Journal of Applied Physics

Spectral neural network potentials for binary alloys

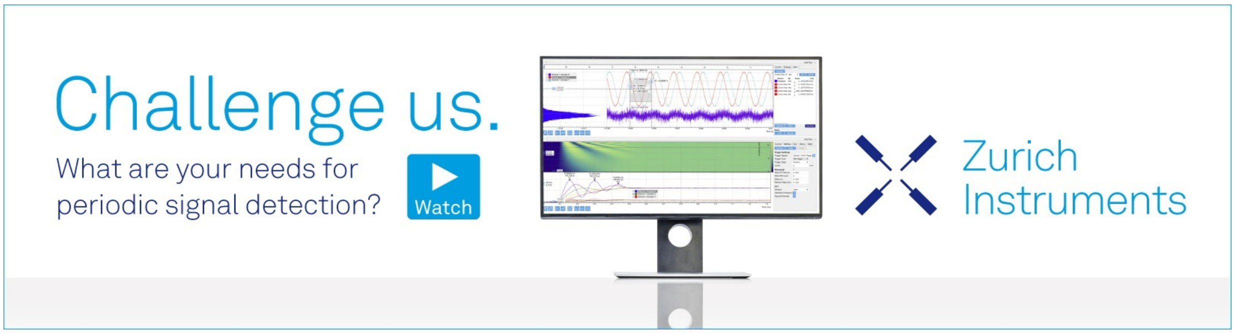
Cite as: J. Appl. Phys. **128**, 045113 (2020); https://doi.org/10.1063/5.0013208
Submitted: 11 May 2020 . Accepted: 09 July 2020 . Published Online: 28 July 2020

David Zagaceta, Howard Yanxon, and Qiang Zhu

COLLECTIONS

Paper published as part of the special topic on Machine Learning for Materials Design and Discovery

View Online Export Citation CrossMark



AIP Publishing

Spectral neural network potentials for binary alloys

Cite as: J. Appl. Phys. **128**, 045113 (2020); doi: 10.1063/5.0013208
Submitted: 11 May 2020 · Accepted: 9 July 2020 ·
Published Online: 28 July 2020 · Corrected: 31 July 2020

View Online Export Citation CrossMark

David Zagaceta, Howard Yanxon, and Qiang Zhu[a)]

AFFILIATIONS

Department of Physics and Astronomy, University of Nevada, Las Vegas, Nevada 89154, USA

Note: This paper is part of the special collection on Machine Learning for Materials Design and Discovery

[a)]**Author to whom correspondence should be addressed:** qiang.zhu@unlv.edu

ABSTRACT

In this work, we present a numerical implementation to compute the atom-centered descriptors introduced by Bartok *et al.* [Phys. Rev. B **87**, 184115 (2013)] based on the harmonic analysis of the atomic neighbor density function. Specifically, we focus on two types of descriptors, the smooth SO(3) power spectrum with the explicit inclusion of a radial basis and the SO(4) bispectrum obtained through mapping the radial component onto a polar angle of a four dimensional hypersphere. With these descriptors, various interatomic potentials for binary Ni–Mo alloys are obtained based on linear and neural network regression models. Numerical experiments suggest that both descriptors produce similar results in terms of accuracy. For linear regression, the smooth SO(3) power spectrum is superior to the SO(4) bispectrum when a large band limit is used. In neural network regression, better accuracy can be achieved with even less number of expansion components for both descriptors. As such, we demonstrate that spectral neural network potentials are feasible choices for large scale atomistic simulations.

I. INTRODUCTION

The development of accurate and efficient interatomic potentials is a central issue critical in many areas of modern chemical physics. Although *ab initio* methods such as the Kohn–Sham density functional theory (DFT)[1] are accurate and transferable, they are also costly and, therefore, limited to applications to systems consisting of only a few thousand atoms. On the other hand, empirical force fields are able to handle systems of a much larger scale, although accuracy is generally problematic. As a consequence, there has been a substantial effort in the last decade to develop efficient and accurate interatomic potentials using machine learning.[2]

The development of machine learning interatomic potentials (MLIAPs) has been primarily focused on *feature engineering*, i.e., a numerical descriptor used to represent the local chemical environment for each atomic structure. A representation of a chemical environment should be real-valued, unique, invariant to rotation of the system, translation of the system, and permutation of homonuclear atoms.[2,3] Several representations satisfying these conditions are widely used in fitting MLIAPs, examples of which are smooth overlap of atomic positions (SOAPs),[3] atom-centered symmetry functions (ACSFs),[4] moment tensor potentials (MTPs),[5] and spectral neighbor analysis potential (SNAP).[6] Potentials are constructed from these representations through machine learning on *ab initio* data using regression methods such as generalized linear regression, artificial neural networks, and Gaussian process regression. Notable potentials include the SNAP method, which is constructed from the SO(4) bispectrum components and fit using either a linear or quadratic regression,[6,7] Gaussian approximation potentials (GAPs) constructed using SOAP with Gaussian process regression,[8] and high-dimensional neural network (NN) potentials constructed using atom-centered symmetry functions with an artificial neural network.[9] For a comprehensive review on descriptor construction and machine learning, please refer to recent literature studies.[10–13]

Recently, we demonstrated that neural network potentials (NNPs) constructed using the SO(4) bispectrum components as the descriptor can achieve good transferability on a rather diverse set of atomic configurations obtained from randomly generated crystalline silicon structures.[14] In this work, we aim to extend the capability of the NNPs based on the SO(4) bispectrum and smooth SO(3) power spectrum to multicomponent systems as well as provide a comprehensive study of the performance of the SO(4) bispectrum components and the smooth SO(3) power spectrum components as descriptors.[3,6,15] First, we will review some particular representations of chemical environments related to the harmonic analysis of the atomic neighbor density function, with emphasis on the SO(3) power spectrum and SO(4) bispectrum components. In particular, we introduce a different numerical method to compute these descriptors. This is followed by a brief discussion on the regression methods used in this study. Finally, we apply our approach to a binary-component system Ni–Mo. The code that is used in this study is available on https://github.com/qzhu2017/PyXtal_FF.

