

RESEARCH ARTICLE | FEBRUARY 09 2022

## A nearsighted force-training approach to systematically generate training data for the machine learning of large atomic structures

Special Collection: [Chemical Design by Artificial Intelligence](#)

Cheng Zeng ; Xi Chen ; Andrew A. Peterson  



*J. Chem. Phys.* 156, 064104 (2022)

<https://doi.org/10.1063/5.0079314>

 CHORUS



CrossMark

### Articles You May Be Interested In

Inverted images and noninverted shadows

*The Physics Teacher* (December 2001)

Eyeglasses in the Classroom

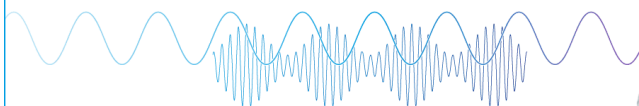
*The Physics Teacher* (January 2010)

Reproductions of Prints, Drawings, and Paintings of Interest in the History of Physics. 64. The First Appearance of Concave Eyeglasses

*American Journal of Physics* (November 1954)

Webinar

### Boost Your Signal-to-Noise Ratio with Lock-in Detection



Sep. 7th – Register now



# A nearsighted force-training approach to systematically generate training data for the machine learning of large atomic structures

Cite as: J. Chem. Phys. 156, 064104 (2022); doi: 10.1063/5.0079314

Submitted: 19 November 2021 • Accepted: 26 January 2022 •

Published Online: 9 February 2022



View Online



Export Citation



CrossMark

Cheng Zeng,  Xi Chen,  and Andrew A. Peterson<sup>a)</sup> 

## AFFILIATIONS

School of Engineering, Brown University, Providence, Rhode Island 02912, USA

**Note:** This paper is part of the JCP Special Topic on Chemical Design by Artificial Intelligence.

<sup>a)</sup> Author to whom correspondence should be addressed: [andrew\\_peterson@brown.edu](mailto:andrew_peterson@brown.edu)

## ABSTRACT

A challenge of atomistic machine-learning (ML) methods is ensuring that the training data are suitable for the system being simulated, which is particularly challenging for systems with large numbers of atoms. Most atomistic ML approaches rely on the nearsightedness principle (“all chemistry is local”), using information about the position of an atom’s neighbors to predict a per-atom energy. In this work, we develop a framework that exploits the nearsighted nature of ML models to systematically produce an appropriate training set for large structures. We use a per-atom uncertainty estimate to identify the most uncertain atoms and extract chunks centered around these atoms. It is crucial that these small chunks are both large enough to satisfy the ML’s nearsighted principle (that is, filling the cutoff radius) and are large enough to be converged with respect to the electronic structure calculation. We present data indicating when the electronic structure calculations are converged with respect to the structure size, which fundamentally limits the accuracy of any nearsighted ML calculator. These new atomic chunks are calculated in electronic structures, and crucially, only a single force—that of the central atom—is added to the growing training set, preventing the noisy and irrelevant information from the piece’s boundary from interfering with ML training. The resulting ML potentials are robust, despite requiring single-point calculations on only small reference structures and never seeing large training structures. We demonstrated our approach via structure optimization of a 260-atom structure and extended the approach to clusters with up to 1415 atoms.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0079314>

## I. INTRODUCTION

Reliable atomistic simulations depend on the accuracy of the *ab initio* methods used. In practice, Kohn–Sham density functional theory (DFT) is probably the most widely used method for electronic structure calculations. DFT calculations scale as approximately  $\mathcal{O}(N^3)$ ,<sup>1</sup> where  $N$  is an indicator of the system scale, such as the number of atoms, the number of electrons, or the number of basis functions. Higher-accuracy methods also exist but with less favorable computational scaling; methods also exist with more favorable scaling than DFT, but trade-off calculation accuracy. All widely used electronic-structure methods exist on such a Pareto frontier, where

one chooses the appropriate trade-off between accuracy and speed for their particular application.

In recent years, atomistic machine learning (ML) has demonstrated its capability in fitting potential-energy surfaces (PES’s) created by electronic structure codes, while reducing the computational cost by orders of magnitude.<sup>2–5</sup> Typical ML codes can scale with  $\mathcal{O}(N^1)$ ,<sup>6</sup> and, in principle, their accuracy can be made to mirror that of the parent calculator from which they learn. The performance of ML potentials strongly depends on both the ML method and how we encode atomic configurations, i.e., the descriptors used to generate features for the ML method.<sup>7</sup> Many ML algorithms have proven effective and efficient in fitting PES’s,

including the Gaussian process,<sup>4</sup> kernel ridge regression,<sup>5,8</sup> and neural networks.<sup>2,3,9–11</sup> A wide variety of descriptors have been established, such as smooth overlap of atomic positions,<sup>12</sup> Coulomb matrices,<sup>5</sup> atom-centered symmetry functions,<sup>13</sup> and Zernike descriptors.<sup>11</sup> However, the accuracy and range of applicability are always limited by the training data chosen, which come from electronic structure calculations.

In principle, ML methods' favorable scaling can allow such models to make predictions on large systems that would be unaffordable to the parent electronic structure calculator. However, this also can create a problem, as it is impractical to create training data at sizes that are comparable to the machine-learning predictions. Instead, one typically needs to rely on smaller-sized training structures to create robust ML models that can be used to make predictions for these larger structures. However, it is often not straightforward to understand what small-scale images need to be included for large-scale model predictions, and in this work, we provide a systematic framework to enable training-data collection, in an active-learning scheme.

Normally, a ML model needs a large amount of data from high-level *ab initio* calculations. Once the model is trained on the data, it can give accurate predictions in regions where there are abundant data, while sometimes performing catastrophically in regions where training data are scarce. Active learning schemes are recently receiving much attention to minimize the number of expensive *ab initio* calculations, focusing instead on providing the “right” training data for the prediction at hand.<sup>14–20</sup> The key to a successful active learning scheme is a proper choice of the query strategy—such as an uncertainty estimate<sup>21</sup> or a strategic check of key results in a predictor–corrector fashion<sup>14</sup>—used to detect when the potential moves away from familiar regions. Much prior work motivates the current study. A “learn on the fly” scheme was formulated by Csányi *et al.*<sup>22</sup> for improving classical models: it takes a predictor–corrector-style force fitting on atoms in the quantum region, and forces acting on separate atoms can be computed independently by carving out a small cluster centered on a given atom. A query-by-bagging method was suggested by Artrith and Behler:<sup>23</sup> if the predictions by two neural network models differ on a given configuration by more than a threshold, it needs to be calculated at the *ab initio* level and added to the database. Peterson *et al.*<sup>21</sup> demonstrated that the halfspread of a bootstrap ensemble offers a reliable upper bound for the uncertainty of ML predictions and showed that these techniques can lead to per-atom uncertainty, isolating exactly what features in a structure lead the ML model to fail. Bernstein *et al.* suggested a self-guided approach for structure search: a combination of Boltzmann-probability sampling and the leverage-score CUR algorithm is employed to select the few most relevant and diverse structures at each active learning step.<sup>17</sup> The CUR algorithm is a dimensional reduction technique that is based on interpretable low-rank matrix decompositions, as it selects actual columns and/or rows of the data matrix that exert large statistical influences.<sup>24</sup> Zhang *et al.* proposed a procedure that used the maximum standard deviation of the predicted atomic forces as the model deviation indicator to speed up the search.<sup>18</sup>

Although it is well-acknowledged that ML models enable long-time and large-scale simulations, validation of ML predictions without performing expensive *ab initio* calculations on large systems is non-trivial and has received comparatively little investigation.

