The Journal of Chemical Physics

Extending the accuracy of the SNAP interatomic potential form

Cite as: J. Chem. Phys. **148**, 241721 (2018); https://doi.org/10.1063/1.5017641@jcp.2018.DETC2018.issue-1
Submitted: 28 November 2017 . Accepted: 22 February 2018 . Published Online: 29 March 2018

Mitchell A. Wood, and Aidan P. Thompson

COLLECTIONS

Paper published as part of the special topic on Collection and Data-Enabled Theoretical Chemistry

View Online Export Citation CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Atom-centered symmetry functions for constructing high-dimensional neural network potentials
The Journal of Chemical Physics **134**, 074106 (2011); https://doi.org/10.1063/1.3553717

Perspective: Machine learning potentials for atomistic simulations
The Journal of Chemical Physics **145**, 170901 (2016); https://doi.org/10.1063/1.4966192

Extending the accuracy of the SNAP interatomic potential form
The Journal of Chemical Physics **148**, 241721 (2018); https://doi.org/10.1063/1.5017641

AIP Publishing

© 2018 Author(s).

Extending the accuracy of the SNAP interatomic potential form

Mitchell A. Wood and Aidan P. Thompson
Center for Computing Research, Sandia National Laboratories, Albuquerque, New Mexico 87185, USA

(Received 28 November 2017; accepted 22 February 2018; published online 29 March 2018)

The Spectral Neighbor Analysis Potential (SNAP) is a classical interatomic potential that expresses the energy of each atom as a linear function of selected bispectrum components of the neighbor atoms. An extension of the SNAP form is proposed that includes quadratic terms in the bispectrum components. The extension is shown to provide a large increase in accuracy relative to the linear form, while incurring only a modest increase in computational cost. The mathematical structure of the quadratic SNAP form is similar to the embedded atom method (EAM), with the SNAP bispectrum components serving as counterparts to the two-body density functions in EAM. The effectiveness of the new form is demonstrated using an extensive set of training data for tantalum structures. Similar to artificial neural network potentials, the quadratic SNAP form requires substantially more training data in order to prevent overfitting. The quality of this new potential form is measured through a robust cross-validation analysis. *Published by AIP Publishing.* https://doi.org/10.1063/1.5017641

I. INTRODUCTION

Simulation methods such as classical molecular dynamics (MD) or Monte Carlo sampling are powerful tools for examining the behaviors of materials that originate on the atomic scale. Using scalable algorithms and large parallel computers, it is now possible to routinely simulate the structural evolution of chunks of material containing millions of atoms on time scales approaching 1 μs. This is possible because, in many situations, an explicit quantum mechanical treatment of the electronic degrees of freedom can be avoided, by defining a classical interatomic potential function (IAP) that depends only on the atomic degrees of freedom. Typical IAP forms provide short-ranged explicit expressions for the forces on the atoms that can be efficiently decomposed into independent parallel computations. As a result, MD simulations are many orders of magnitude faster than the corresponding quantum mechanical simulation for systems composed of tens or hundreds of atoms. Moreover, while the cost of even a single force evaluation becomes prohibitively expensive for quantum calculations involving more than a few hundred atoms, the computational speed of classical MD simulations remains largely independent of the total number of atoms, provided that large problems are executed on an equally large number of processors.[1]

The extent to which MD simulations provide an accurate description of real materials is largely dependent on how well the IAP approximates the true quantum mechanical energy of the system.[2] An IAP is essentially a reduced order model that takes the full quantum many-body problem of ionic and electronic interactions and casts it into computationally tractable expressions for the total energy as the sum of individual atom contributions.[3] However, this simplification can be done in many different ways, giving rise to the numerous different types of IAPs that are seen in the literature. The simplest and computationally most efficient express the energies and forces solely in terms of pair-wise interactions between nearby atoms.[4,5] In recent years, there has been a rise in the use of more complex and costly potentials.[6] These advancements have enabled MD simulations of chemical reactions in organic systems[7–9] and electric charge polarization in insulating materials.[10,11]

Nearly all of these IAPs have mathematical forms that are physically and chemically motivated. They exploit our understanding of the nature of a chemical bond or mechanical deformation of solids to formulate the interatomic potential. For example, the embedded atom method (EAM) decomposes the potential energy of each atom into pair interactions with neighbors and a many-body embedding energy that depends on the local mean electron density due to neighbor atoms.[12,13] Extensions of EAM use higher-order moments of the local electron density to represent polarization and directional bonding effects.[14] Bond order potentials like Tersoff[8] and ReaxFF[7,15] simplify covalent bonding into smooth functions of interatomic distance and atomic coordination. The finer details of the chemical environment around a pair of bonded atoms can be described by more complex energy expressions that describe specific bonding patterns.[9,16]

An alternative to this approach relies not on advances in physics and chemistry theory, but rather on data science and artificial intelligence. Instead of developing theory-inspired functions for IAPs, the approach of machine-learned interatomic potentials (ML-IAPs) adopts a very general mathematical representation of the potential energy that satisfies just a few essential physical requirements. These IAPs contain many free parameters that are trained to reproduce features of the high fidelity electronic structure calculations.[17,18] Because of this generality, it is thought that ML-IAPs are more flexible and are capable of accurately describing a wide range of chemical systems, provided that sufficient reference data are used to train and test.

The improved accuracy of these ML-IAPs comes at the price of increased computational cost. This cost is incurred both in the process of generating the training data for the

ML-IAP and also when it is used to run large-scale MD simulations. Nonetheless, because generation of the training data requires large numbers of quantum electronic structure calculations with small numbers of atoms, while the ML-IAP force calculations scale even better than conventional IAPs, the increase in computational cost is greatly mitigated by the increasing abundance of low-cost massively parallel computational resources.

In conceiving and refining our particular approach to ML-IAPs, we have sought to find a good balance between accuracy and computational cost. These considerations lead us to construct the Spectral Neighbor Analysis Potential (SNAP) model. The SNAP method was shown to provide a good description of a range of different properties of various solid phases of tantalum and also the liquid structure.[19] In this paper, we demonstrate how several fairly simple extensions to the SNAP model significantly improve both training and cross-validation errors with only modest increase in computational cost.