II. CHEMICAL ENVIRONMENT REPRESENTATIONS

A representation of a chemical environment can be considered as a quantitative measure of atomic correlation, or rather, an order parameter, being invariant to translations and rotations of the system as well as permutations of homonuclear atoms. First, notice that the spatial distribution of atoms in a chemical environment, up to a cutoff radius (r_{cut}), can be represented by a sum of δ functions,

$$\rho(\boldsymbol{r}) = \sum_{i}^{r_i \leq r_{\text{cut}}} \delta(\boldsymbol{r} - \boldsymbol{r_i}). \tag{1}$$

This is referred to as the atomic neighbor density function.[3] The distribution of atoms described by the atomic neighbor density function is not particularly useful by itself, a more useful description of the chemical environment is the angular distribution of atoms in the environment obtained through expanding $\rho(\boldsymbol{r})$ as a series on the 2-sphere using spherical harmonics,

$$\rho(\boldsymbol{r}) = \sum_{l=0}^{+\infty} \sum_{m=-l}^{+l} c_{lm} Y_{lm}(\hat{\boldsymbol{r}}),$$

where the expansion coefficients c_{lm} are given by

$$c_{lm} = < Y_{lm}(\hat{\boldsymbol{r}}) | \rho(\boldsymbol{r}) > = \sum_{i}^{r_i \leq r_{\text{cut}}} Y_{lm}(\hat{\boldsymbol{r}}_{\mathbf{i}}). \tag{2}$$

For simplicity, we use $\sum_{lm}$ to denote the double summation over l and m from now on. Several representations have been constructed using these expansion coefficients. Steinhardt constructed his bond order parameters using second and third order combinations of the expansion coefficients [Eq. (2)] to quantify order in liquids and glasses.[16] More generally, Kondor constructed an SO(3)-invariant kernel on the 2-sphere using the expansion coefficients of a function defined on the 2-sphere; this kernel provides a method of calculating both the power spectrum and the bispectrum of a function on the 2-sphere.[15] The SO(3)-invariant power spectrum of Eq. (1) is constructed through taking the autocorrelation of the sequence of expansion coefficients in Eq. (2),

$$p_l = \sum_{m=-l}^{+l} c_{lm} c_{lm}^{*}. \tag{3}$$

Though p_l from Eq. (3) satisfies the necessary conditions to represent a chemical environment, it does not carry sufficient information to be useful due to the fact that the expansion coefficients in Eq. (2) would not carry any radial information. For better application to MLIAPs, Bartok introduced two modifications[3] as follows.

A. Smooth SO(3) power spectrum with explicit radial components

The first modification is to add radial information by expanding ρ not only as a series on the 2-sphere but on a radial basis simultaneously. In the second modification, to ensure a smooth similarity kernel, Bartok[3] also expanded Eq. (1) using Gaussians,

$$\rho'(\boldsymbol{r}) = \sum_{i}^{r_i \leq r_{\text{cut}}} \exp(-\alpha |\boldsymbol{r} - \boldsymbol{r_i}|^2), \tag{4}$$

Then, expanding Eq. (4) on the 2-sphere yields

$$\begin{aligned} \rho'(\boldsymbol{r}) &= \sum_{r_i \leq r_{\text{cut}}} e^{-\alpha(r^2 + r_i^2)} e^{2\alpha \boldsymbol{r} \cdot \boldsymbol{r_i}} \\ &= \sum_{r_i \leq r_{\text{cut}}} \sum_{lm} 4\pi e^{-\alpha(r^2 + r_i^2)} I_l(2\alpha r r_i) Y_{lm}^{*}(\hat{\boldsymbol{r}}_{\mathbf{i}}) Y_{lm}(\hat{\boldsymbol{r}}), \end{aligned}$$

where I_l is a modified spherical Bessel function of the first kind. The second equation is derived through a spherical harmonic transform of $e^{2\alpha \boldsymbol{r} \cdot \boldsymbol{r_i}}$.

Radial information can be explicitly added to the representation. A convenient radial basis for this purpose proposed by Bartok consists of cubic and higher order polynomials,[3] $g_n(r)$, orthonormalized on the interval (0, r_{cut}), while also vanishing at r_{cut},

$$\phi_k(r) = (r_{\text{cut}} - r)^{k+2} / N_k,$$

where

$$\begin{aligned} N_k &= \sqrt{\int_0^{r_{\text{cut}}} r^2 (r_{\text{cut}} - r)^{2(k+2)}\, dr} \\ &= \sqrt{\frac{2 r_{\text{cut}}^{(2k+7)}}{(2k+5)(2k+6)(2k+7)}}. \end{aligned}$$

Then, orthonormalizing linear combinations of ϕ_k from ϕ_1 up to ϕ_{nmax},

$$g_n(r) = \sum_{k=1}^{n_{\max}} \boldsymbol{W}_{nk} \phi_k(r), \tag{5}$$

where $\boldsymbol{W}$ is constructed from the overlap matrix $\boldsymbol{S}$ by the relation $\boldsymbol{W} = \boldsymbol{S}^{-1/2}$. The overlap matrix is given by the inner product[3]

$$\begin{aligned} S_{pq} &= \int_0^{r_{\text{cut}}} r^2 \phi_p(r) \phi_q(r)\, dr \\ &= \frac{\sqrt{(2p+5)(2p+6)(2p+7)(2q+5)(2q+6)(2q+7)}}{(5+p+q)(6+p+q)(7+p+q)}. \end{aligned} \tag{6}$$

In their original work,[3] Bartok *et al.* omitted the r^2 term in the integrand of N_k and S_{pq}. We included this term to explicitly orthonormalize the radial basis in the spherical polar coordinate system.

Then, expanding $\rho'(\boldsymbol{r})$ on the 2-sphere and radial basis $g(r)$ in Eq. (5), the new expansion coefficients are given by Ref. 3 as

$$\begin{aligned} c_{nlm} &= < g_n(r) Y_{lm}(\hat{\boldsymbol{r}}) | \rho'(\boldsymbol{r}) > \\ &= 4\pi \sum_{i}^{r_i \le r_{\text{cut}}} e^{-\alpha r_i^2} Y_{lm}^{*}(\hat{\boldsymbol{r}}_{\mathbf{i}}) \int_0^{r_{\text{cut}}} r^2 g_n(r)\, e^{-\alpha r^2} I_l(2\alpha r r_i)\, dr. \end{aligned} \tag{7}$$

The power spectrum components then follow similarly to Eq. (3),

$$p_{n_1 n_2 l} = \sum_{m=-l}^{+l} c_{n_1 l m} c^{*}_{n_2 l m}. \tag{8}$$

Note that Bartok further constructed the SOAP kernel to measure the similarity between two chemical environments for Gaussian Process Regression.[3] For our purpose, we do not utilize the SOAP kernel itself but use the smooth SO(3) power spectrum as a descriptor for MLIAPs.