Moreover, if and when a prediction on a large system is found to be unreliable, it is challenging to sample small relevant reference structures, which can be used to extend the reference database and improve the ML model.

In this work, we present a robust active-learning approach for generating reference databases right from the target structure, exploring the configuration space with minimal effort. Briefly, our approach relies on a per-atom uncertainty prediction to identify which regions of a target structure are lacking in training data. It then extracts “chunks” centered around these uncertain atoms, which are calculated in the parent electronic-structure calculator. Importantly, as these chunks are added to the ML calculator's training set, only the force on the central atom is included in the loss function, avoiding noisy and irrelevant information from the surrounding atoms. The approach we present can be applied to any finite-ranged ML potential given where atomic forces are trained, and a per-atom uncertainty estimate can be made. We examined the efficacy and efficiency of this approach via a structure-optimization calculation, for systems ranging in size from 260 to 1415 atoms.

## II. THEORETICAL APPROACH

A “nearsightedness” principle was put forward by Kohn in the context of electronic structure calculations,<sup>25,26</sup> suggesting that in typical atomic systems, all chemistry is local. That is, isolated perturbations in the electronic structure do not affect the local properties, such as atomic forces, of far-away atoms. Kohn attributed this phenomenon to destructive interference among wave functions. This principle may be violated if long-range charge transfer occurs in the system; for example, interactions with atoms outside the local region have been observed in the study of adsorption of metal clusters on a doped substrate, where long-range charge transfer between the dopant in the substrate and adsorbed metal atoms is crucial in determining the global structure.<sup>27</sup>

Most atomistic ML schemes, such as the seminal Behler–Parrinello scheme,<sup>2</sup> rely on an *ansatz* that the potential energy  $E$  can be decomposed into atomic contributions, i.e.,  $E = \sum_i E_i$ . Each atomic contribution  $E_i$  comes from the atom  $i$  interacting with all neighboring atoms  $j$  within a given cutoff radius  $R_c$ ; that is,  $E_i = E_i(\{R_{ij}\})$ , where  $R_{ij}$  is the relative position vector between atoms  $i$  and  $j$ . In this sense, such ML potentials merely depend on local chemical environments, each of which is represented by a chunk of atoms centered on the atom of interest. When Kohn's nearsightedness principle applies, we can expect that ML models can give good representations of the parent data. This is the intrinsic limitation for all finite-ranged ML potentials. In this work, we aim to exploit this limitation to build an efficient training scheme for methods that operate under this *ansatz*.

In earlier work,<sup>21</sup> we showed that it is possible not only to predict the uncertainty on an ML model's prediction but also to identify the specific atoms most responsible for the uncertainty. In that work, a bootstrap ensemble of calculators provided independent estimates of the per-atom energy and force; the variance in the calculators' predictions could then be used as an indicator of the most uncertain atoms in the structure. Using such an approach, we can examine a

large structure and detect which atoms should be refined with more training data.

In principle, just these particular atoms—along with their neighbors within the ML cutoff radius ( $R_c$ )—would need to be extracted from the structure due to the nearsighted nature of the ML model. For typical values of  $R_c$ , this would result in chunks of about 10–20 atoms, which are trivial to calculate in electronic structure codes such as DFT. After calculation, each chunk could be added to the training set, increasing the certainty of the prediction for the central atom. However, since the neighboring atoms will be in irrelevant environments in this small structure—for example, they may be surrounded by vacuum in the region outside  $R_c$ —training to the energy of such structures would result in adding overwhelmingly irrelevant data to the training set; that is, roughly 10–20 times more irrelevant data (corresponding to the 10–20 surrounding atoms) than relevant data (corresponding to the central, uncertain atom). This increases the training difficulty and will reduce the prediction accuracy of the relevant atoms. It would be better if the ML model's training set could be expanded by *only* data involving the central atom, and not the surrounding atoms.

Unfortunately, there are no well-defined atomic energies in DFT—although various schemes to decompose the total energy have been proposed over the years. For example, Blanco *et al.* proposed a scheme in which the atomic energy of an atom is ascribed to intra- and inter-atomic components.<sup>28</sup> Wang *et al.* presented an energy-partition scheme for DFT based on local energy densities and isolated atom charge density.<sup>29</sup> However, it is difficult to achieve a desired energy partition, and in most cases, the energy partition is not unique and will not necessarily match the decomposition implicitly found by the ML model.

A more straightforward approach is to focus on the forces on the atoms, which are inherently per-atom quantities. Atomic forces and the system potential energy are consistent with one another because forces—or more specifically conservative forces—are simply the negative derivative of potential energies with respect to atomic positions. Therefore, it is possible to learn the potential energy surface from only the forces, except for a constant offset, which corresponds to a constant of integration. So long as some images in the training set are trained to energy, this offset will automatically be learned by the ML algorithm, even if the remainder of the images are trained to only forces. It is noteworthy that forces present much more granular information of the PES, as an  $N$ -atom configuration only has one potential energy, but  $3N$  forces. Such an approach is compatible with either a machine-learning scheme that predicts energies and provides forces as numerical or analytical derivatives, such as the Behler–Parrinello<sup>2</sup> approach, or a scheme that predicts forces directly, such as those from Botu and Ramprasad<sup>30</sup> or Glielmo *et al.*<sup>31</sup> Herein, we use the traditional Behler–Parrinello approach.

Therefore, a plausible scheme is that whenever a new chunk is calculated, only the (three-component) force on the central atom is added to the training set and the rest of the output of the training structure is discarded. Then, the performance of the ML potential can be improved by comparing the DFT and ML forces on the atom whose local environment is of interest.

However, there is an important property of machine-learned forces that we should emphasize, which complicates the implication.

Although it seems counterintuitive, the locality of the force is double the locality of the atomic energy in finite-ranged ML potentials. In other words, forces acting on a central atom are affected by atoms outside of  $R_c$ . This force property has also been noted in previous studies by Artrith and Behler<sup>23</sup> and Bernstein *et al.*<sup>32</sup> It is simplest to understand the longer-range nature of forces in the ML representation if we consider taking a numerical derivative, which converges to the analytical derivative at small enough perturbations. That is, if we perturb a central atom by  $\Delta x$ , its force in the  $x$  direction is approximately  $-\Delta E/\Delta x$ . Perturbing this atom will affect the per-atom energies of atoms up to a distance of  $R_c$  away from this central atom, but each of these atoms' energies is a function of the atoms within  $R_c$  of them. Therefore, the terms in the energy summation that change include all atoms within  $R_c$ , but those atoms' energies are a function of atoms up to  $2R_c$  away from the central atom. Therefore, the force on the central atom is affected by atoms up to a distance of  $2R_c$  away, albeit with severely diminishing effects. Therefore, one must be careful in balancing the cutoff radius of the ML model with the nearsighted radius of the electronic-structure calculations.

Based on the above, we can work out a qualitative scheme for “nearsighted force training” (NFT) for large systems: first, we create a minimal set of training images on which we train an ensemble of ML calculators; next, we predict per-atom forces for our large structure and search for atoms where the models disagree; we then extract “chunks” of the original image, ensuring atomic sizes to be converged with respect to the electronic structure calculation and filling the ML force cutoff radius of  $2R_c$ ; and finally, we add these chunks to our training set but train *only* on the force on the central atom, discarding the extraneous information. This process can be repeated intelligently throughout the atomistic procedure, depending on the goals of the research.

### III. ALGORITHM

Depending on the researcher's goals, there are many logical ways to implement the scheme described above. Here, we focused on a relatively simple scheme designed to systematically reduce the uncertainty on a large target structure's atomic forces, to understand if this approach will work for scalable calculations. We will describe how we expand this basic algorithm to integrate with a structural relaxation in Sec. V C. In this work, we used a finite-ranged neural-network potential as the ML model, full details of which are given in the [supplementary material](#).