II. REGRESSION METHODOLOGY

The purpose of any ML-IAP is to express the potential energy surface (PES) in terms of a set of scalar functions known as descriptors. The descriptors are somewhat arbitrary functions that depend on the local atomic environment of an atom. Each atomic environment, expressed as Cartesian coordinates of the neighbor atoms relative to the central atom, is first mapped to a point in the descriptor space. That point in turn is mapped to a value of energy. The overall effectiveness of the ML-IAP is sensitive to the choice of descriptor space. As a minimal requirement, the descriptor space should be invariant under rotation, permutation, and translation of the atomic environment.[17,20] In addition, for multi-element potentials, the descriptors should distinguish between different atom types.[21,22] In recent years, many different types of descriptors have been proposed.[23–25] More complicated descriptor spaces include basis expansions based on symmetries in the local environment around each atom.[26–29]

The bispectrum components, which are the focus of this work, fall into this category.[19,30,31] They have also been used as descriptors in several recent studies by other groups.[32,33] The question of which approach to describing atomic neighborhoods is most effective is an active area of research. For example, there is some evidence that direct pairwise comparison of atomic neighborhoods using a similarity measure is more effective than trying to resolve the atomic neighborhood using invariants extracted from a basis function expansion, such as the bispectrum components.[29] However, basis expansion invariants have the advantage of being directly amenable to linear regression methods.

ML-IAPs must be trained using reference data in order to construct a model form that connects the descriptor space to the target feature space, i.e., known energies and forces from a higher fidelity model.[34–36] This training step can be approached in a variety of ways. One powerful class of method is the artificial neural network (ANN).[37–39] This uses multiple convolutions (layers) of the descriptor space to capture the most important features. The weight assigned to connections between nodes in successive layers are free variables in the fit. While shown to produce some of the most accurate representations of training data,[40,41] a typical ANN requires significantly more training data than other model forms and contains $\mathcal{O}(10^3–10^5)$ fit parameters, which can make it difficult to train. Other model forms interpolate between data in the training set through some similarity metric for unknown structures.[26,42,43] However, these ML-IAPs are computationally expensive to use for MD simulations because the cost scales with the size of the training set.[44–46]

Lastly there exists a set of linear regression models that reduce the dimensionality of the training set size to that of the descriptor space via a linear transformation.[47] These linear models are a special case of ANN with only a single convolution layer, i.e., the number of fitting terms in the regression is equal to the size of the descriptor space. Linear models can also be viewed as a special form of Gaussian process regression in which a dot product kernel is used for the covariance matrix.[48]

The mapping from the descriptor space (D) to the training space (T) is determined by the vector of linear coefficients $\boldsymbol{\beta}$, chosen to minimize the Euclidean norm distance between training and predicted points in the training space,

$$\min(||\mathbf{w} \cdot D\boldsymbol{\beta} - T||^2 - \gamma_n \, ||\boldsymbol{\beta}||^n). \tag{1}$$

A regularization penalty [second term in Eq. (1)] of order n can be added to the linear least squares (LLS) term. Solutions with $n = 1$ regularization enforce sparsity of the $\boldsymbol{\beta}$ solution, known as Least Absolute Shrinkage and Selection Operator (LASSO) solutions.[49] This can be used to isolate key features of the descriptor space but is rarely used to construct ML-IAPs because the training space is often much larger than the descriptor space. Similarly, solutions with $n = 2$ will penalize large values of $\boldsymbol{\beta}$ which are hallmarks of an overfit solution, known as Tikhonov regularization.[50] In this work, an overfit solution is the result of a lack of diversity in the training set relative to the number of free variables used in the fit. Avoiding overfit solutions that are also capable of accurate out-of-sample prediction is critical for producing a viable ML-IAP for use in MD. A simple means to improve the fit quality is to add more degrees of freedom, but if this is done haphazardly, the overall transferability of the potential will be sacrificed. Additional weights can be applied to specific terms of the descriptor space through the vector $\mathbf{w}$.

III. LINEAR SNAP

In this section, we discuss some general practical concerns around the use of ML-IAPs. While there has been significant evidence that ML-IAPs are capable of producing high accuracy results with respect to density functional theory (DFT) quantum electronic structure calculations, a balance must be struck between the effort required to generate training data and the computational cost of running a simulation. To many end users, limitations on computing resources are an important factor when selecting an IAP for MD simulation. In addition, for developers of IAPs, significant time is required to train and test candidate potentials. Therefore, regardless of how elegant the model form or absolute accuracy of the method is, consideration of these front- and back-end computational costs must be made.

It is difficult to precisely quantify the time spent constructing training data because this will largely depend on the details of what the ML-IAP is intended for. For example, crystalline properties of metals can be calculated quickly within DFT because perfectly periodic structures can be represented with just a few atoms. However, extrapolation of such a potential to configurations such as defects and surfaces will result in large errors since there is nothing in this limited training set to constrain the model in those regions of the PES. Therefore a diverse training set is preferred in practice so that end usage of the potential is less susceptible to large extrapolation errors. This, of course, adds to the upfront cost of parameterizing the IAP.

The dominant cost of running an MD simulation is the evaluation of the descriptors, from which the corresponding force contributions can be obtained. We outline here the structure of the SNAP ML-IAP in terms of the underlying descriptor space.[19] The total potential energy of a configuration of atoms is first written as the sum of SNAP energy contributions associated with each atom, combined with a reference potential

$$E(\mathbf{r}^N) = E_{ref}(\mathbf{r}^N) + \sum_{i=1}^{N} E^i_{SNAP}, \tag{2}$$

where $\mathbf{r}^N$ is the vector of N atom positions in the configuration. E and E_{ref} are the total and reference potential energies, respectively. E^i_{SNAP} is the SNAP potential energy associated with a particular atom i and depends only on the relative positions of its neighbor atoms. Including a reference potential is advantageous because it can correctly represent known limiting cases of atomic interactions, leaving the SNAP contribution to capture many-body effects. The Ziegler-Biersack-Littmark (ZBL) pair potential[51] is a convenient choice because it captures the known short-range repulsive interactions between atomic cores that are not well represented by quantum calculations.

The construction of the SNAP component of the potential energy in terms of the bispectrum components follows the same approach described in Ref. 19, which we briefly summarize here. The SNAP formulation begins with a very general characterization of the neighborhood of an atom. The density of neighbor atoms at location $\mathbf{r}$ relative to a central atom i can be considered as a sum of δ-functions located in a three-dimensional space

$$\rho_i(\mathbf{r}) = \delta(\mathbf{r}) + \sum_{r_{i'} < R_{cut}} f_c(r_{i'}) w_{i'} \delta(\mathbf{r} - \mathbf{r}_{i'}), \tag{3}$$

where $\mathbf{r}_{i'}$ is the position of neighbor atom i' relative to central atom i. The $w_{i'}$ coefficients are dimensionless weights that are chosen to distinguish atoms of different types, while the central atom is arbitrarily assigned a unit weight. The sum is over all atoms i' within some cutoff distance R_{cut}. The switching function $f_c(r)$ ensures that the contribution of each neighbor atom goes smoothly to zero at R_{cut}. Typically, this density function is expanded in an angular basis of spherical harmonics combined with an orthonormal radial basis.[23] Instead, we use an idea originally proposed by Bartók *et al.*,[30] in which the radial coordinate r is mapped on to a third angular coordinate $\theta_0 = \theta_0^{max} r / R_{cut}$. Each neighbor position (r, θ, ϕ) is mapped to (θ_0, ϕ, θ), a point on the unit 3-sphere. The natural basis for functions on the 3-sphere is formed by the 4D hyperspherical harmonics $U^j_{m,m'}(\theta_0, \theta, \phi)$, defined for $j = 0, \frac{1}{2}, 1, \ldots$ and m, $m' = -j, -j+1, \ldots, j-1, j$.[52] The neighbor density function can now be expanded on the basis of hyperspherical harmonics $U^j_{m,m'}$. Because the neighbor density is a weighted sum of δ-functions, each expansion coefficient is a sum over discrete values of the corresponding basis function evaluated at each neighbor position,

$$u^j_{m,m'} = U^j_{m,m'}(0) + \sum_{r_{ii'} < R_{cut}} f_c(r_{ii'}) w_{i'} U^j_{m,m'}(\theta_0, \theta, \phi). \tag{4}$$