B. SO(4) bispectrum components

An alternative approach to include radial information is to map the atomic neighbor density function within a cutoff radius r_{cut} onto the surface of the four dimensional hypersphere (3-sphere) with a radius of r_0 based on the following relations:[3,6]

$$\begin{aligned} s_1 &= r_0 \cos\omega, \\ s_2 &= r_0 \sin\omega \cos\theta, \\ s_3 &= r_0 \sin\omega \sin\theta \cos\phi, \\ s_4 &= r_0 \sin\omega \sin\theta \sin\phi, \end{aligned}$$

where the polar angles are defined by

$$\begin{aligned} \theta &= \arccos\left(\frac{z}{r}\right), \\ \phi &= \arctan\left(\frac{y}{x}\right), \\ \omega &= \frac{\pi r}{r_0}. \end{aligned} \tag{9}$$

Then, to ensure that the contribution from atoms at $r = r_{\text{cut}}$ smoothly goes to zero, it is necessary to augment the atomic neighbor density function [Eq. (1)] with a cutoff function,[6] while also including the center atom to avoid unphysical invariance with respect to ω,[3]

$$\rho(\boldsymbol{r}) = \delta(\boldsymbol{r}) + \sum_i f_{\text{cut}}(r)\delta(\boldsymbol{r} - \boldsymbol{r}_i), \tag{10}$$

where the cutoff function is defined as[6]

$$f_{\text{cut}}(r) = \begin{cases} \frac{1}{2}\left[\cos\left(\frac{\pi r}{r_{\text{cut}}}\right) + 1\right], & r \le r_{\text{cut}}, \\ 0, & r > r_{\text{cut}}. \end{cases} \tag{11}$$

To ensure that the mapping produces, a one-to-one function defined on the 3-sphere, r_0 has to be no smaller than r_{cut}. For convenience, we simply choose $r_0 = r_{\text{cut}}$ to map the atomic neighbor density function onto the entire 3-sphere.

Now, the atomic neighbor density function mapped onto the 3-sphere by Eq. (9) can be represented in an expansion of Wigner-*D* matrix elements in the angle-axis representation, where 2ω is the rotation angle and θ, ϕ define the axis,

$$\rho(\boldsymbol{r}) = \sum_{j=0}^{+\infty} \sum_{m',m=-j}^{+j} c^{j}_{m',m} D^{j}_{m',m}(2\omega;\, \theta,\, \phi).$$

The Wigner-*D* matrix elements are mutually orthogonal over the double volume of SO(3) and conveniently the area measure of the 3-sphere corresponds to exactly that (in the angle-axis representation).[17] Therefore, the expansion coefficients are obtained by the inner product[3]

$$\begin{aligned} c^{j}_{m',m} &= < D^{j}_{m',m} | \rho > \\ &= \int_0^{\pi} d\omega \sin^2\omega \int_0^{\pi} d\theta \sin\theta \int_0^{2\pi} d\phi D^{*j}_{m',m}(2\omega;\, \theta,\, \phi)\rho(\boldsymbol{r}) \\ &= D^{*j}_{m',m}(\mathbf{0}) + \sum_i f_{\text{cut}}(r_i) D^{*j}_{m',m}(\boldsymbol{r}_i). \end{aligned} \tag{12}$$

To obtain the bispectrum components, the triple-correlation of the expansion coefficients is used.[15] The result of which is shown as follows:

$$B_{j_1,j_2,j} = \sum_{m',m=-j}^{+j} c^{*j}_{m',m} \sum_{m'_1,m_1=-j_1}^{+j_1} c^{j_1}_{m'_1,m_1} \sum_{m'_2,m_2=-j_2}^{+j_2} c^{j_2}_{m'_2,m_2} C^{jj_1j_2}_{mm_1m_2} C^{jj_1j_2}_{m'm'_1m'_2}, \tag{13}$$

where C is a Clebsch–Gordan[17] coefficient.

III. NUMERICAL IMPLEMENTATION

To fit a MLIAP, both the representation and its gradient are needed. The SO(4) bispectrum components, the smooth SO(3) power spectrum components, and their gradients are implemented in our in-house software package PyXtal-FF. Although most of the calculations are straightforward as described Sec. II, we will discuss the necessary details where the calculations are nontrivial.

A. Bispectrum

For the SO(4) bispectrum components, we need to calculate the Wigner-*D* matrices for each neighbor. Here, we use a polynomial form of the Wigner-*D* matrix elements suggested by Boyle,[18]

$$\begin{cases} D^j_{m',m} = (-1)^{(j+m)} R_b^{2m} \delta_{-m',m}, & |R_a| < 10^{-15}, \\ D^j_{m',m} = R_a^{2m} \delta_{m',m}, & |R_b| < 10^{-15}, \\ D^j_{m',m} = \sqrt{\frac{(j+m)!(j-m)!}{(j+m')!(j-m')!}} |R_a|^{2j-2m} R_a^{m'+m} R_b^{-m'+m} \times \sum_k \binom{j+m'}{k} \binom{j-m'}{j-m-k} \left(-\frac{|R_b^2|}{|R_a^2|}\right)^k, & |R_a| \geq |R_b|, \\ D^j_{m',m} = (-1)^{j-m} \sqrt{\frac{(j+m)!(j-m)!}{(j+m')!(j-m')!}} R_a^{m'+m} R_b^{m-m'} |R_b|^{2j-2m} \times \sum_k \binom{j+m'}{j-m-k} \binom{j-m'}{k} \left(-\frac{|R_a^2|}{|R_b^2|}\right)^k, & |R_a| < |R_b|, \end{cases} \tag{14}$$

where R_a and R_b are the Cayley–Klein parameters representing the rotation. In the angle-axis representation of rotation, the Cayley–Klein parameters representing a rotation about an axis defined by $\boldsymbol{r} = (x, y, z)$ through an angle ω can be written as

$$\begin{aligned} R_a &= \cos(\omega/2) + i\frac{\sin(\omega/2)}{r} z, \\ R_b &= \frac{\sin(\omega/2)}{r}(y + ix). \end{aligned} \tag{15}$$

These polynomials are finite and the coefficients of each term are known. Different from previous works[3,6] based on a recursive scheme as discussed in the Appendix, we evaluate the Wigner-D matrix elements using Horner's method for the terms in the summation, which allows evaluation of a polynomial of degree n with only n multiplications and n additions,

$$\begin{aligned} P(x) &= a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \\ &= a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + x a_n))). \end{aligned} \tag{16}$$

Using Horner's method is also convenient for the simultaneous computation of the gradient. To obtain the gradient with respect to Cartesian coordinates, the chain rule is applied through the Cayley–Klein parameters and their conjugates.

