silicon," Modell. Simul. Mater. Sci. Eng. **8**, 825 (2000).
[12]A. C. Van Duin *et al.*, "ReaxFF: A reactive force field for hydrocarbons," J. Phys. Chem. A **105**, 9396 (2001).
[13]W. Han, "Molecular modeling by machin learning," Math. Numer. Sin. **43**, 261 (2021).
[14]J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).
[15]K. T. Schütt *et al.*, "Quantum-chemical insights from deep tensor neural networks," Nat. Commun. **8**, 13890 (2017).
[16]A. P. Bartók *et al.*, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).
[17]O. T. Unke and M. Meuwly, "PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges," J. Chem. Theory Comput. **15**, 3678 (2019).
[18]L. Zhang *et al.*, "End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems," Adv. Neural Inf. Process. Syst. **31**, 4436 (2018).
[19]A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," Multiscale Model Simul. **14**, 1153 (2016).
[20]J. S. Smith, O. Isayev, and A. E. Roitberg, "ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost," Chem. Sci. **8**, 3192 (2017).
[21]K. Yao *et al.*, "The TensorMol-0.1 model chemistry: A neural network augmented with long-range physics," Chem. Sci. **9**, 2261 (2018).
[22]S. Chmiela *et al.*, "Machine learning of accurate energy-conserving molecular force fields," Sci. Adv. **3**, e1603015 (2017).
[23]S. Chmiela *et al.*, "Towards exact molecular dynamics simulations with machine-learned force fields," Nat. Commun. **9**, 3887 (2018).
[24]L. Zhang *et al.*, "Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics," Phys. Rev. Lett. **120**, 143001 (2018).
[25]M. Rupp *et al.*, "Fast and accurate modeling of molecular atomization energies with machine learning," Phys. Rev. Lett. **108**, 058301 (2012).
[26]J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," J. Chem. Phys. **134**, 074106 (2011).
[27]V. Botu *et al.*, "Machine learning force fields: Construction, validation, and outlook," J. Phys. Chem. C **121**, 511 (2017).
[28]G. P. Pun *et al.*, "Physically informed artificial neural networks for atomistic modeling of materials," Nat. Commun. **10**, 2339 (2019).
[29]W. Jia *et al.*, in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis* (IEEE, 2020), p. 1.
[30]L. Li *et al.*, "Pair-distribution-function guided optimization of fingerprints for atom-centered neural network potentials," J. Chem. Phys. **152**, 224102 (2020).
[31]A. Khorshidi and A. A. Peterson, "*Amp*: A modular approach to machine learning in atomistic simulations," Comput. Phys. Commun. **207**, 310 (2016).
[32]N. Artrith and A. Urban, "An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO_2," Comput. Mater. Sci. **114**, 135 (2016).
[33]N. Artrith and J. Behler, "High-dimensional neural network potentials for metal surfaces: A prototype study for copper," Phys. Rev. B **85**, 045439 (2012).
[34]I. A. Basheer and M. Hajmeer, "Artificial neural networks: Fundamentals, computing, design, and application," J. Microbiol. Methods **43**, 3 (2000).
[35]A. P. Bartók and G. Csányi, "Gaussian approximation potentials: A brief tutorial introduction," Int. J. Quantum Chem. **115**, 1051 (2015).
[36]V. L. Deringer *et al.*, "Gaussian process regression for materials and molecules," Chem. Rev. **121**, 10073 (2021).
[37]C. Chen *et al.*, "Graph networks as a universal machine learning framework for molecules and crystals," Chem. Mater. **31**, 3564 (2019).
[38]T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," Phys. Rev. Lett. **120**, 145301 (2018).
[39]A. Hajibabaei, C. W. Myung, and K. S. Kim, "Sparse Gaussian process potentials: Application to lithium diffusivity in superionic conducting solid electrolytes," Phys. Rev. B **103**, 214102 (2021).