The bispectrum components are formed as the scalar triple products of the expansion coefficients,

$$B_{j_1 j_2 j} = \sum_{m,m'} u^{j*}_{m,m'} \sum_{\substack{m_1,m'_1 \\ m_2,m'_2}} H^{jmm'}_{\substack{j_1 m_1 m'_1 \\ j_2 m_2 m'_2}} u^{j_1}_{m_1,m'_1} u^{j_2}_{m_2,m'_2}, \tag{5}$$

where * indicates complex conjugation and the constants $H^{jmm'}_{\substack{j_1 m_1 m'_1 \\ j_2 m_2 m'_2}}$ are Clebsch-Gordan coupling coefficients for the hyperspherical harmonics. The bispectrum components are real-valued and invariant under rotation.[31] They are also symmetric in the three indices j_1, j_2, j up to a normalization factor.[19] They characterize the strength of density correlations at three points on the 3-sphere. The lowest-order components describe the coarsest features of the density function, while higher-order components reflect finer detail. The number of distinct bispectrum components with indices j_1, j_2, j less than or equal to J_{max} is given in Table I. By increasing the value of J_{max} we can, in principle, systematically increase the accuracy of the SNAP potential, at the price of greater computational cost, as shown in Fig. 1. For a particular choice of J_{max}, we can list the K bispectrum components in some arbitrary order as B_1, ..., B_K. The SNAP energy of an atom is written as a linear function of these K bispectrum components,

$$E^i_{SNAP} = \beta_0 + \sum_{k=1}^{K} \beta_k (B^i_k - B^i_{k0}), \tag{6}$$

$$= \beta_0 + \boldsymbol{\beta} \cdot \mathbf{B}^i, \tag{7}$$

where B^i_k is the kth bispectrum component of atom i and β_k is the associated linear coefficient, a free parameter in the SNAP model. As a computational convenience, the

TABLE I. Number of bispectrum components (descriptors), linear coefficients, and quadratic coefficients for increasing values of J_{max}.

J_{max}	Descriptors K	Linear coefficients	Quadratic coefficients
1	5	5	15
2	14	14	105
3	30	30	465
4	55	55	1540
5	91	91	4186
J	$\frac{(J+1)(J+\frac{3}{2})(J+2)}{3}$	K	$\frac{K(K+1)}{2}$

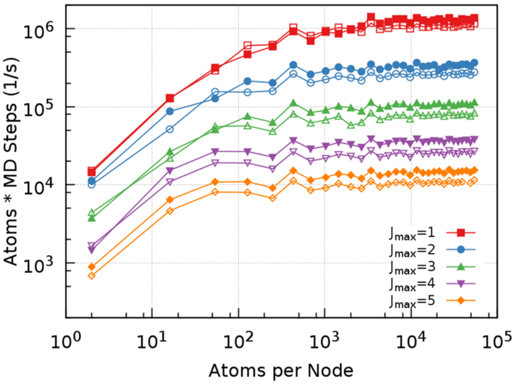

FIG. 1. Computational speed versus atom count for linear (closed symbols) and quadratic (open symbols) SNAP potential forms with increasing values of J_{max} and correspondingly larger descriptor spaces (see Table I). Timings were made using a single Intel Broadwell node. For each potential, the speed in atom timestep/s becomes roughly constant for sufficiently large atom count, reflecting the linear scaling of the underlying LAMMPS[54] neighbor list algorithm.

contribution of each bispectrum component to the SNAP energy is shifted by the contribution of an isolated atom, $\beta_k B^i_{k0}$, so that the SNAP energy of the isolated atom is equal to β_0 by construction. Similarly, the force on each atom j due to the SNAP potential can be expressed as a weighted sum over the derivatives w.r.t. $\mathbf{r}_j$ of the bispectrum components of each atom i,

$$\mathbf{F}^j_{SNAP} = -\nabla_j \sum_{i=1}^{N} E^i_{SNAP} = -\boldsymbol{\beta} \cdot \sum_{i=1}^{N} \frac{\partial \mathbf{B}^i}{\partial \mathbf{r}_j}. \tag{8}$$

In this way, the total energy, forces, and also the stress tensor can be written as linear functions of quantities related to the bispectrum components of the atoms. In addition to shifting the bispectrum components by B^i_{k0}, it also makes sense to set $\beta_0 = 0$, constraining the potential energy of an isolated atom to be zero. This ensures that SNAP correctly reproduces the cohesive energy of the reference solid structure, an important physical attribute of any general purpose interatomic potential.

Using a linear regression model such as LLS whose solution yields $\boldsymbol{\beta}$, the cost of SNAP is decoupled from the size of the training set and will only depend on the number of bispectrum components calculated for each atom. Table I shows how the size of the descriptor space K grows as a function of J_{max}, the maximum order of the bispectrum components. It is important to note the ratio of fitting coefficients to the size of the descriptor space for either SNAP form. The quadratic SNAP will be able to capture more features of the training set due to the increased number of free fitting variables. Since the SNAP potential energy is a linear function of these K descriptors, the number of coefficients that are solved for in the linear regression [$\boldsymbol{\beta}$ in Eq. (1)] is equal to the size of the descriptor space.