In addition, we make use of the symmetries of the SO(4) bispectrum components discovered by Thompson,[6]

$$\frac{B_{j_1 j_2 j}}{2j+1} = \frac{B_{j j_2 j_1}}{2j_1+1} = \frac{B_{j_1 j j_2}}{2j_2+1}. \tag{17}$$

These symmetries reduce the number of necessary bispectrum components to compute to only the unique components, which also greatly reduces the complexity of the gradient calculation. For brevity, we denote the two inner sums of the bispectrum component calculation as $Z^{m,m'}_{j_1,j_2,j}$,[6]

$$\sum_{m_1,m_1'=-j_1}^{j_1} \sum_{m_2,m_2'=-j_2}^{j_2} c^{j_1}_{m_1',m_1} c^{j_2}_{m_2',m_2} C^{jj_1j_2}_{mm_1m_2} C^{jj_1j_2}_{m'm_1'm_2'}. \tag{18}$$

So that, when utilizing the symmetries in Eq. (17), the gradient of the bispectrum components with respect to an atom i can be written as[6]

$$\begin{aligned} \nabla_i B^{(i)}_{j_1,j_2,j} = & \sum_{m,m'=-j}^{j} \nabla_i \left(c^j_{m',m}\right)^* Z^{m,m'}_{j_1,j_2,j} \\ & + \frac{2j+1}{2j_1+1} \sum_{m_1,m_1'=-j_1}^{j_1} \nabla_i \left(c^{j_1}_{m_1',m_1}\right)^* Z^{m_1,m_1'}_{j,j_2,j_1} \\ & + \frac{2j+1}{2j_2+1} \sum_{m_2,m_2'=-j_2}^{j_2} \nabla_i \left(c^{j_2}_{m_2',m_2}\right)^* Z^{m_2,m_2'}_{j_1,j,j_2}, \end{aligned} \tag{19}$$

where the gradient of the inner product with respect to one atom is

$$\nabla_i c^j_{m',m} = \nabla_i \left(f_{\text{cut}}(r_i) D^{*j}_{m',m}(\boldsymbol{r}_i) \right). \tag{20}$$

B. Smooth SO(3) power spectrum

In calculating the smooth SO(3) power spectrum, three main challenges exist. First, the calculation of the spherical harmonics, and second the radial inner product in Eq. (7), and third the gradient of the expansion coefficients in Eq. (7). To start, the spherical harmonics can be considered as a subset of the Wigner-D matrices in the z–y–z Euler-angle representation, where the spherical harmonic vector $\boldsymbol{Y_l}$ is a row vector of the corresponding D-matrix $\boldsymbol{D^l}$ with some additional scalar factors as given in the equation below[17]

$$Y_{lm}(\theta, \phi) = (-1)^m \sqrt{\frac{2l+1}{4\pi}} D^l_{0,-m}(\chi, \theta, \phi),$$

where the Wigner-D matrices are in the z–y–z Euler-angle representation and χ is arbitrary, thus, without loss of generality, we choose $\chi = 0$.

The z–y–z Euler-angle representation represents a rotation about the original z axis through an angle α, then a rotation about the new y axis through an angle β, and then a rotation about the new z axis through an angle γ, which we can parameterize through composing rotations using the Cayley–Klein parameters in the angle-axis representation. For the case of calculating spherical harmonics and choosing $\chi = 0$, we have a rotation about the original y axis through an angle θ then a rotation about the new z axis through an angle ϕ. We represent each of these rotations individually by the

Cayley–Klein parameters in the angle-axis representation

$$R_{a\theta} = \cos\frac{\theta}{2},$$
$$R_{b\theta} = \sin\frac{\theta}{2},$$
$$R_{a\phi} = \cos\frac{\phi}{2} + i\sin\frac{\phi}{2},$$
$$R_{b\phi} = 0.$$

To make sense of how to compose rotations represented by the Cayley–Klein parameters, it is worthwhile to note that the Cayley–Klein parameters are the matrix elements of the SU(2) representation of rotation. So that the rotation (denoted by $\hat{\boldsymbol{R}}$) can be represented as

$$\hat{\boldsymbol{R}} = \begin{pmatrix} R_a & R_b \\ -R_b^* & R_a^* \end{pmatrix}.$$

Then, when composing rotations

$$\hat{\boldsymbol{R}} = \hat{\boldsymbol{R}}_2\hat{\boldsymbol{R}}_1,$$

where $\hat{\boldsymbol{R}}_1$, $\hat{\boldsymbol{R}}_2$ are SU(2) matrices that represent arbitrary rotations. Performing the matrix multiplication, we obtain the composition rule for rotations represented by the Cayley–Klein parameters,

$$\begin{aligned} R_a &= R_{a2}R_{a1} - R_{b2}R_{b1}^*, \\ R_b &= R_{a2}R_{b1} + R_{b2}R_{a1}^*. \end{aligned} \tag{21}$$

Then, for the case of spherical harmonics, the composition rule reduces to

$$\begin{aligned} R_a &= R_{a\phi}R_{a\theta} = \left(\cos\frac{\phi}{2} + i\sin\frac{\phi}{2}\right)\cos\frac{\theta}{2}, \\ R_b &= R_{a\phi}R_{b\theta} = \left(\cos\frac{\phi}{2} + i\sin\frac{\phi}{2}\right)\sin\frac{\theta}{2}. \end{aligned} \tag{22}$$

Finally, using the composition rule, we can then calculate the spherical harmonics using their relationship to the Wigner-D matrices,

$$Y_{lm}(R_a, R_b) = (-1)^m\sqrt{\frac{2l+1}{4\pi}}D^l_{0,-m}(R_a, R_b). \tag{23}$$

The radial inner product $\int_0^{r_{\text{cut}}} r^2 g_n(r)\, e^{-\alpha r^2} I_l(2\alpha r r_i) dr$ in Eq. (7) cannot be solved analytically, so we employ numerical integration for this purpose. The Chebyshev–Gauss quadrature is used so that the quadrature nodes for the interval $(0, r_{\text{cut}})$ never include $r = 0$ for any N number of nodes in the quadrature; the Chebyshev–Gauss quadrature nodes for the interval $(0, r_{\text{cut}})$ are given by

$$x_i = \frac{r_{\text{cut}}}{2}\left[\cos\left(\frac{2i-1}{2N}\pi\right) + 1\right]. \tag{24}$$