A per-atom measure of uncertainty is crucial to the operation of the algorithm. As in earlier work,<sup>21</sup> we used a bootstrap resampling algorithm to create an ensemble of trained ML calculators, which produce a spread of predictions for each ML output. In our earlier work, we focused on uncertainties in energy predictions, for which it sufficed to look at the halfspread of the ensemble models' predictions. However, since each atom's force has a component in each ( $x$ ,  $y$ ,  $z$ ) direction, taking a simple halfspread would produce rotationally variant predictions of uncertainty, which do not match the physics of the problem.

Instead, we developed a rotationally invariant uncertainty metric for the atomic force on atom  $i$  as predicted by an  $M$ -member ensemble,

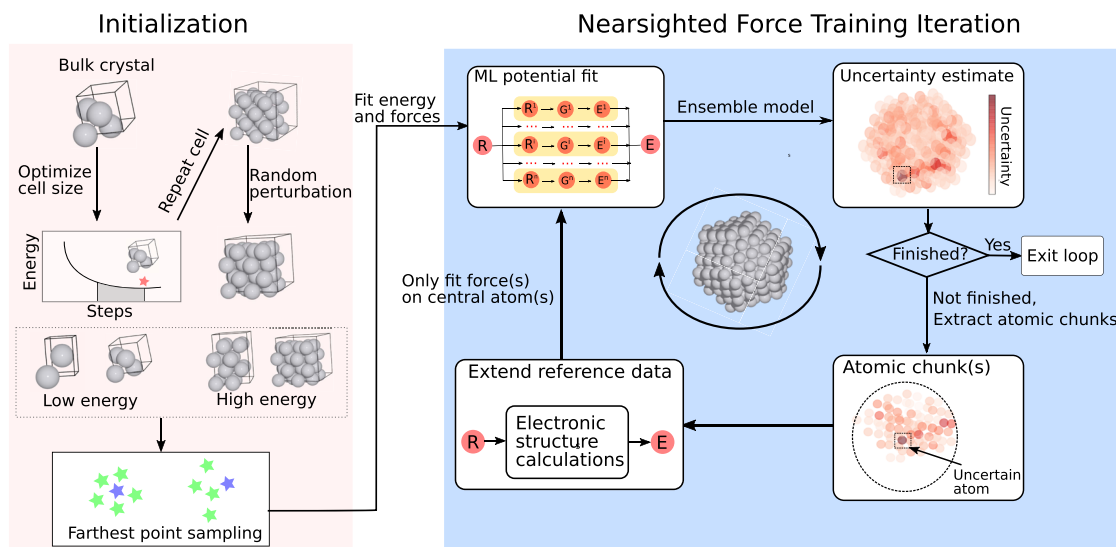


FIG. 1. Schematic representation of the initialization protocol and nearsighted force training iteration.

$$\delta_i = 2.58\sigma_f = 2.58\sqrt{\frac{\sum_{j=1}^M \|\mathbf{f}_i^{(j)} - \bar{\mathbf{f}}_i\|^2}{M-1}}, \quad (1)$$

where  $\mathbf{f}_i^{(j)}$  is the vector force predicted by model  $j$  and  $\bar{\mathbf{f}}_i$  is the average force vector of atom  $i$  over models of the ensemble. This uncertainty metric is simply the standard deviation ( $\sigma_f$ ) multiplied by a constant of 2.58 to capture 99% of the ensemble's variation in force prediction. It turns out to be a reliable estimate of force prediction residuals, which we will show in the Sec. V. We term this the “atomic uncertainty,” and define the “structure uncertainty” as the maximum atomic uncertainty in a given structure, i.e.,  $\delta = \max_i(\{\delta_i\})$ .

Using this definition of uncertainty, the algorithm (also shown in Fig. 1) proceeds as follows:

1. We prime the algorithm with a modest amount of initial training data, which is chosen to be reasonable for the system of interest. For our system—a nanoparticle—we provided only bulk structures in the initial training set; these training data are described in Sec. IV. A bootstrap ensemble is trained to these data.
2. We query the bootstrap ensemble for atomic uncertainties ( $\delta_i$ ) of the structure of interest, in this case a Pt<sub>260</sub> nanoparticle, and we extract “chunks” centered on the most uncertain atoms. Specifically, each “chunk” includes the uncertain atom and its neighboring atoms within a given cutoff range. We will describe how we determine an appropriate cutoff range in Sec. V A.

The number of “chunks” to be extracted per pass is up to the user: while only extracting one chunk per pass may be the most efficient approach in terms of total computational time used by the parent calculator, extracting more chunks per pass allows one to take advantage of the embarrassingly parallel nature of submitting multiple electronic structures to a

computational queue simultaneously and avoids the need for excessive re-training. In this work, we extracted 52 chunks per pass for the DFT results and 26 for the EMT results.

3. The extracted chunks are evaluated by single-point calculations (e.g., DFT) and added into the fitting database.
4. The ensemble is re-trained. Upon re-training, only the force on the central atom of each cluster is included in the loss function; the forces on the other atoms, as well as the total energy, are discarded.

Return to step No. 2.

The iterative procedure runs until the maximum atomic uncertainty is below a given threshold, the number of nearsighted force training iterations is larger than the allowed number of iterations, or the uncertainty is not improved for two consecutive iterations.

We have implemented this procedure as part of the open-source machine-learning code *Amp* (the atomistic machine-learning package),<sup>11</sup> and scripts to replicate this work are included as the [supplementary material](#). In *Amp*, this includes algorithms for farthest-point sampling, feature selection (using the force-correlation or CUR approximation), and active learning within this NFT method.

## IV. TEST SYSTEM

### A. Atomic system

We tested the method on a 260-atom octahedron platinum nanoparticle, Pt<sub>260</sub>. Pt nanoparticles are well-known base materials for electrochemical catalysis. After creating a symmetric nanoparticle with a lattice constant of 3.99 Å (the DFT-optimized bulk value), we randomly displaced each atom by a distance drawn from a normal distribution with mean 0 and standard deviation 0.1 Å. This resulted in forces varying in magnitude from 0.1 to 5.1 eV/Å. From this “rattled” structure, we aimed at finding its optimized structure,

which is normally the first step in computational simulations of reactions on nanoparticle surfaces.

## B. Parent calculators

We tested with two different types of parent calculators: the effective medium theory (EMT)<sup>33</sup> and DFT.<sup>34</sup> We chose these two calculators to understand the nearsightedness effect. As EMT is a short-range calculator, the underlying PESs can, in principle, be exactly regressed by the finite-ranged ML potentials. In contrast, DFT is inherently a long-range calculator; thus finite-ranged ML potentials based on local interactions can only approximate the underlying PESs with limited accuracy. The DFT calculations were conducted in GPAW;<sup>35</sup> full calculator parameters are provided in the [supplementary material](#).

We emphasize that the NFT approach does not require the energy and forces of the full target structure, but for benchmarking purposes, we performed the full-size electronic structure calculations for key structures in this work. The full-size calculations allowed us to calculate what we term the “true forces,” which we can use to understand how well locally-approximated forces perform. In addition, we calculated the relaxation trajectory for Pt<sub>260</sub> and a larger cluster Pt<sub>309</sub> to show the capability of the model in the extrapolation PES regions.