The computational cost of SNAP depends strongly on the size of the descriptor space.[53] Figure 1 captures this behavior for values of J_{max} ranging from 1 to 5. We measure computational cost by running a short MD simulation using LAMMPS[54,55] for each SNAP potential on a single Intel Broadwell node of Sandia's Serrano computing cluster. The node consists of two 2.1 GHz Intel Broadwell E5-2695v4 processors with 18 physical cores each. At runtime, the performance of each SNAP potential was optimized by varying the number of Message Passing Interface (MPI) tasks while adjusting the number of OpenMP threads per MPI task to keep the total number of threads on the node equal to 36, the total number of cores. We used default memory access settings, we did not enable hyperthreading, and we did not allow any cores to remain idle. We repeated this over a wide range of atom counts because the optimal number of atoms per node is different for different SNAP potentials.[56] The closed symbols in Fig. 1 correspond to the linear form of SNAP, while open symbols refer to the quadratic formulation of the SNAP energy function which will be discussed further in Sec. IV. Clearly, as the number of bispectrum components increases, the speed of the MD calculation diminishes. In going from $J_{max} = 1$ to $J_{max} = 5$, the size of the descriptor space increases by about a factor of 20, and the computational cost increases by approximately two orders of magnitude. For comparison, a Modified Embedded Atom Method (MEAM) potential[57,58] for tungsten reached a peak speed of $5{\cdot}10^6$ atom timestep/s on the same hardware, while the Gaussian Approximation Potentials (GAP) potential for tungsten[26,59] reached a peak performance near $5{\cdot}10^2$ atom timesteps/s.

IV. QUADRATIC SNAP

One of the potential limitations of the SNAP approach is the restriction to a linear relationship between the atom potential energy and the bispectrum components of the atom neighbor density. It is shown below that this limits the bodiedness or maximum complexity of energy terms to four-body. Just as the embedded atom method has proven to be an effective way to add many-body interactions to pair potentials,[12,13] we can extend the bodiedness of SNAP potentials by adding an embedding energy term

$$E^i_{SNAP}(\mathbf{r}^N) = \boldsymbol{\beta} \cdot \mathbf{B}^i + F(\rho_i), \tag{9}$$

where $F(\rho_i)$ represents the energy of embedding atom i into the electron density contributed by its neighboring atoms. Just as ρ_i is written as a sum of pair contributions in EAM, we write it here as a linear function of the bispectrum components,

$$\rho_i = \mathbf{a} \cdot \mathbf{B}^i. \tag{10}$$

We express the embedding energy as a Taylor expansion about some reference structure with density ρ_0,

$$F(\rho) = F_0 + (\rho - \rho_0)F' + \frac{1}{2}(\rho - \rho_0)^2 F'' + \cdots, \tag{11}$$

where F_0, F', and F'' are constant values equal to the embedding function and its first and second derivatives at the reference density ρ_0, respectively. The constant and linear contributions to the embedding energy can be written as equivalent contributions to the linear SNAP energy. This is directly analogous to how the linear contributions to the EAM embedding energy can be written in terms of an equivalent pair potential.[13] Substituting the quadratic term into Eq. (9) we find

$$E^i_{SNAP}(\mathbf{r}^N) = \boldsymbol{\beta}\cdot\mathbf{B}^i + \frac{1}{2}F''(\mathbf{a}\cdot\mathbf{B}^i)^2, \qquad (12)$$

$$= \boldsymbol{\beta}\cdot\mathbf{B}^i + \frac{1}{2}(\mathbf{B}^i)^T\cdot\boldsymbol{\alpha}\cdot\mathbf{B}^i, \qquad (13)$$

where $\boldsymbol{\alpha} = F''\mathbf{a}\otimes\mathbf{a}$ is a symmetric $K \times K$ matrix. Equation (13) is simply the extension of the linear SNAP form to include all distinct pairwise products of bispectrum components and we refer to it as the quadratic SNAP model.

In comparing different interatomic potential forms, it is instructive to consider their "bodiedness" or maximum complexity of the energy expressions generated. Simple interatomic potentials are often expressed as truncated cluster expansions, in which the total energy is decomposed into sums over terms depending on scalar distance between pairs of atoms (two-body), angle formed by triplets of atoms (three-body), dihedral angle formed by four atoms, and so on. This idea can be extended to more complex potential by first fully expanding the total potential energy expression into a sum of individual terms and then considering the maximum number of atom positions appearing in a single term. Performing this exercise for the linear SNAP form requires fully expanding the sums in Eqs. (4) and (5). For the SNAP energy of a particular atom i we obtain individual terms of the form

$$U^{j_1*}_{m_1,m_1'}(\mathbf{r}_{i_1}-\mathbf{r}_i)U^{j_2}_{m_2,m_2'}(\mathbf{r}_{i_2}-\mathbf{r}_i)U^{j_3}_{m_3,m_3'}(\mathbf{r}_{i_3}-\mathbf{r}_i),$$

where $\mathbf{r}_i$, $\mathbf{r}_{i_1}$, $\mathbf{r}_{i_2}$, and $\mathbf{r}_{i_3}$ are the positions of atom i and three neighbors i_1, i_2, and i_3, respectively. The associated prefactors of $f_c(r)$ and w_i have been omitted, and the mapping from $\mathbf{r}$ to (θ_0, ϕ, θ) is not shown explicitly. This shows that the maximum number of distinct atom positions appearing in a single term is four. In the case of quadratic SNAP, the individual terms are pairwise products of these terms. Because both factors involve atom i, the maximum number of distinct atom positions appearing is seven (one central atom plus two sets of three distinct neighbor atoms). While the maximum complexity of the energy terms generated by linear SNAP is four-body, quadratic SNAP generates terms up to seven-body. It is unclear to what extent limiting the bodiedness of an interatomic potential fundamentally restricts its predictive power, but the results presented below suggest that there is a practical benefit to going being four-body terms. The total energy remains a linear function of $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ which means it can still be solved using linear regression methods as in Eq. (1).

The quadratic SNAP model can be viewed as a special case of an ANN whose inputs are the coordinates of neighbor atoms relative to each central atom. There are two hidden layers whose activation functions are the bispectrum components of each atom and the product of pairs of bispectrum components, respectively. The weights on the connections between layers depend on the values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. If $\boldsymbol{\alpha}$ is set to zero, then the two layers collapse into one and the linear SNAP form is recovered. Whereas a typical ANN uses two-body and three-body descriptors as inputs,[39] the four-body SNAP descriptors require fewer hidden layers to achieve good predictive accuracy. A clear benefit of quadratic SNAP over linear SNAP, similar to that general purpose ANN, is the introduction of nonlinear terms and a corresponding increase in the number of fitting coefficients, without expanding the size of the descriptor space.

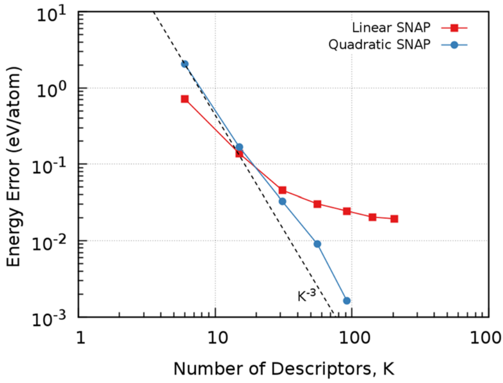

FIG. 2. Mean absolute energy error for SNAP models relative to the DFT training data for tantalum versus the number of descriptors K. The linear SNAP form (red squares) shows negligible improvements in accuracy beyond K = 30. In contrast, the quadratic SNAP form (blue squares) exhibits continuously improving accuracy, with error decreasing in proportion to $\sim K^{-3}$ (dashed line).