Avoiding the removable singularity at $r = 0$ due to I allows for the use of the following recursion relation to compute I at each of the nodes:

$$\begin{cases} I_0(x) = \frac{\sinh(x)}{x}, \\ I_1(x) = \frac{x\cosh(x) - \sinh(x)}{x^2}, \\ \vdots \\ I_n(x) = I_{n-2}(x) - \frac{2n-1}{x}I_{n-1}(x). \end{cases} \tag{25}$$

The gradient of the smooth SO(3) power spectrum components then follows:

$$\nabla_i p_{nn'l} = \sum_{m=-l}^{+l}\left(c^*_{n'lm}\nabla_i c_{nlm} + c_{nlm}\nabla_i c^*_{n'lm}\right), \tag{26}$$

where the gradient of the expansion coefficients is obtained through applying the product rule on Eq. (7) and then differentiating under the integral sign (as r_i is independent of r).

$$\begin{aligned} \nabla_i c_{nlm} = {} & 4\pi\nabla_i\left(e^{-\alpha r_i^2}\right)Y^*_{lm}(\hat{\mathbf{r}}_\mathbf{i})\int_0^{r_{\text{cut}}} r^2 g_n(r)\, e^{-\alpha r^2} I_l(2\alpha r r_i)\, dr \\ & + 4\pi\, e^{-\alpha r_i^2}\nabla_i\left(Y^*_{lm}(\hat{\mathbf{r}}_\mathbf{i})\right)\int_0^{r_{\text{cut}}} r^2 g_n(r)\, e^{-\alpha r^2} I_l(2\alpha r r_i)\, dr \\ & + 4\pi\, e^{-\alpha r_i^2}Y^*_{lm}(\hat{\mathbf{r}}_\mathbf{i})\nabla_i\left(\int_0^{r_{\text{cut}}} r^2 g_n(r)\, e^{-\alpha r^2} I_l(2\alpha r r_i)\, dr\right), \end{aligned} \tag{27}$$

$$\nabla_i\left(e^{-\alpha r_i^2}\right) = -2\alpha r_i\, e^{-\alpha r_i^2}\hat{\mathbf{r}}_\mathbf{i},$$

$$\nabla_i\left(\int_0^{r_{\text{cut}}} r^2 g_n(r)\, e^{-\alpha r^2} I_l(2\alpha r r_i)\, dr\right) = 2\alpha\int_0^{r_{\text{cut}}} r^3 g_n(r)\, e^{-\alpha r^2} I'_l(2\alpha r r_i)\, dr\hat{\mathbf{r}}_\mathbf{i}.$$

We again evaluate the radial integral using the Chebyshev–Gauss quadrature and use the following recursion relation for the evaluation of the first derivative of the modified spherical Bessel function:

$$I'_n(x) = \frac{1}{2n+1}\left[nI_{n-1}(x) + (n+1)I_{n+1}(x)\right].$$

Computing the gradient of the spherical harmonics is not as trivial as computing the gradient of the Wigner-D functions due to the singularities that exist at the north and south poles of the 2-sphere in Cartesian and spherical polar coordinates. Here, we remove those singularities by taking the gradient with respect to the covariant spherical coordinates. The covariant spherical coordinates are related to Cartesian coordinates by the following relations:[17]

$$x_{+1} = -\frac{1}{\sqrt{2}}(x + iy),$$
$$x_0 = z,$$
$$x_{-1} = \frac{1}{\sqrt{2}}(x - iy).$$

Then, the gradient of the spherical harmonics with respect to the covariant spherical coordinates is given by Ref. [17],

$$\begin{aligned}\nabla_0 Y_{l,m} = &-\frac{l}{r}\sqrt{\frac{(l+1)^2-m^2}{(2l+1)(2l+3)}} \times Y_{l+1,m}\\ &-\frac{l+1}{r}\sqrt{\frac{l^2-m^2}{(2l-1)(2l+1)}} \times Y_{l-1,m},\\ \nabla_{\pm 1} Y_{l,m} = &-\frac{l}{r}\sqrt{\frac{(l\pm m+1)(l\pm m+2)}{2(2l+1)(2l+3)}} \times Y_{l+1,m\pm 1}\\ &-\frac{l+1}{r}\sqrt{\frac{(l\mp m-1)(l\mp m)}{2(2l-1)(2l+1)}} \times Y_{l-1,m\pm 1}\end{aligned} \tag{28}$$

so that we can obtain the gradient with respect to Cartesian coordinates by transforming the basis vectors back to Cartesian unit vectors,[17]

$$\begin{aligned}\boldsymbol{e_x} &= \frac{1}{\sqrt{2}}(\boldsymbol{e_{-1}} - \boldsymbol{e_{+1}}),\\ \boldsymbol{e_y} &= \frac{i}{\sqrt{2}}(\boldsymbol{e_{-1}} + \boldsymbol{e_{+1}}),\\ \boldsymbol{e_z} &= \boldsymbol{e_0}.\end{aligned} \tag{29}$$

IV. INTERATOMIC POTENTIAL FITTING

In the present work, we adopted two fitting approaches: neural networks and linear regressions. Both techniques predict collection of atomic energies of a given structure: $E_{\text{total}} = \Sigma_i E_i$, where i loops through all atoms in the structure. Each $E_i = f(\boldsymbol{X}_i)$ is a function of descriptors, $\boldsymbol{X}_i$, representing the chemical environment around $\boldsymbol{r}_i$, a set of atomic positions relative to the i-th center atom within a cutoff radius (r_{cut}). Since the atom-centered descriptors are derived analytically as shown in Sec. III, one can deduce explicit forms of the functional to calculate forces and stress tensors.