## C. Initial training structures

All of the initial training structures in the current work were bulk structures—that is, periodic with no vacuum—and here we describe how we chose to create DFT reference structures. First, we generated an ensemble of randomized bulk structures, similar in spirit to the *ab initio* random structure search approach.<sup>36,37</sup> The structures included two components. First, we created a unit cell with two to four individual atoms based on a given space group. In this study, two space groups were employed, including space group No. 225 for a face-centered cubic (fcc) cell with four atoms and space group No. 79 for a tetragonal cell with two atoms. We then optimized the lattice constant of the bulk structure. We selected bulk cells whose lattice constants are close to the optimized one, aiming for sampling structures with low per-atom energies. Three structures with four atoms/cell and seven structures with two atoms/cell were selected for space group Nos. 225 and 79, respectively. This led to ten structures with the first approach. Second, we periodically doubled the cell in all dimensions of the optimized structure to obtain a larger cell. The repeated fcc and tetragonal cells consist of respective 32 and 16 atoms. We randomly perturbed atomic positions, ten times, to generate more diverse local environments, aiming for sampling structures with high per-atom energies. In total, this generated 20 randomized structures for the second approach. From these 30 structures, we selected the 20 most diverse and representative structures using a hierarchical farthest point sampling approach, which has been demonstrated to efficiently sample points from PESs in many studies.<sup>38,39</sup> The initial ensemble of neural network models was trained to these 20 bulk structures.

## D. Machine learning models

We built a ten-member ensemble of ML models with the bootstrap resampling approach.<sup>21</sup> When taking ionic steps in the

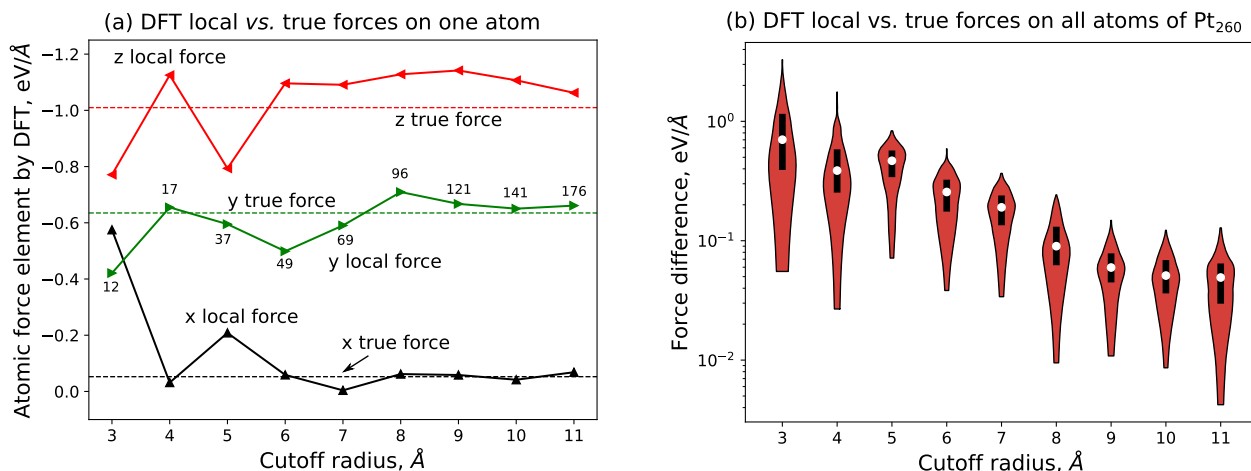
structural relaxation, we used the ensemble average for both the ML-predicted forces and energy. Note that since the average is a linear combination of ensemble member predictions—each of which is conservative—then the mean forces will be consistent with the mean potential energy. Note that this would not be the case if we had chosen the median rather than the average. Each ML model was a Behler–Parrinello type neural network.<sup>2</sup> A selection of Gaussian-type symmetry functions was used as the descriptor for local chemical environments. We added new uncertain structures into the fitting database by using an accelerated resampling method described by Peterson *et al.*<sup>21</sup> The specific details of the machine learning models and farthest point sampling algorithm are included in the [supplementary material](#).

## V. RESULTS AND DISCUSSION

Here, we present the results of the implementation of this methodology in piecewise simulating a Pt<sub>260</sub> nanoparticle, where the “chunk” calculations, as well as the full-size validation, all come from DFT. DFT is inherently a long-ranged calculator that poses no limits on the locality of electrons in the electronic structure. When we first developed the method, we employed the EMT calculator with the identical Pt<sub>260</sub> system; since EMT is inherently a nearsighted calculator (that is, it contains a built-in cutoff radius), the method can, in principle, give an exact match. These results are available in the [supplementary material](#), and we do, indeed, see a nearly perfect match when using EMT. Thus, we focus the remainder of the discussion on the DFT results, which contains the more challenging problem and realistic use-case.

### A. DFT force locality

Before implementing the nearsighted force training algorithm, we decided to investigate the inherent locality of the forces in DFT itself, which can help us to understand a reasonable cutoff radius for the machine-learning model going forward. Specifically, we first calculated the full-size Pt<sub>260</sub> nanoparticle in DFT, which provided what we term a “true force” on each atom. The true forces on the central atom are shown as the horizontal lines in [Fig. 2\(a\)](#). We next extracted chunks of increasing cutoff radius around this central atom and recalculated its forces in DFT; these are also shown in [Fig. 2\(a\)](#). We generally see large deviations for small cutoff radii, which smooth out around 6–7 Å, with the largest deviations on the largest force (which asymptotes very slowly to the true force). We repeated this process for each of the 260 atoms in the Pt<sub>260</sub> cluster, and we show statistics of these results in [Fig. 2\(b\)](#). (Presumably, the match between local and true forces could be improved with such methods as electrostatic embedding; we did not pursue such a strategy here.<sup>32,40</sup>) In order to achieve a good balance between accuracy and efficiency, we chose a cutoff of 8 Å for the force locality. Similar force-locality magnitudes have been reported for other metal systems involving a variety of defects,<sup>41</sup> while for covalent systems, a larger force locality was reported.<sup>42</sup> With such a cutoff, the maximum and mean force deviations between local and true forces are 0.24 and 0.10 eV/Å, respectively. Note that force deviation/difference used in this work is defined as the magnitude of vector difference between two forces. Since finite-ranged ML models can only describe the local contribution, the best possible accuracy



**FIG. 2.** DFT force locality analysis. (a) DFT local force components acting on one atom of Pt<sub>260</sub> at various cutoffs. The number of atoms included in the cutoff sphere is labeled in the y local force curve. (b) Distribution of deviations between DFT local and true forces on Pt<sub>260</sub> at various cutoffs. The distribution of deviations on all 260 atoms is represented by the violin shape. The 25th and 75th quartiles are indicated by the thick black line inside the violins. The median is represented by the hollow circle.