A demonstration of the accuracy gained from adding the quadratic SNAP terms is given in Fig. 2. For both the linear (red squares) and quadratic (blue circles) SNAP forms, the number of unique descriptors is the same for a given J_{max}, but there are significantly more coefficients to fit in the latter (see Table I). Figure 2 shows diminishing improvements in the accuracy of the linear SNAP form as the size of the descriptor space is increased. In contrast, the quadratic SNAP forms shows continued systematic improvement in accuracy over the same range,[60] with the energy error decreasing roughly in proportion to $\sim K^{-3}$. At J_{max} = 5 (K = 91), there is an order of magnitude better energy error for quadratic SNAP over the linear form. As shown in Fig. 1, this large improvement in accuracy between model forms is accompanied by less than a factor of two decrease in speed (atom timesteps/s). In Secs. V and VI, we will provide more details on how SNAP potentials are optimized, as well as a discussion of how to avoid pitfalls related to overfitting and extrapolation beyond the range of training data.

Before closing this section, it is worth mentioning that we see a natural extension of SNAP as a many-body correction to existing potentials. This could be anything from a simple Lennard-Jones[4] potential to a complex many-body potential such as MEAM[14] or COMB.[11] Existing empirical potentials could include an additional SNAP many-body correction, given some specific set of electronic structure data relevant to the current application. For example, an EAM potential for a metallic system could include a SNAP correction in order to capture the relative stability of metastable crystal phases that were previously poorly represented. This could also be done for semi-empirical methods such as tight-binding,[61] where many-body repulsions are empirically added to the total energy functional.

V. OPTIMIZATION METHODOLOGY

The need for a new IAP is driven by the scientific needs of a particular application. This means there exists a set of target properties for which the potential should be optimized. These quality metrics are material properties such as elastic constants, defect formation energies, melting temperature, and relative phase stability. We consider these externally imposed quality metrics as probes of how well the generated potential describes physically important configurations that may not be well-represented in the training data. In addition, a ML-IAP must also be optimized to predict energies and forces contained in the training data. A combination of these training and prediction error metrics constitute a set of objective functions that are used to optimize the SNAP potential. In this section, we describe in some detail the steps we take to optimize potentials using both categories of objective functions. While these steps are discussed, a direct comparison of the accuracy between linear and quadratic SNAP will be made for elemental tantalum.[19]

As was mentioned in Sec. III, a significant amount of expert opinion is needed in the generation of useful training data. In a computationally tractable way, DFT (or any high fidelity model) data that sample the most important regions of a material's PES must be generated. For the results presented here, we have used the previously generated training set for tantalum described in Ref. 19. This training set was constructed in order to train a SNAP potential to the energies and forces from DFT of both crystalline and disordered states. A full list of the different groups of training configurations is given in the legend of Fig. 4. The *Displaced A15, BCC, and FCC* groups correspond to randomly displaced atoms of a large (approx. 50 atoms) super cell of each crystal phase; these groups contribute 27 energy and 4482 force training points. Groups *Elastic BCC and FCC* are unit cells of either phase with the simulation cell perturbed from the equilibrium structure; these contribute 200 energies and 1200 forces in total. Similarly, *Volume A15, BCC, and FCC* are simply volumetric compressions and expansions of the equilibrium crystal structure; 81 energy and 1218 force training points are attributed to these groups. The *Liquid* group contributes 3 energy and 4203 force points; *Surface* group adds 7 energy and 630 forces. Lastly, the generalized stacking fault (*GSF*) *110 and 112* are generalized stacking fault calculations on the (110) and (112) crystallographic planes in BCC tantalum, another 44 energies and 3564 force training points. The defect-free training groups (*Displaced, Volume, and Elastic*) therefore contribute a total of 308 energies and 6900 forces compared to the disordered groups (*GSF, Liquid, and Surface*) which contribute 54 energies, and 8397 forces.

Once all of these energies and forces are assembled into the training set [Eq. (1), T], the atomic configurations need to be translated into the descriptor space (set of bispectrum components, D). However, there are a few hyper-parameters used in the fit that will affect D, namely the radial cutoff (r_{cut}) and the parameter J_{max} that determines the total number of bispectrum components used. Multi-element SNAP fits require an optimization of additional hyper-parameters in order to allow for distinct element types to contribute differently to the local density. However, the present application involves only one element, so the only hyper-parameter to search over is the radial cutoff. Figure 2 shows how the accuracy of the potentials increases with the number of bispectrum components. Note that the linear SNAP shows a diminishing improvement in reducing the energy errors as the size of the descriptor space increases. Because of this trend of diminishing returns, we fix $J_{max} = 4$ for the subsequent comparisons. It is our approach to first search for optimal hyper-parameters that minimize training errors. An example of this is shown in Fig. 3 where r_{cut} is varied for both linear and quadratic forms of the SNAP potential in the range 3-12 Å. For each r_{cut}, D is recalculated and Eq. (1) is solved using $\gamma_n = 0$ and the identity matrix for $\mathbf{w}$, giving the unweighted LLS solution for $\boldsymbol{\beta}$.

For all values of r_{cut}, the addition of quadratic terms resulted in lower regression errors in both energies and forces. This is expected, since the quadratic SNAP form includes $K(K+1)/2$ adjustable coefficients, in addition to the K coefficients of the linear form, where K is the number of bispectrum components used. While the force error is relatively insensitive to r_{cut}, there are multiple local minima in the energy errors. The globally optimal values of r_{cut} were found to be 5.672 Å and 5.594 Å for the linear and quadratic forms, respectively. Optimized radial cutoffs for each SNAP functional form and value of J_{max} are captured in Tables S1 and S2 of the supplementary material. In general, we found that the optimal radial cutoff increased with increasing J_{max} but was similar for both linear and quadratic forms.