A. Linear regression

Given the atom-centered descriptors ($\boldsymbol{X}_i$), the functional form for E_i can be expressed as a linear combination of the descriptors,

$$E_{\text{total}} = \sum_{i=1}^{N} E_i = \theta_0 + \boldsymbol{\theta} \cdot \sum_{i=1}^{N} \boldsymbol{X}_i, \tag{30}$$

where θ_0 and $\boldsymbol{\theta}$ denote as the weight parameters and N is the total atoms in the structure. The forces on each atom can be obtained by computing the partial derivative of $-\partial E/\partial \boldsymbol{r_i}$ through the chain rule. Optionally, one can also include information on virial stress in the training. In the context of linear regression, the objective is then to minimize the overall errors with respect to energy, forces, and stresses between the linear model and the training samples. To prevent overfitting, a penalty function, usually the l_1 or l_2 norm of $\boldsymbol{\theta}$ can be added to the expression of loss function to serve as a regularization term. Therefore, the final expression is

$$\Delta = \bar{E}_{\text{mse}} + \beta \bar{F}_{\text{mse}} + \gamma \bar{\sigma}_{\text{mse}} + \lambda \|\boldsymbol{W}\|_n, \tag{31}$$

where $\bar{E}_{\text{mse}}$, $\bar{F}_{\text{mse}}$, $\bar{\sigma}_{\text{mse}}$ denote the mean squared errors due to energy, force, and virial stress; $\|\boldsymbol{W}\|_n$ denotes the n-norm of the weight vector; and β, γ, λ denote the coefficients to balance the emphasis of training on force, stress, and penalty function. For the case of linear regression, $\boldsymbol{W}$ is the concatenated vector of $\{\theta_0, \boldsymbol{\theta}\}$.

B. Neural network regression

In a NN regression, the atom-centered descriptors serve as the inputs to the first layer of the neural network architecture. The NN architecture also consists of an output layer and hidden layers, where hidden layers reside in between input and output layers. Within a layer, there are a collection of units or nodes called neurons. The connectivity between these neurons in the layers mimics synapses of neurons in a biological structure. The signals or atom-centered descriptors permeate into the hidden layer to the output neuron in the following general form:

$$X^l_{n_i} = a^l_{n_i}\left(b^{l-1}_{n_i} + \sum_{n_j=1}^{N} W^{l-1,l}_{n_j,n_i} \cdot X^{l-1}_{n_j}\right). \tag{32}$$

The neuron $X^l_{n_i}$ at lth layer is established by the relationships between the weight parameter $W^{l-1,l}_{n_j,n_i}$, the bias parameter $b^{l-1}_{n_i}$, and the neurons in the prior layers $X^{l-1}_{n_j}$. Here, $W^{l-1,l}_{n_j,n_i}$ specifies the connectedness of the n_j neuron at $(l-1)$th layer to the neuron n_i at lth layer. Then, an activation function $a^l_{n_i}$ is applied to the process for the purpose of introducing non-linearity to the neurons. X_{n_i} at the output layer is equivalent to atomic energy in the scope of this study, in which the collection of these atomic energies is the total energy of the system. The details about the NN architecture and its application in interatomic potential fitting have been discussed in many excellent review works recently.[9,12,19]

V. RESULTS AND DISCUSSION

In this section, we will first compare the computational costs for each descriptor calculation as a function of the hyperparameters. The accuracy of each representation in relation to both the number of descriptors and its computational cost will be then investigated by regressing on energies, forces, and stresses of a representative binary alloy Ni_3Mo/Ni_4Mo system using linear regression. Last, we will introduce a more flexible NN regression model to improve the accuracy of fitting on the extended Ni–Mo dataset within a larger chemical space.

In parallel to force field fitting, generating a diverse training dataset is also a challenging task. Recently, there is an increasing trend for research groups to share their own data with the entire MLIAP community. Thanks to this trend, we choose to examine the dataset from a recent work by Li *et al.*,[20] which includes 4019 atomic configurations for elemental Ni, Mo, Ni_3Mo, Ni_4Mo, and doped Ni–Mo alloys. The training dataset consists of (1) undistorted ground state structures for Ni, Mo, Ni_3Mo, and Ni_4Mo, (2) distorted structures obtained by applying strains of -10% to 10% at 1% intervals to a bulk supercell, (3) surface structures of elemental structures, (4) snapshots from *ab initio* molecular dynamics simulations of the bulk supercell at several temperatures, and (5) doped alloy structures constructed by partial substitution of the bulk fcc Ni with Mo and the bulk bcc Mo with Ni. In addition, we also used the extra dataset on Mo from Ref. 21. For the computation of each descriptor below, we used a uniform cutoff distance of 4.9 Å.

A. Computational cost comparison

We begin with evaluating the computational cost of the SO(4) bispectrum components and the Smooth SO(3) power spectrum components, which requires some measure of the cost of each method. By far, the gradient is the most expensive part of the calculation, so we estimate the cost of each method by the accumulation of the gradient for one neighbor. The cost function for each method is evaluated by the asymptotic cost of accumulating the gradient plus the cost of precomputing the expansion coefficients and their gradients for a given truncation. For the SO(4) bispectrum components, the cost of precomputation is equivalent to the number of Wigner-D matrix elements to evaluate, $\sum_{j=0}^{2j_{\max}}(j+1)^2$, where for the smooth SO(3) power spectrum the cost of precomputation is equal to the number of Wigner-D matrix elements to evaluate, $(l_{\max}+2)^2$, added to the number of radial functions to evaluate for the quadrature [Eq. (34)], to compute each integral, we use $10(n+l+1)$ quadrature nodes. In our implementation, the cost of evaluating the radial functions is less than that of evaluating the D-functions, although for the sake of simplicity of the cost model we treat these costs as equal,

$$\text{cost} = \text{cost}_{\text{accum}} + \text{cost}_{\text{precomputation}}. \tag{33}$$

Therefore, we estimate the computational cost of each descriptor as follows:

$$\begin{aligned} &\text{SO(4)}: j_{\max}^5 + \sum_{j=0}^{2j_{\max}}(j+1)^2,\\ &\text{SO(3)}: n_{\max}^2 l_{\max}^2 + \left[(l_{\max}+2)^2 + \sum_{n=1}^{n_{\max}}\sum_{l=0}^{l_{\max}} 10(n+l+1)\right]. \end{aligned} \tag{34}$$

The cost of each descriptor is then compared with the number of elements of that descriptor. The number of unique elements of each descriptor are given by

$$\begin{aligned} N_{\text{SO(4)}} &= (j_{\max}+1)(j_{\max}+2)(j_{\max}+3/2)/3,\\ N_{\text{SO(3)}} &= n_{\max}(n_{\max}+1)(l_{\max}+1)/2. \end{aligned} \tag{35}$$

In Fig. 1, we plot the computational cost given by Eq. (34) with respect to the number of descriptors [Eq. (35)] for both the SO(4) bispectrum and SO(3) power spectrum. Clearly, we find that in the low band limit ($N \leq 30$), the SO(4) bispectrum components are much less costly than the smooth SO(3) power spectrum components, where at higher band limits, including more terms in the radial expansion of the smooth SO(3) power spectrum results in a less costly computation in comparison to the SO(4) bispectrum components.