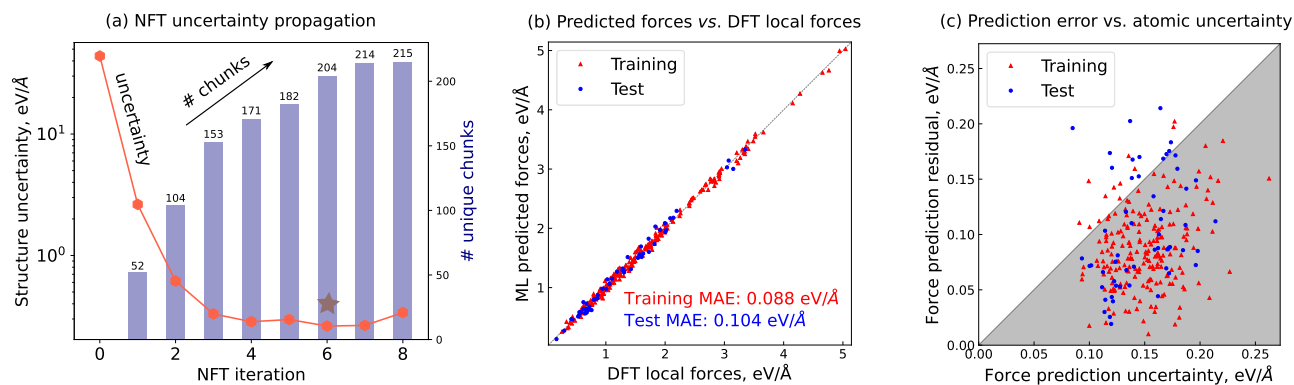
using ML models to approximate DFT true forces will be given by the intrinsic deviations between the DFT local forces and true forces. We emphasize that although we calculated the full-size Pt<sub>260</sub> nanoparticle in order to assess the nearsightedness of the DFT calculator, we did not use any full-size structures in the machine-learning training described in Sec. V B.

## B. Nearsighted force training

To prime the initial ensemble model, we started with a training set of 20 small bulk structures selected using the procedure described earlier. At each NFT iteration, we identified the 52 most uncertain atoms in the structure and added these 52 atomic chunks into the fitting database. As described earlier, only the force on

each central atom was added to the loss function to train the ensemble.

The progress of the algorithm is shown in Fig. 3(a); note that, after some time, some of the identified (most uncertain) atoms are repeated, so the rate of addition of atomic chunks diminishes over the number of steps. The predicted uncertainty is plotted in Fig. 3(a); the ensemble model giving the lowest uncertainty is obtained with six NFT steps, which shows a structure uncertainty of 0.26 eV/Å. At this point, the ensemble model has seen the forces of 204 atoms; that is, it has been trained on 204 atomic chunks. We compare the ML predicted forces to the DFT local forces (that is, the central-atom force from each chunk) in Fig. 3(b). The mean absolute errors (MAEs) for 204 training and 56 test atomic chunks are 0.088 and 0.104 eV/Å, respectively. In a normal application of this method, one



**FIG. 3.** Nearsighted force training on Pt<sub>260</sub> with reference data calculated by DFT. (a) Structure uncertainty propagation and the cumulative number of chunks during NFT iterations. The iteration step with the lowest structure uncertainty is marked with a star and selected as the model for further analyses. (b) ML-predicted forces vs DFT local forces. The forces represent force magnitudes, and the MAEs are calculated by averaging the force prediction residual, which is defined as the magnitude of difference of force vectors by ML models and DFT. (c) Force prediction residuals vs prediction uncertainties. A prediction uncertainty is the atomic uncertainty, as defined in Eq. (1).

would not have the true forces from the full structure to compare to, but as we noted earlier, we have calculated the full  $\text{Pt}_{260}$  structure, and thus, we also compare the ML forces and the true forces in Fig. S6. Against the true forces, the MAE is  $0.118 \text{ eV/\AA}$ ; thus, the true fit is close to the chunk-wise and best possible fit. In Fig. 3(c), we show a “runepplot” comparing the prediction uncertainties to the actual residuals. The structure uncertainty of  $0.26 \text{ eV/\AA}$  gives a reasonable upper bound for the force-prediction residuals, the maximum value of which is  $0.21 \text{ eV/\AA}$  for all atomic chunks.

### C. Structure optimization

With the NFT model trained to the initial  $\text{Pt}_{260}$  structure, we undertook a structural optimization. The relative energy vs relaxation step is shown as the “initial” curve of Fig. 4(a), which upon initial examination appears to show a well-behaved structural relaxation. The model’s uncertainty estimate is shown in Fig. 4(b); here, we are alerted that, after a few relaxation steps, the model loses confidence in the predictions. The large deviation is attributed to uncertain new local environments encountered during relaxation.

To improve the ensemble model, we further selected a small number of uncertain atomic chunks from the relaxation trajectory using the CUR algorithm, which has been used for fingerprint and image selections in previous ML potential studies.<sup>17,38,43,44</sup> The CUR algorithm was used because it allows structure selection without knowing the structure properties such as forces. Ten atomic chunks were selected because the feature matrix of all uncertain atomic

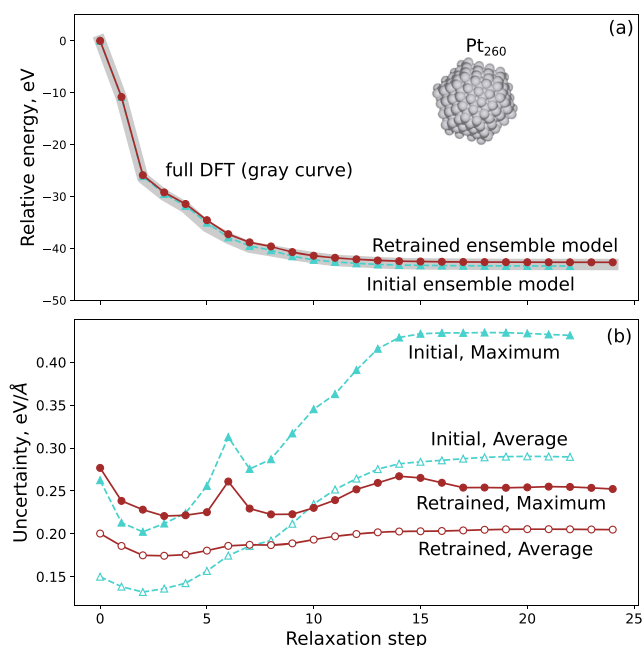
chunks can be largely represented by the reduced feature matrix of the selected chunks. These 10 chunks were added to the previous training set (comprised of 20 bulk structures and 204 chunks selected from the initial structure of  $\text{Pt}_{260}$ ), and the ensemble was re-trained with nearsighted force training on the initial structure. The model was retrained three times until the maximum structure uncertainty on the entire relaxation trajectory has not been improved for two consecutive retraining iterations, which shows a maximum structure uncertainty of  $0.28 \text{ eV/\AA}$ . In total, 54 new chunks were added into the fitting database. The best retrained model is as good as the initial model in terms of fitting the DFT local and true forces of the initial  $\text{Pt}_{260}$ , as shown in Fig. S6. Full details of the retraining algorithm can be found in Secs. III and IV B of the [supplementary material](#).

With the re-trained ensemble, we again undertook a structural relaxation, whose trajectory and predicted uncertainty are shown in Figs. 4(a) and 4(b). With the extra training data, we now see that the predicted uncertainty stays approximately constant throughout the trajectory, indicating a much higher confidence in these results. To validate the relaxation, we also undertook the full relaxation in DFT; these results are shown in Fig. 4(a) as well, and we see that there is a good match between the energetic pathway through the re-trained model and the true DFT relaxation. We also calculated the ML-predicted relaxed structure by DFT, and found a mean and maximum force of  $0.14$  and  $0.38 \text{ eV/\AA}$ , which is certainly not as relaxed as that would be found by a pure-DFT calculation, but is within the range of the uncertainty estimate. As a cross validation, we calculated the DFT relaxed structure by the retrained model and found a MAE and maximum deviation of  $0.13$  and  $0.33 \text{ eV/\AA}$ , which are on par with what we expect from the nearsighted analysis.