Once the hyper-parameters affecting D have been chosen, the next step in the optimization process is focused on the predictive objective functions. Intuitively, not all of the training points will be equally important for these objective functions. For example, energies and forces of training configurations corresponding to the generalized stacking fault (GSF) in tantalum will help to accurately reproduce the Peierls stress but will not constrain the bulk modulus, since there is no volume change in these configurations. How much weight should be given to these training points relative to others is something

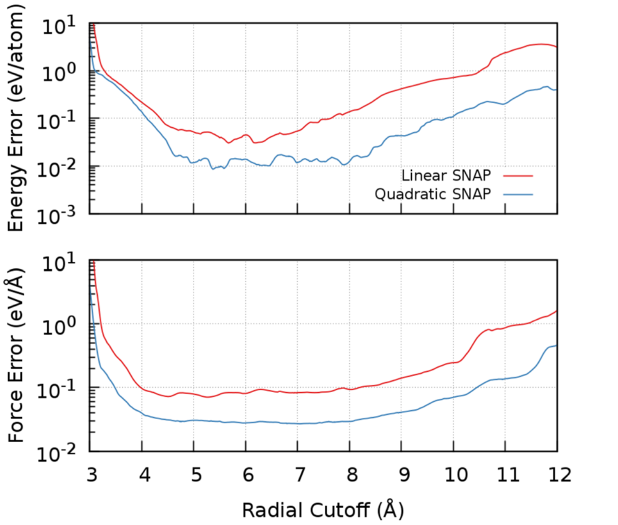

FIG. 3. Training errors for the linear (red) and quadratic (blue) SNAP forms for different values of the radial cutoff distance r_{cut}. **Top** Mean absolute error for energy data. **Bottom** Mean absolute error for force data. For both energy and force errors, the quadratic form of SNAP yields lower training errors than the linear form, while the size of the descriptor space is the same in both cases ($J_{max} = 4$).

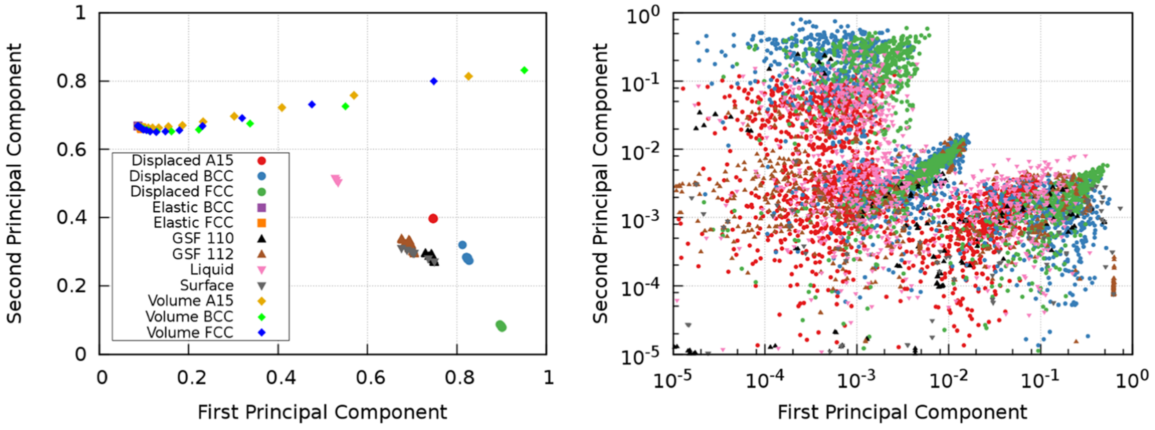

FIG. 4. **Left** Projection of the descriptor space for energy training data onto the first two principal components. **Right** Projection of the descriptor space for force training data onto the first two principal components.

that can be optimized by adjusting **w** of Eq. (1). In order to reduce the complexity of this optimization step, training data is sorted into a limited number of groups, and all rows in D belonging to a group are assigned the same weight. In addition, we also assign a different weight to energy and forces rows within a group. Although the number of free variables is doubled, this is necessary to prevent the LLS fit being dominated by force errors since there are far more rows in D for forces than energies.

It is by no means obvious how best to separate configurations into groups in order to perform this weighted regression. In Ref. 19, the training data was grouped according to the phase of tantalum as it was calculated within DFT. Additional subgroups were formed based on perturbations made to the structure. However, there is a more robust way to analyze the similarity of configurations and in turn efficiently assign group weights that aid in optimizing the potential. In Fig. 4, the matrix D is projected onto its first two principal components for either the energy (left panel) or force (right panel) rows. There is very little distinction between the training groups in terms of the forces. Significant scatter and overlaps are observed between all groups. In contrast, the first two energy principal components, left panel of Fig. 4, reveal unique features and similarities of the tantalum training set. Expectedly, the volumetric expansion and compression of the A15, BCC, and FCC phases (diamond symbols) cluster together, but liquid structures (pink triangles) are isolated from all other groups. Although the authors designated separate groups for either GSF and the free surface configurations, in the energy principal component projection of D, these groups do not seem unique. This similarity suggests that one can eliminate some free variables during optimization of the group weights.

Once training groups are defined, we employ a genetic algorithm (GA) within the DAKOTA[62] software package to search for optimal group weights. The quality of a fitted potential is determined by the objective functions specified for the particular application of interest. This is necessarily open-ended because the same training data can be used to fit many unique SNAP potentials when using a weighted LLS regression model. A variety of other approaches could be used to obtain optimal group weights. For example, Bayesian analysis could be used to infer the maximum likelihood values of the group weights. We do not expect the overall quality of the SNAP potential to be sensitive to the particular choice of optimization method. As the GA converges to some set of group weights for a set of objective functions, one can identify which training data was most important to the fit quality. Since the calculation of bispectrum components is the computational bottleneck with SNAP, this sequential approach to fitting hyper-parameters first, followed by group weights, is significantly more computationally efficient because D does not need to be recalculated for each iteration of the GA optimization loop. Being a gradient-free method, the GA never fully converges the group weights. Typically, we see little variation in fit quality after $\mathcal{O}(10^4)$ candidate potentials are generated. This convergence rate holds for both linear and quadratic forms of SNAP.

VI. PREDICTION BEYOND TRAINING

One of the primary concerns with large ANN potentials, which includes quadratic SNAP as a special case, is that the large number of available fitting coefficients may result in a ML-IAP that is overfit to the training data and objectives. In other words, while the ML-IAP may accurately reproduce data to which it was fit to or evaluated against, it may produce highly spurious results for other points in the descriptor space. A common way to test for overfitting is to withhold training data and compare the accuracy of the model for these out-of-sample structures (prediction errors) to the accuracy of the model for structures included in training (training errors). This is known as cross-validation and is a staple of robustness testing for ANN models.[40]

In the context of SNAP fitting, cross-validation is achieved simply by excluding a fixed fraction of training data from each training group. This includes both the energies and forces for an excluded configuration. We are primarily concerned with how much training data is needed to converge prediction errors for excluded data to the same level as the training errors for included data. To do so, we incrementally increase the fraction of training data included in 5% intervals, and at each fraction,