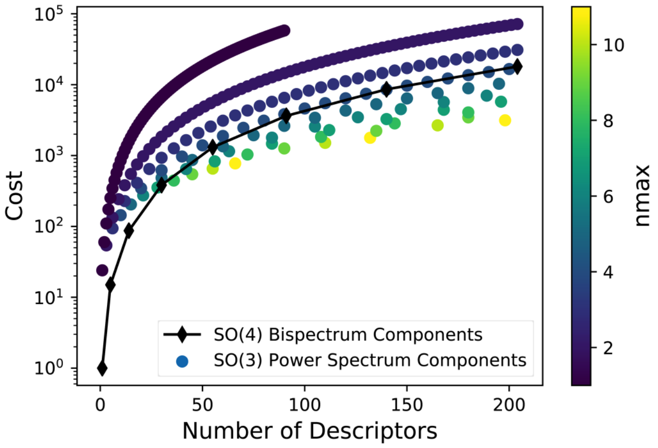


FIG. 1. The computational cost of the SO(4) bispectrum descriptor and the smooth SO(3) power spectrum descriptor vs the total number of elements of that descriptor. The smooth SO(3) power spectrum is also colored according to the number of radial components in the expansion.

B. Linear regressions on Ni$_4$Mo and Ni$_3$Mo

To evaluate the performance of these two descriptors, we first choose a subset of data from the Ni–Mo dataset, which includes 642 atomic configurations only in Ni$_3$Mo and Ni$_4$Mo stoichiometries. We then fit linear regressions to this data for each representation using a set of descriptors obtained through different hyperparameters in Eq. (34), while varying the coefficients of force's contribution to the total loss function.

The results of these regressions are shown in Fig. 2. Clearly, there is a general trend that both SO(3) power spectrum and SO(4) bispectrum can continuously achieve better accuracy with the inclusion of more components, although at high band limits that increased accuracy becomes marginal. In addition, the results show that high band limit fits vary less with respect to the change of force coefficient, indicating a convergence of the regression. However, a full convergence at high band limits results in incredibly expensive calculations. In real applications, it is generally advised to choose a smaller band limit. For the SO(4) bispectrum components, holding the truncation of $j_{\max} = 3$ is a rather common choice.[2,6,20] A more detailed analysis regarding the cost of computing the SO(4) bispectrum components with respect to $j_{\max}$ can be found in Ref. 7. Therefore, we aim to for a better solution through investigating the smooth SO(3) power spectrum.

Indeed, we find that the smooth SO(3) power spectrum components converge more quickly to lower errors in comparison to the SO(4) bispectrum components while also converging to a lower error overall. For instance, using only 90 smooth SO(3) power spectrum components yields similar accuracy (2.14 meV/atom in energy mean absolute error (MAE) and 0.06 eV/Åin force MAE) to 204 bispectrum components (1.68 meV/atom in energy MAE

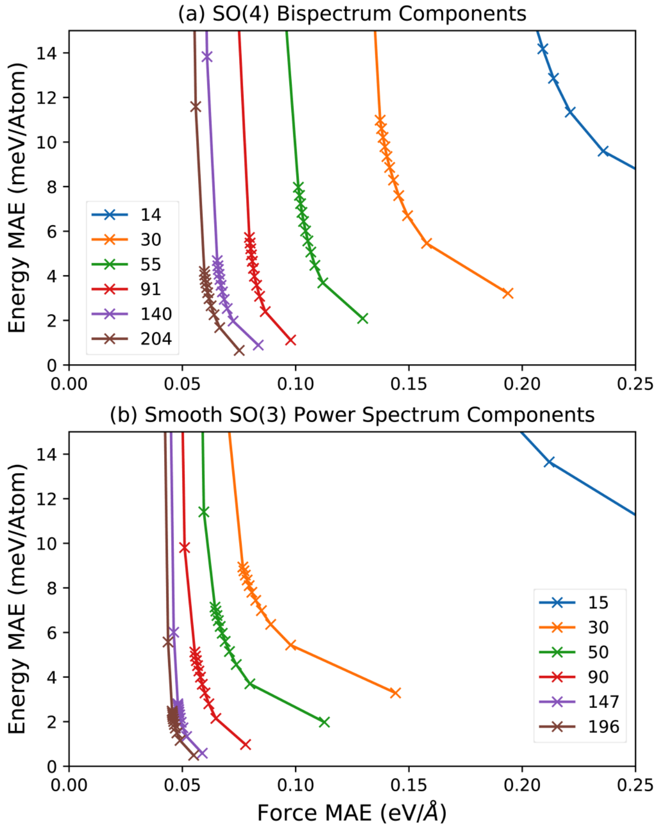


FIG. 2. Linear regressions of both the SO(4) and SO(3) representations with varying numbers of components. The force coefficients used fall between 1 × 10^{-6} and 1 × 10^{0} with most points falling between 1 × 10^{-5} and 1 × 10^{-4}.

and 0.07 eV/Å in force MAE), if we hold the force coefficient at 1×10^{-5}. To further illustrate the performance of both descriptors in terms of computational cost, we calculate both the 90 component smooth SO(3) power spectrum and 204 component SO(4) bispectrum for the ground state Ni_3Mo structure with eight atoms in the unit cell at a cutoff radius of 4.9 Å with the gradient; the smooth SO(3) power spectrum component calculation is completed 0.56 s, whereas the SO(4) bispectrum component calculation is completed in 3.97 s. Since our code is written in Python (using the LLVM compiler through Numba[22]), we expect the run time will be less if the code is rewritten in C++ or Fortran. These results suggest that the smooth SO(3) power spectrum is a more efficient descriptor in terms of both accuracy and computational cost in the context of linear regression.

Although both descriptors yield satisfactory accuracy on the Ni_3Mo/Ni_4Mo dataset, we found it hard to maintain the same level of accuracy when extending the training dataset with other stoichiometries (e.g., elemental Ni/Mo) for the regression. In principle, one can improve the regression by tuning force and stress coefficients, applying regularization, and adopting a nonuniform weight scheme on each sample.[7,20] However, a more automated approach to dealing with large data is to employ a more flexible regression model such as NN regression to be presented in Subsection V C.