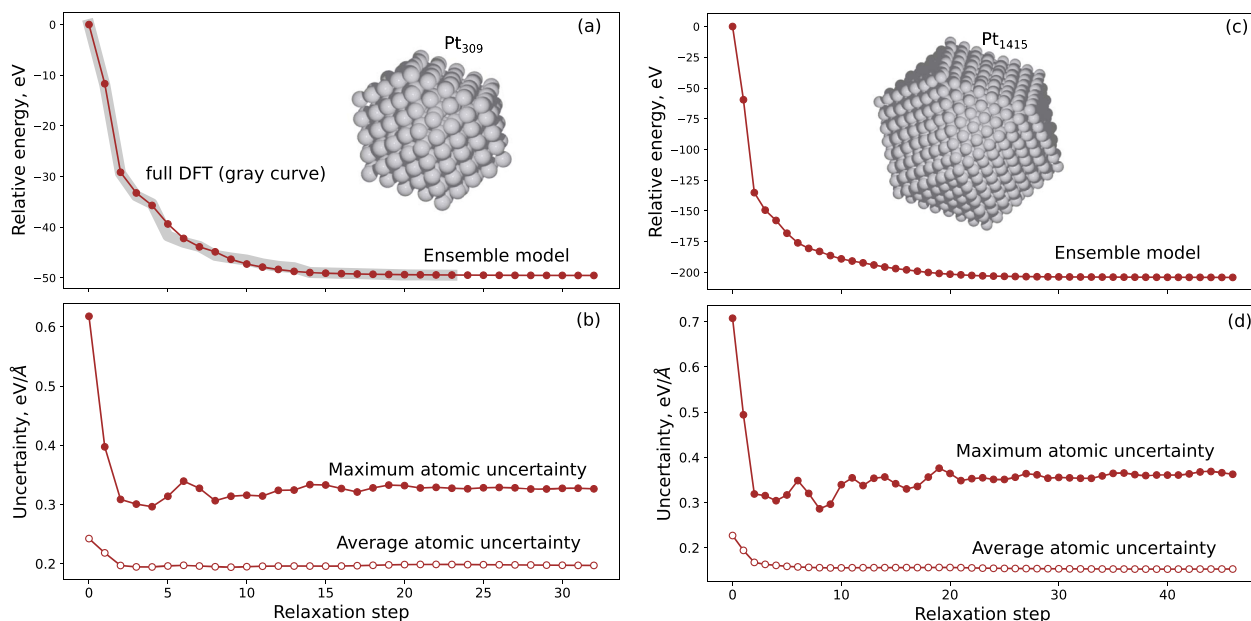
Note that both the initial and retrained models have only seen forces acting on central atoms of atomic chunks, rather than the entire nanoparticle. Therefore, the NFT approach should be able to do a decent job on larger systems consisting of tens of thousands of atoms as the NFT approach only relies on fitting atomic chunks comprising a limited number of atoms.

#### 1. Transferability to larger systems

We further examined the retrained model on a larger rattled cluster,  $\text{Pt}_{309}$ . The relative energy change by the retrained ensemble model matches well with the DFT result, see Fig. 5(a). This is consistent with the uncertainty propagation during relaxation, see Fig. 5(b). Although the first two structures of the ML-predicted trajectory are located outside the well-trained region (with uncertainties  $>0.40 \text{ eV/\AA}$ ), after two steps, the structure relaxes to a familiar region where the ML model can give more accurate force predictions, as indicated by the lower uncertainties. We also compared ML-predicted forces and DFT-evaluated true forces on the initial structure and ML-predicted relaxed structure, which gave MAEs of  $0.13$  and  $0.13 \text{ eV/\AA}$ , respectively. These errors are on par with the MAE of the retrained model on the initial structure of  $\text{Pt}_{260}$ , as shown in Fig. S6(b). We also extended this analysis to a much larger nanoparticle,  $\text{Pt}_{1415}$ , which is, to the best of our knowledge, the largest platinum nanoparticle that has ever been studied in a pure DFT calculation.<sup>45</sup> Although the relative energy change in Fig. 5(c) cannot be validated by a pure DFT calculation, the



**FIG. 4.** Structure optimization of the  $\text{Pt}_{260}$  nanoparticle by the ensemble models and DFT. (a) Relative energy vs relaxation step. (b) Uncertainty propagation during relaxation. The solid shape represents the maximum atomic uncertainty (that is, the structure uncertainty), whereas the hollow shape represents the average atomic uncertainty.



**FIG. 5.** Structure optimization of  $\text{Pt}_{309}$  and  $\text{Pt}_{1415}$  nanoparticles with the retrained ensemble model. (a) Relative energy vs relaxation step for  $\text{Pt}_{309}$ . (b) Uncertainty propagation during relaxation for  $\text{Pt}_{309}$ . (c) Relative energy vs relaxation step for  $\text{Pt}_{1415}$ . (d) Uncertainty propagation during relaxation for  $\text{Pt}_{1415}$ .

structure uncertainty shown in Fig. 5(d) falls into the high confidence region after two relaxation steps, which suggests that the ML relaxed  $\text{Pt}_{1415}$  should be close to the one predicted by a pure DFT calculation.

This simple analysis indicates that the re-trained ensemble model exhibited transferability to a larger system. Obviously, if this were not the case, we could just repeat the process of identifying uncertain atoms and re-training the model on these newly extracted chunks, identically as we did for the smaller particle.

## 2. Computational time, scalability, and parallelizability

The algorithm described here is expected to scale linearly with the system size (since the extracted chunks are of a constant size) vs the inherent cubic scaling of methods such as DFT—or the even higher scaling of more accurate electronic structure calculations. Here, we make a comparison of the number of DFT hours expended by each of the major tasks.

The relaxation of  $\text{Pt}_{206}$  in DFT took us about 8807 cpu-h, whereas the DFT relaxation of  $\text{Pt}_{309}$  took about 28 680 cpu-h. In contrast, the DFT calculations to create the training data for the ML method took about 7224 cpu-h and it took about 400 cpu-h for the NFT training. This represents a modest savings for the  $\text{Pt}_{260}$  system and a fairly dramatic savings for the larger  $\text{Pt}_{309}$  system. We would expect this savings to increase dramatically for larger systems, exploiting the linear scaling of the NFT method vs the cubic (or higher) scaling of typical electronic-structure methods.

The method described here offers major advantages in parallelizability. Whereas the large-scale DFT calculations can benefit from parallelization, DFT calculations can be difficult to

parallelize<sup>46</sup> and require all cores to be simultaneously available, creating difficulties for batching systems and possible reliability issues. In contrast, the many small DFT jobs created by the NFT algorithm are embarrassingly parallel (or “proudly parallel”); that is, they can be submitted individually to a queuing system and do not require all cores to be available simultaneously. For instance, it requires 120 cpu cores and 8 h to perform a single-point DFT calculation for  $\text{Pt}_{260}$ . In comparison, 16–48 cpu cores are sufficient for DFT calculations for atomic chunks. For example, it only takes 24 cores and 11 min for a single-point calculation of an atomic chunk with 46 atoms.

## VI. DISCUSSION AND CONCLUSION

The accuracy of most atomistic ML calculators is fundamentally limited by the locality assumption: most approaches employ a cutoff radius and only examine atoms within this radius when calculating per-atom quantities such as forces and energies. This assumption relies on the “nearsightedness” of electronic structure calculations; when we systematically examined the nearsightedness of the  $\text{Pt}_{260}$  structure by pulling out “chunks” of various radius and calculating in DFT, we found that at typical machine-learning cutoffs of 6–7 Å, the mean error in atomic forces was still around  $\sim 0.3$  eV/Å, suggesting an upper-bound for the ability of atomistic ML calculators in replicating the true DFT forces. When we increase this to 8 Å, we find the discrepancy to drop to about 0.1 eV/Å, which we took to be a decent trade-off between precision and computational speed.