[40]A. Hajibabaei and K. S. Kim, "Universal machine learning interatomic potentials: Surveying solid electrolytes," J. Phys. Chem. Lett. **12**, 8115 (2021).

[41] V. L. Deringer *et al.*, "Origins of structural and electronic transitions in disordered silicon," Nature **589**, 59 (2021).
[42] V. Kapil *et al.*, "The first-principles phase diagram of monolayer nanoconfined water," Nature **609**, 512 (2022).
[43] L. Zhang *et al.*, "Phase diagram of a deep potential water model," Phys. Rev. Lett. **126**, 236001 (2021).
[44] M. F. Calegari Andrade *et al.*, "Free energy of proton transfer at the water–TiO_2 interface from *ab initio* deep potential molecular dynamics," Chem. Sci. **11**, 2335 (2020).
[45] J. Byggmästar, K. Nordlund, and F. Djurabekova, "Gaussian approximation potentials for body-centered-cubic transition metals," Phys. Rev. Mater. **4**, 093802 (2020).
[46] A. Hajibabaei *et al.*, "Machine learning of first-principles force-fields for alkane and polyene hydrocarbons," J. Phys. Chem. A **125**, 9414 (2021).
[47] M. Ha *et al.*, "Al-doping driven suppression of capacity and voltage fadings in 4d-element containing Li-ion-battery cathode materials: Machine learning and density functional theory," Adv. Energy Mater. **12**, 2201497 (2022).
[48] A. P. Bartók *et al.*, "Machine learning a general-purpose interatomic potential for silicon," Phys. Rev. X **8**, 041048 (2018).
[49] E. V. Podryabinkin and A. V. Shapeev, "Active learning of linearly parametrized interatomic potentials," Comput. Mater. Sci. **140**, 171 (2017).
[50] R. Ramakrishnan *et al.*, "Quantum chemistry structures and properties of 134 kilo molecules," Sci. Data **1**, 140022 (2014).
[51] L. Ruddigkeit *et al.*, "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17," J. Chem. Inf. Model. **52**, 2864 (2012).
[52] G. Mills and H. Jónsson, "Quantum and thermal effects in H_2 dissociative adsorption: Evaluation of free energy barriers in multidimensional quantum systems," Phys. Rev. Lett. **72**, 1124 (1994).
[53] G. Mills, H. Jónsson, and G. K. Schenter, "Reversible work transition state theory: Application to dissociative adsorption of hydrogen," Surf. Sci. **324**, 305 (1995).
[54] M. Born and W. Heisenberg, "Zur quantentheorie der molekeln," Ann. Phys. **379**, 1 (1924).
[55] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," Phys. Rev. **136**, B864 (1964).
[56] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," Phys. Rev. **140**, A1133 (1965).
[57] C. Møller and M. S. Plesset, "Note on an approximation treatment for many-electron systems," Phys. Rev. **46**, 618 (1934).
[58] R. Car and M. Parrinello, "Unified approach for molecular dynamics and density-functional theory," Phys. Rev. Lett. **55**, 2471 (1985).
[59] G. Kresse and J. Hafner, "*Ab initio* molecular dynamics for liquid metals," Phys. Rev. B **47**, 558 (1993).
[60] Y. I. Yang *et al.*, "Efficient sampling over rough energy landscapes with high barriers: A combination of metadynamics with integrated tempering sampling," J. Chem. Phys. **144**, 094105 (2016).
[61] C. Abrams and G. Bussi, "Enhanced sampling in molecular dynamics using metadynamics, replica-exchange, and temperature-acceleration," Entropy **16**, 163 (2013).
[62] Y. Zhang and G. A. Voth, "Combined metadynamics and umbrella sampling method for the calculation of ion permeation free energy profiles," J. Chem. Theory Comput. **7**, 2277 (2011).
[63] A. Barducci, M. Bonomi, and M. Parrinello, "Metadynamics," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **1**, 826 (2011).