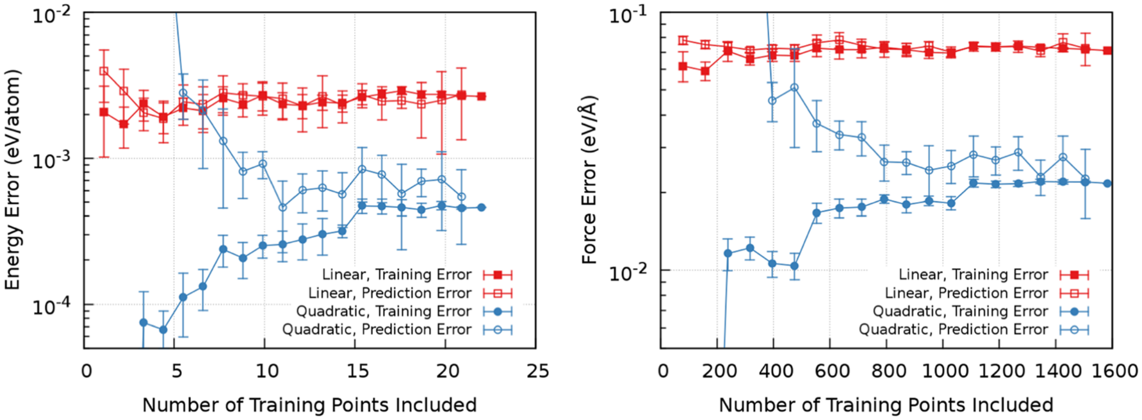

FIG. 5. Comparison of training errors (closed symbols) and prediction errors (open symbols) for the linear (red) and quadratic (blue) SNAP forms. Different proportions of randomly selected training data are excluded from the fitting process and are used to measure prediction error. Each point in these plots adds an additional 5% of the total training data; the last point includes all training points and as such has no prediction error to report. Error bars indicate standard deviation across 10 random draws. **Left** Mean absolute energy error for the GSF(110) structures. **Right** Mean absolute force error for the GSF(110) structures. While the quadratic SNAP errors are lower than the linear SNAP form, more training data is needed to fully converge the prediction errors. For training data fractions less than ~30%, prediction errors for the quadratic SNAP form are extremely large because there are fewer training points than fitting coefficients and the resultant potential is strongly overfit.

ten random selections of included data are generated for each group. Note that the energy and forces from a single DFT calculation cannot be split across the included/excluded sets. To exemplify this process, Fig. 5 shows training and prediction errors for the GSF(110) group. Error bars represent the standard deviation among the ten random draws at each fraction included. The GSF(110) group is made up of 22 unique DFT calculations each with 24 atoms, totaling 22 energy and 1584 force data points. The left panel of Fig. 5 represents the energy errors, while the right panel captures the force errors observed in this cross-validation.

Reiterating what was discussed earlier and shown in Figs. 2 and 3, while the quadratic SNAP form has the same number of bispectrum components, the overall errors are lower than the linear SNAP form. This is also true when training data is omitted as seen in Fig. 5. The prediction errors in quadratic SNAP are lower than linear SNAP after only 30% of the total data are used in fitting. However, the prediction errors do not converge to the regression errors as quickly for quadratic SNAP. This points to a need for larger training sets when exercising the expanded set of fitting coefficients available in quadratic SNAP in order to prevent overfit solutions. Interestingly, this convergence rate of cross-validation errors varies between training groups. More complex configurations such as GSF, surfaces, and liquids require more training data in order to make accurate predictions for new configurations. Simpler groups, such as high symmetry crystal structures, make excellent predictions with very little training.

We see this variation in data requirements as an effect of the chosen descriptor space. From our experience the bispectrum components easily resolve areas of the PES related to crystalline phases but require more training in disordered and over/under-coordinated atomic environments. Unsurprisingly, these types of training data that we identify as important are also computationally the most costly to acquire from DFT. This cross-validation, in conjunction with the principal component analysis, could be done prior to group weight optimization and be used as feedback where additional DFT calculations should be made. Therefore, the training set construction is done in a way that expands D in a physically meaningful way to improve the robustness of the resultant potential.

VII. CONCLUSIONS

The promise of machine learning for interatomic potentials is enormous. This is due to the fact that once a methodology has been developed and validated for one particular material or chemical system, it can easily be extended to many more systems, subject only to the availability of appropriate high accuracy training data. This is in stark contrast to traditional interatomic potentials, where successful development of an IAP for one particular system is not easily extended to others. For this reason, there is significant activity within the computational physics, chemistry, and materials science communities to not only generate new ML-IAPs for a range of materials and applications but also develop new descriptors and representations that are capable of robustly describing atomic potential energy landscapes with an accuracy approaching that of the underlying quantum electronic structure calculations. However, the common pitfall that plagues many ML-IAPs is their inability to make accurate predictions for configurations that were not explicitly included in the training data.

We have shown that augmenting the SNAP potential to include quadratic terms in the bispectrum components of the neighbor density transforms SNAP from a four-body to a seven-body potential and in turn is able to more accurately capture energies and forces from a diverse training set. Interestingly, in the context of Gaussian process prediction of forces in crystalline silicon structures, Glielmo *et al.* recently demonstrated a similar improvement when replacing a two-body linear kernel with a three-body quadratic kernel[63] The quadratic SNAP form can be seen as a special case of an ANN with two hidden layers, while the linear SNAP form has only one hidden layer. Similar to ANN potentials, the quadratic SNAP form requires substantially more training data in order to prevent overfitting, as demonstrated using cross-validation analysis with an extensive set of DFT training data for tantalum structures. This large accuracy improvement, an order of magnitude reduction for training error, incurs less than a twofold increase in computational cost.

SUPPLEMENTARY MATERIAL

See supplementary material for numerical values of the optimal cutoffs, along with the energy errors, for both the linear and quadratic SNAP potentials for each value of J_{max}.

ACKNOWLEDGMENTS

We would like to thank Art Voter, Danny Perez, Tom Lenosky, and Alex Shapeev for their fruitful discussions relevant to this manuscript. This research was supported by the Exascale Computing Project (No. 17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under Contract No. DE-NA0003525.

[1] M. Griebel, S. Knapek, and G. Zumbusch, *Numerical Simulation in Molecular Dynamics: Numerics, Algorithms, Parallelization, Applications (Texts in Computational Science and Engineering)* (Springer, 2007), pp. 1–3.

[2] M. Finnis, *Interatomic Forces in Condensed Matter* (Oxford University Press, Oxford, 2003), Vol. 1.

[3] D. W. Brenner, Phys. Status Solidi B **217**(1), 23–40 (2000).

[4] J. E. Lennard-Jones, Proc. Phys. Soc. **43**(5), 461 (1931).

[5] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard III, and W. M. Skiff, J. Am. Chem. Soc. **114**(25), 10024–10035 (1992).

[6] S. J. Plimpton and A. P. Thompson, MRS Bull. **37**(5), 513–521 (2012).

[7] A. C. T. van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard III, J. Phys. Chem. A **105**(41), 9396–9409 (2001).

[8] J. Tersoff, Phys. Rev. B **38**(14), 9902 (1988).