Here, we exploit this limitation to systematically produce training data for large structures, based only on small structures that replicate the surrounding environment of the most uncertain

atoms in the structure. As a result, we see that the deviation of the ML-predicted forces from the true DFT forces is  $\sim 0.12$  eV/Å about that expected by the inherent limitation of the cutoff radius employed.

The nearsightedness analysis we performed here is by design pessimistic and meant to test the extreme case; that is, metal atoms outside the cutoff radius were replaced with vacuum. This was done both for the nearsighted analysis and the subsequent NFT sampling algorithm. In practice, one could make intelligent choices based on the system being simulated; for example, in this case, one might electrostatically embed the extracted cluster to simulate metal surroundings.<sup>32,40</sup> In cases involving covalent or insulator systems, for example, one might passivate dangling bonds or add a buffer layer to alleviate the issue of undesirable long-range interactions.<sup>32</sup> In a large-scale slab calculation, one would likely want to make the chunks extracted be periodic. This would presumably lower the uncertainty of both the nearsighted analysis and the resulting ML model fit to the extracted chunks.

The NFT method is expected to scale with  $\mathcal{O}(N)$  at worst, where  $N$  is the number of atoms in the system [as compared to  $\mathcal{O}(N^3)$  scaling for methods such as DFT]. Because of the similarity between regions of a large structure and the strategy we have employed for focusing on the most uncertain atoms, in practice, the scaling may actually be somewhat *less* than first-order. However, we should note that machine-learning is not necessary to make a similar algorithm with  $\mathcal{O}(N)$  scaling: for example, one could start with an  $N$ -atom structure and create  $N$  “chunks” of radius  $R_c$ , which are run in DFT, then use the central-atom forces from each of the chunks to take an ionic step. This would take  $N$  calculations per ionic step regardless of the system size, resulting in an  $\mathcal{O}(N)$ -method with approximately the same accuracy as that presented here. The machine-learning NFT method still saves considerable computational effort compared to such a divide-and-conquer approach. For our Pt<sub>260</sub> example, we needed 204 DFT-calculated atomic chunks to train the initial structure, which is similar to the 260 chunks needed in a divide-and-conquer strategy. However, when we undertook the structural relaxation, we added only a total of 54 more DFT-calculated chunks, whereas a divide and conquer would require  $260 \times 25 = 6500$  DFT-calculated chunks for this 25-step relaxation. The savings are even greater for the larger structures, where the NFT method did not require any additional training data, but the divide and conquer approach would have requested  $309 \times 33 = 10\,197$  and  $1415 \times 50 = 70\,750$  calculations for the Pt<sub>309</sub> and Pt<sub>1415</sub> calculations, respectively, compared to only 258 DFT calculations for the ML approach.

The NFT approach can be easily adapted for other ML models, if a certain atomic property, such as forces, is an output of the model. Although forces are chosen as the atomic property in the current work, this framework is not limited to this specific atomic property. It should be suitable for other properties such as charges, as long as the atomic property can be (largely) determined via the local environments. Prior work by Bianchini *et al.* found a local character in various metal systems for other atomic properties, such as charges and magnetization.<sup>41</sup> Since it trains on atomic chunks only, the NFT approach exploits the locality of the atomic property; therefore, it may be problematic for cases where strong long-range interactions cannot be ignored, for example, in many electrochemical scenarios including water.<sup>47</sup> Nevertheless, if

the long-range interactions, for example, electrostatic interactions, can be decoupled from the short-range interactions, as implemented in the third-generation high-dimensional neural network potentials,<sup>48</sup> the NFT approach can still take advantage of the atomic properties and improve the model by addressing uncertain local environments.

Training on simple atomic chunks representing local chemical environments, we have demonstrated that the NFT approach offers a general and robust approach for fitting PESs, making an important step toward applications of ML potentials in molecular simulations. Although a structure optimization is used as an example in this study, this approach can also be employed in other applications, such as reactivity studies on larger nanoparticles and global optimization of large nanoparticles.

## SUPPLEMENTARY MATERIAL

The [supplementary material](#) includes the derivation of atomic force cutoff in finite-range ML potentials (Sec. I), machine learning models (Sec. II), the retraining procedure for structure optimization (Sec. III), the image and feature selection algorithms (Sec. IV), computational settings of DFT calculations (Sec. V), supplemental results (Sec. VI), application with a nearsighted parent calculator of EMT (Sec. VII), and example scripts to replicate this work (Sec. VIII).

## ACKNOWLEDGMENTS

The authors thank Mayank Agrawal (Brown University) for providing valuable technical feedback. Financial support was provided by the U.S. Department of Energy’s Basic Energy Science, Computational Chemical Sciences Program Office, under Award No. DE-SC0019441. C.Z. is also indebted to the financial support from the Presidential Fellowship of Brown University. The computational simulations were undertaken at Brown University’s Center for Computation and Visualization (CCV).

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts of interest to disclose.

## DATA AVAILABILITY

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

## REFERENCES

- 1 D. R. Bowler and T. Miyazaki, “ $\mathcal{O}(N)$  methods in electronic structure calculations,” *Rep. Prog. Phys.* **75**, 036503 (2012).
- 2 J. Behler and M. Parrinello, “Generalized neural-network representation of high-dimensional potential-energy surfaces,” *Phys. Rev. Lett.* **98**, 146401 (2007).
- 3 J. Behler, “Four generations of high-dimensional neural network potentials,” *Chem. Rev.* **121**, 10037 (2021).
- 4 A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, “Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons,” *Phys. Rev. Lett.* **104**, 136403 (2010).