[64] W. Jeong *et al.*, "Toward reliable and transferable machine learning potentials: Uniform training by overcoming sampling bias," J. Phys. Chem. C **122**, 22790 (2018).
[65] Y. Zuo *et al.*, "Performance and cost assessment of machine learning interatomic potentials," J. Phys. Chem. A **124**, 731 (2020).
[66] W. Kohn, A. D. Becke, and R. G. Parr, "Density functional theory of electronic structure," J. Phys. Chem. **100**, 12974 (1996).
[67] E. J. Baerends and O. V. Gritsenko, "A quantum chemical view of density functional theory," J. Phys. Chem. A **101**, 5383 (1997).
[68] S. De *et al.*, "Comparing molecules and solids across structural and alchemical space," Phys. Chem. Chem. Phys. **18**, 13754 (2016).
[69] G. Imbalzano *et al.*, "Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials," J. Chem. Phys. **148**, 241730 (2018).
[70] T. D. Huan *et al.*, "A universal strategy for the creation of machine learning-based atomistic force fields," npj Comput. Mater. **3**, 37 (2017).
[71] M. O. Jäger *et al.*, "Machine learning hydrogen adsorption on nanoclusters through structural descriptors," npj Comput. Mater. **4**, 37 (2018).
[72] V. Botu and R. Ramprasad, "Learning scheme to predict atomic forces and accelerate materials simulations," Phys. Rev. B **92**, 094306 (2015).
[73] V. Botu and R. Ramprasad, "Adaptive machine learning framework to accelerate *ab initio* molecular dynamics," Int. J. Quantum Chem. **115**, 1074 (2015).
[74] V. Botu, J. Chapman, and R. Ramprasad, "A study of adatom ripening on an Al (1 1 1) surface with machine learning force fields," Comput. Mater. Sci. **129**, 332 (2017).
[75] T. Suzuki, R. Tamura, and T. Miyazaki, "Machine learning for atomic forces in a crystalline solid: Transferability to various temperatures," Int. J. Quantum Chem. **117**, 33 (2017).
[76] G. Pilania *et al.*, "Accelerating materials property predictions using machine learning," Sci. Rep. **3**, 2810 (2013).
[77] T. D. Huan, A. Mannodi-Kanakkithodi, and R. Ramprasad, "Accelerated materials property predictions and design using motif-based fingerprints," Phys. Rev. B **92**, 014106 (2015).
[78] A. Mannodi-Kanakkithodi *et al.*, "Machine learning strategy for accelerated design of polymer dielectrics," Sci. Rep. **6**, 20952 (2016).
[79] A. P. Bartók *et al.*, "Machine learning unifies the modeling of materials and molecules," Sci. Adv. **3**, e1701816 (2017).
[80] P. Friederich *et al.*, "Machine-learned potentials for next-generation matter simulations," Nat. Mater. **20**, 750 (2021).
[81] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," Adv. Neural Inf. Process. Syst. **30**, 6402 (2017).
[82] Y. Yang, O. A. Jiménez-Negrón, and J. R. Kitchin, "Machine-learning accelerated geometry optimization in molecular simulation," J. Chem. Phys. **154**, 234704 (2021).
[83] Q. Lin *et al.*, "Searching configurations in uncertainty space: Active learning of high-dimensional neural network reactive potentials," J. Chem. Theory Comput. **17**, 2691 (2021).
[84] N. Xu *et al.*, "Training data set refinement for the machine learning potential of Li-Si alloys via structural similarity analysis," arXiv:2103.04347 (2021).
[85] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," Ann. Statist. **36**, 1171 (2008).
[86] L. Li *et al.*, "'Atom-centered machine-learning force field package," Comput. Phys. Commun. **292**, 108883 (2023).
[87] L. Himanen *et al.*, "DScribe: Library of descriptors for machine learning in materials science," Comput. Phys. Commun. **247**, 106949 (2020).
[88] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res. **12**, 2825 (2011).