[9] D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, and S. B. Sinnott, J. Phys.: Condens. Matter **14**(4), 783 (2002).

[10] A. K. Rappe and W. A. Goddard III, J. Phys. Chem. **95**(8), 3358–3363 (1991).

[11] T.-R. Shan, B. D. Devine, T. W. Kemper, S. B. Sinnott, and S. R. Phillpot, Phys. Rev. B **81**(12), 125328 (2010).

[12] M. S. Daw and M. Baskes, Phys. Rev. Lett. **50**(17), 1285 (1983).

[13] M. S. Daw and M. I. Baskes, Phys. Rev. B **29**, 6443 (1984).

[14] M. I. Baskes, Phys. Rev. B **46**(5), 2727 (1992).

[15] T. P. Senftle, S. Hong, M. M. Islam, S. B. Kylasa, Y. Zheng, Y. K. Shin, C. Junkermeier, R. Engel-Herbert, M. Janik, H. M. Aktulga, T. Verstraelen, A. Grama, and A. C. T. van Duin, npj Comput. Mater. **2**, 15011 (2016).

[16] S. J. Stuart, A. B. Tutein, and J. A. Harrison, J. Chem. Phys. **112**(14), 6472–6486 (2000).

[17] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**(14), 146401 (2007).

[18] J. Behler, J. Chem. Phys. **145**(17), 170901 (2016).

[19] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, J. Comput. Phys. **285**, 316–330 (2015).

[20] G. Ferre, J.-B. Maillet, and G. Stoltz, J. Chem. Phys. **143**, 104114 (2015).

[21] R. Kobayashi, D. Giofré, T. Junge, M. Ceriotti, and W. A. Curtin, Phys. Rev. Mater. **1**(5), 053604 (2017).

[22] A. Takahashi, A. Seko, and I. Tanaka, Phys. Rev. Mater. **1**(6), 063801 (2017).

[23] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**(18), 184115 (2013).

[24] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Phys. Rev. Lett. **108**(5), 058301 (2012).

[25] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, New J. Phys. **15**(9), 095003 (2013).

[26] W. L. Szlachta, A. Bartók, and G. Csayni, Phys. Rev. B **90**, 104108 (2014).

[27] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, and A. Knoll, Int. J. Quantum Chem. **115**(16), 1084–1093 (2015).

[28] E. V. Podryabinkin and A. V. Shapeev, Comput. Mater. Sci. **140**, 171–180 (2017).

[29] A. P. Bartók and G. Csányi, Int. J. Quantum Chem. **115**(16), 1051–1057 (2015).

[30] A. Bartók, M. Payne, R. Kondor, and G. Csayni, Phys. Rev. Lett. **104**, 136403 (2010).

[31] A. Bartók, Ph.D. thesis, Cambridge University, 2009; e-print arXiv:1003.2817.

[32] C. Chen, Z. Deng, R. Tran, H. Tang, I.-H. Chu, and S. P. Ong, Phys. Rev. Mater. **1**(4), 043603 (2017).

[33] J. Shweta and S. S. Bulusu, Chem. Phys. Lett. **693**, 152–158 (2018).

[34] G. Csányi, T. Albaret, M. C. Payne, and A. De Vita, Phys. Rev. Lett. **93**(17), 175503 (2004).

[35] Z. Li, J. R. Kermode, and A. De Vita, Phys. Rev. Lett. **114**(9), 096405 (2015).

[36] V. L. Deringer and G. Csányi, Phys. Rev. B **95**, 094203 (2017).

[37] C. M. Bishop, *Neural Networks for Pattern Recognition* (Oxford University Press, 1995).

[38] J. Behler, Phys. Chem. Chem. Phys. **13**(40), 17930–17955 (2011).

[39] J. Behler, J. Phys.: Condens. Matter **26**(18), 183001 (2014).

[40] N. Lubbers, J. S. Smith, and K. Barros, preprint arXiv:1710.00017 (2017).

[41] J. S. Smith, O. Isayev, and A. E. Roitberg, Chem. Sci. **8**(4), 3192–3203 (2017).

[42] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, Phys. Chem. Chem. Phys. **18**(20), 13754–13769 (2016).

[43] J. Barker, J. Bulin, J. Hamaekers, and S. Mathias, *Scientific Computing and Algorithms in Industrial Simulations* (Springer Verlag, 2017), pp. 25–42.

[44] J. Ischtwan and M. A. Collins, J. Chem. Phys. **100**(11), 8080–8088 (1994).

[45] C. Crespos, M. A. Collins, E. Pijper, and G. J. Kroes, Chem. Phys. Lett. **376**(5), 566–575 (2003).

[46] C. Crespos, M. A. Collins, E. Pijper, and G. J. Kroes, J. Chem. Phys. **120**(5), 2392–2404 (2004).

[47] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2nd ed. (Springer, New York, 2007).

[48] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, 2006).

[49] R. Tibshirani, J. R. Stat. Soc. B (Methodol.) **58**, 267–288 (1996).

[50] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola, *Numerical Methods for the Solution of III-Posed Problems* (Springer Science & Business Media, 2013), Vol. 328.

[51] J. P. Biersack and J. F. Ziegler, *Ion Implantation Techniques* (Springer, Berlin, Heidelberg, 1982), pp. 122–156.

[52] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii, *Quantum Theory of Angular Momentum* (World Scientific, Singapore, 1987).

[53] C. R. Trott, S. D. Hammond, and A. P. Thompson, in *Proceedings of International Supercomputing Conference*, Lecture Notes in Computer Science (Springer Verlag, 2014), Vol. 8488, p. 19.

[54] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[55] See http://lammps.sandia.gov for LAMMPS.

[56] Documentation for SNAP in LAMMPS can be found at http://lammps.sandia.gov/doc/pair_snap.html. Working examples of molecular dynamics simulations using SNAP potentials are provided in the LAMMPS public download under examples/snap.

[57] D. Scheiber, R. Pippan, P. Pushnig, and L. Romaner, Modell. Simul. Mater. Sci. Eng. **24**, 035013 (2016).

[58] M. A. Wood and A. P. Thompson, preprint arXiv:1702.07042 (2017).

[59] The QUIP atomistic simulation software package is available from https://github.com/libAtoms/QUIP.

[60] For each point in Fig. 2, the radial cutoff was independently optimized on the tantalum training set. The numerical values of the optimal cutoffs, along with the energy errors, are given in Tables S1 and S2 of the supplementary material.

[61] M. J. Cawkwell and A. M. N. Niklasson, J. Chem. Phys. **137**(13), 134105 (2012).

[62] M. Eldred *et al.*, DAKOTA Version 4.0 Users Manual, Technical Report SAND 2006–6337, Sandia National Laboratories, 2006.

[63] A. Glielmo, P. Sollich, and A. De Vita, Phys. Rev. B **95**, 214302 (2017).