- <sup>5</sup>M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, “Fast and accurate modeling of molecular atomization energies with machine learning,” *Phys. Rev. Lett.* **108**, 058301 (2012).
- <sup>6</sup>T. Mueller, A. Hernandez, and C. Wang, “Machine learning for interatomic potential models,” *J. Chem. Phys.* **152**, 050902 (2020).
- <sup>7</sup>L. Himanen, M. O. J. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, “DScribe: Library of descriptors for machine learning in materials science,” *Comput. Phys. Commun.* **247**, 106949 (2020).
- <sup>8</sup>M. Rupp, “Machine learning for quantum mechanics in a nutshell,” *Int. J. Quantum Chem.* **115**(16), 1058–1073 (2015).
- <sup>9</sup>K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, “Quantum-chemical insights from deep tensor neural networks,” *Nat. Commun.* **8**, 13890 (2017).
- <sup>10</sup>K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet—A deep learning architecture for molecules and materials,” *J. Chem. Phys.* **148**, 241722 (2018).
- <sup>11</sup>A. Khorshidi and A. A. Peterson, “Amp: A modular approach to machine learning in atomistic simulations,” *Comput. Phys. Commun.* **207**, 310–324 (2016).
- <sup>12</sup>A. P. Bartók, R. Kondor, and G. Csányi, “On representing chemical environments,” *Phys. Rev. B* **87**, 184115 (2013).
- <sup>13</sup>J. Behler, “Atom-centered symmetry functions for constructing high-dimensional neural network potentials,” *J. Chem. Phys.* **134**, 074106 (2011).
- <sup>14</sup>A. A. Peterson, “Acceleration of saddle-point searches with machine learning,” *J. Chem. Phys.* **145**, 074106 (2016).
- <sup>15</sup>E. L. Kolsbjerg, A. A. Peterson, and B. Hammer, “Neural-network-enhanced evolutionary algorithm applied to supported metal nanoparticles,” *Phys. Rev. B* **97**, 195424 (2018).
- <sup>16</sup>K. Gubaev, E. V. Podryabinkin, G. L. W. Hart, and A. V. Shapeev, “Accelerating high-throughput searches for new alloys with active learning of interatomic potentials,” *Comput. Mater. Sci.* **156**, 148–156 (2019).
- <sup>17</sup>N. Bernstein, G. Csányi, and V. L. Deringer, “De novo exploration and self-guided learning of potential-energy surfaces,” *npj Comput. Mater.* **5**, 99 (2019).
- <sup>18</sup>L. Zhang, D.-Y. Lin, H. Wang, R. Car, and W. E., “Active learning of uniformly accurate interatomic potentials for materials simulation,” *Phys. Rev. Mater.* **3**, 023804 (2019).
- <sup>19</sup>M. Shuaibi, S. Sivakumar, R. Q. Chen, and Z. W. Ulissi, “Enabling robust offline active learning for machine learning potentials using simple physics-based priors,” *Mach. Learn.: Sci. Technol.* **2**, 025007 (2021).
- <sup>20</sup>K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev, “Machine learning of molecular properties: Locality and active learning,” *J. Chem. Phys.* **148**, 241727 (2018).
- <sup>21</sup>A. A. Peterson, R. Christensen, and A. Khorshidi, “Addressing uncertainty in atomistic machine learning,” *Phys. Chem. Chem. Phys.* **19**, 10978–10985 (2017).
- <sup>22</sup>G. Csányi, T. Albaret, G. Moras, M. C. Payne, and A. D. Vita, “Multiscale hybrid simulation methods for material systems,” *J. Phys.: Condens. Matter* **17**, R691–R703 (2005).
- <sup>23</sup>N. Artrith and J. Behler, “High-dimensional neural network potentials for metal surfaces: A prototype study for copper,” *Phys. Rev. B* **85**, 045439 (2012).
- <sup>24</sup>M. W. Mahoney and P. Drineas, “CUR matrix decompositions for improved data analysis,” *Proc. Natl. Acad. Sci. U. S. A.* **106**, 697–702 (2009).
- <sup>25</sup>W. Kohn, “Density functional and density matrix method scaling linearly with the number of atoms,” *Phys. Rev. Lett.* **76**, 3168–3171 (1996).
- <sup>26</sup>E. Prodan and W. Kohn, “Nearsightedness of electronic matter,” *Proc. Natl. Acad. Sci. U. S. A.* **102**(33), 11635–11638 (2005).
- <sup>27</sup>T. W. Ko, J. A. Finkler, S. Goedecker, and J. Behler, “A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer,” *Nat. Commun.* **12**, 398 (2021).
- <sup>28</sup>M. A. Blanco, A. Martín Pendás, and E. Francisco, “Interacting quantum atoms: A correlated energy decomposition scheme based on the quantum theory of atoms in molecules,” *J. Chem. Theory Comput.* **1**, 1096–1109 (2005).
- <sup>29</sup>L.-W. Wang, “Charge-density patching method for unconventional semiconductor binary systems,” *Phys. Rev. Lett.* **88**, 256402 (2002).
- <sup>30</sup>V. Botu and R. Ramprasad, “Learning scheme to predict atomic forces and accelerate materials simulations,” *Phys. Rev. B* **92**, 094306 (2015).
- <sup>31</sup>A. Glielmo, P. Sollich, and A. De Vita, “Accurate interatomic force fields via machine learning with covariant kernels,” *Phys. Rev. B* **95**, 214302 (2017).
- <sup>32</sup>N. Bernstein, J. R. Kermode, and G. Csányi, “Hybrid atomistic simulation methods for materials systems,” *Rep. Prog. Phys.* **72**, 026501 (2009).
- <sup>33</sup>J. Nørskov and N. D. Lang, “Effective-medium theory of chemical binding: Application to chemisorption,” *Phys. Rev. B* **21**, 2131 (1980).
- <sup>34</sup>W. Kohn and L. J. Sham, “Self-consistent equations including exchange and correlation effects,” *Phys. Rev.* **140**, A1133–A1138 (1965).
- <sup>35</sup>J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dulak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Møller, M. Strange, G. A. Tritsarlis, M. Vanin, M. Walter, B. Hammer, H. Häkkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nørskov, M. Puska, T. T. Rantala, J. Schiøtz, K. S. Thygesen, and K. W. Jacobsen, “Electronic structure calculations with GPAW: A real-space implementation of the projector augmented-wave method,” *J. Phys.: Condens. Matter* **22**, 253202 (2010).
- <sup>36</sup>C. J. Pickard and R. J. Needs, “Ab initio random structure searching,” *J. Phys.: Condens. Matter* **23**, 053201 (2011).
- <sup>37</sup>V. L. Deringer, C. J. Pickard, and G. Csányi, “Data-driven learning of total and local energies in elemental boron,” *Phys. Rev. Lett.* **120**, 156001 (2018).
- <sup>38</sup>G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, “Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials,” *J. Chem. Phys.* **148**, 241730 (2018).
- <sup>39</sup>B. Cheng, G. Mazzola, C. J. Pickard, and M. Ceriotti, “Evidence for supercritical behaviour of high-pressure liquid hydrogen,” *Nature* **585**, 217–220 (2020).
- <sup>40</sup>N. Choly, G. Lu, W. E, and E. Kaxiras, “Multiscale simulations in simple metals: A density-functional-based methodology,” *Phys. Rev. B* **71**, 094101 (2005).
- <sup>41</sup>F. Bianchini, J. R. Kermode, and A. De Vita, “Modelling defects in Ni–Al with EAM and DFT calculations,” *Modell. Simul. Mater. Sci. Eng.* **24**, 045012 (2016).
- <sup>42</sup>A. Peguiron, L. Colombi Ciacchi, A. De Vita, J. R. Kermode, and G. Moras, “Accuracy of buffered-force QM/MM simulations of silica,” *J. Chem. Phys.* **142**, 064116 (2015).
- <sup>43</sup>R. Jinnouchi, F. Karsai, and G. Kresse, “On-the-fly machine learning force field generation: Application to melting points,” *Phys. Rev. B* **100**, 014105 (2019).
- <sup>44</sup>T. T. Nguyen, E. Székely, G. Imbalzano, J. Behler, G. Csányi, M. Ceriotti, A. W. Götz, and F. Paesani, “Comparison of permutationally invariant polynomials, neural networks, and Gaussian approximation potentials in representing water interactions through many-body expansions,” *J. Chem. Phys.* **148**, 241725 (2018).
- <sup>45</sup>L. Li, A. H. Larsen, N. A. Romero, V. A. Morozov, C. Glinsvad, F. Abild-Pedersen, J. Greeley, K. W. Jacobsen, and J. K. Nørskov, “Investigation of catalytic finite-size-effects of platinum metal clusters,” *J. Phys. Chem. Lett.* **4**, 222–226 (2013).
- <sup>46</sup>V. Mironov, A. Moskovsky, M. D’Mello, and Y. Alexeev, “An efficient MPI/OpenMP parallelization of the Hartree–Fock–Roothaan method for the first generation of Intel® Xeon Phi™ processor architecture,” *Int. J. High Perform. Comput. Appl.* **33**, 212–224 (2019).
- <sup>47</sup>C. H. Choi, H.-K. Lim, M. W. Chung, J. C. Park, H. Shin, H. Kim, and S. I. Woo, “Long-range electron transfer over graphene-based catalyst for high-performing oxygen reduction reactions: Importance of size, N-doping, and metallic impurities,” *J. Am. Chem. Soc.* **136**, 9070–9077 (2014).
- <sup>48</sup>T. Morawietz, V. Sharma, and J. Behler, “A neural network potential-energy surface for the water dimer based on environment-dependent atomic energies and charges,” *J. Chem. Phys.* **136**, 064103 (2012).