C. Neural network regressions on Ni–Mo alloys

When dealing with a large amount of data, linear regression requires very fine-tuning of hyperparameters to achieve acceptable accuracies. To achieve these accuracies, optimization schemes are adopted to adjust hyperparameters such as descriptor size, specie weights, cutoff radii, and nonuniform data weighting so that obtaining an optimal fit requires many training cycles.[7,20,21] NN regression provides a more automated approach to achieve greater accuracy on larger datasets without the need for high band limit descriptors or heavy hyperparameter optimization. In this study, we seek to use a small set of descriptors (30) to train a MLIAP on the entire Ni–Mo dataset consisting of over 4000 structures to satisfactory accuracy through a simple feed forward neural network consisting of two hidden layers of 16 neurons each. For a fair comparison, we prepare two sets of descriptors (1) the bispectrum components with $j_{\max} = 3$ and (2) the smooth SO(3) power spectrum components with $l_{\max} = 4$ and $n_{\max} = 3$. To ensure that the results can describe elastic deformation well, we also consider the virial stresses for the elastic configurations in the training. Correspondingly, we set $\beta = 3 \times 10^{-3}$, $\gamma = 1 \times 10^{-4}$, and $\lambda = 1 \times 10^{-8}$ for the evaluation of the loss functions [Eq. (31)] in all subsequent NN runs.

Table I lists the training results in terms of energy and force for all three models. In the previously reported linear SNAP model,[20] the overall fitting results are 22.5 meV/atom in energy

TABLE I. Comparison of the spectral neural networks models' MAE values from different descriptors. For reference, the previous NiMo model trained from SNAP[20] is also included. Note that in the SNAP model,[20] only 247 Mo structures were used for training. In our work, we replaced the elastic configuration data with the dataset from Ref. 21. In the parenthesis, it gives the number of configurations for each group.

Properties	Descriptor	$j_{\max}$	$l_{\max}$	$n_{\max}$	Architecture	Mo (377)	Ni (414)	Mo_{Ni} (918)	Ni_{Mo} (1668)	Ni_3Mo (321)	Ni_4Mo (321)	Overall (4019)
Energy (meV/atom)	SO(4)[20]	3			Linear Reg.	16.2	7.9	22.7	33.9	5.2	4.0	22.5
	SO(4)	3			30-16-16-1	6.2	7.3	5.6	6.1	6.1	6.4	6.1
	SO(3)		4	3	30-16-16-1	6.3	3.6	6.2	6.7	4.9	4.6	5.9
Force (eV/Å)	SO(4)[20]	3			Linear Reg.	0.29	0.11	0.13	0.55	0.16	0.14	0.23
	SO(4)	3			30-16-16-1	0.19	0.07	0.06	0.10	0.10	0.09	0.10
	SO(3)		4	3	30-16-16-1	0.18	0.04	0.06	0.10	0.09	0.07	0.08

MAE and 0.23 eV/Å. Clearly, both NN models are able to yield significantly better results (6 meV/atom for energy and 0.08 eV/Å for force) than the previous reported linear model. Notably, the linear regression also reports drastically lower accuracy in both energy and force for the Mo_{Ni}/Mo_{Ni} sets, suggesting that the elemental Ni/Mo and Ni_3Mo/Ni_4Mo portions of the data were weighted much higher in the regression. In particular, the 1668 Ni_{Mo} set, occupying the largest percentage of the data, has a energy MAE of 33.9 meV/atom and force MAE of 0.55 eV/Å. As such, the predictability of linear SNAP model is likely to be limited in describing the configurations in the vicinity of the Mo_{Ni}/Mo_{Ni} alloys. In contrast, the neural network regressions do not need a special weighting scheme. The models from both the SO(4) bispectrum and SO(3) power spectrum yield not only lower energy and force errors for the overall fitting. The energy/force errors for each group are also more evenly distributed.

TABLE II. Comparison of elastic properties predicted from several different Models. *B* and *G* denote the empirical Voigt–Reuss–Hill average of bulk and shear moduli, respectively. *ν* is the Poisson's ratio.

		SNAP[20]	SO(4)	SO(3)
σ(MAE) (GPa)	DFT	N/A	0.295	0.289
Mo				
c_{11} (GPa)	472	475	487	479
c_{12} (GPa)	158	163	153	168
c_{44} (GPa)	106	111	108	82
B (GPa)	263	267	265	271
G (GPa)	124	127	129	106
ν	0.30	0.29	0.29	0.33
Ni				
c_{11} (GPa)	276	269	275	271
c_{12} (GPa)	159	150	162	150
c_{44} (GPa)	132	135	137	120
B (GPa)	198	190	199	188
G (GPa)	95	97	96	88
ν	0.29	0.28	0.29	0.30
Ni_3Mo				
c_{11} (GPa)	385	420	426	402
c_{22} (GPa)	402	360	354	382
c_{33} (GPa)	402	408	379	394
c_{12} (GPa)	166	197	159	159
c_{13} (GPa)	145	162	133	109
c_{23} (GPa)	131	145	208	173
c_{44} (GPa)	58	N/A	54	70
c_{55} (GPa)	66	N/A	68	52
c_{66} (GPa)	94	84	79	58
B (GPa)	230	243	240	229
G (GPa)	89	100	80	80
ν	0.33	0.32	0.35	0.34
Ni_4Mo				
c_{11} (GPa)	313	326	319	343
c_{33} (GPa)	300	283	294	293
c_{12} (GPa)	166	179	166	160
c_{13} (GPa)	186	164	199	193
c_{44} (GPa)	130	126	136	131
c_{66} (GPa)	106	N/A	102	113
B (GPa)	223	220	221	222
G (GPa)	91	95	96	102
ν	0.33	0.31	0.31	0.30

The elastic tensor is another important metric to check if the trained MLIAPs are able to reproduce the fine details of the PES on the representative basins. To ensure a satisfactory fitting to the elastic properties, we also included training on the stress tensors for the elastic configurations from the previous works.[20,21] Table II shows the predicted elastic properties from each model for the ground state structures of BCC Mo, FCC Ni, Ni_3Mo, and Ni_4Mo. In agreement with the previously reported linear SNAP model,[20] the elastic data predicted by each MLIAP agree with the reported DFT result within similar levels of accuracy across all four ground state structures. In the previous work, it is likely that the authors adjusted the weight for each group of structures in order to achieve a better fit in the elastic properties at the expense of accuracy in energy and force. However, these NN regressions can circumvent this trade-off by using a more flexible expression in describing the target properties (energy, force, stress tensor) in fitting. As such, the NN models can yield greater accuracy with respect to energy and force while maintaining accuracy in elastic properties all without the need for heavy hyperparameter optimization.

Last, it is also of interest to compare the performance of fitting between the SO(4) bispectrum and smooth SO(3) power spectrum models. In Sec. V B, it is clear that SO(3) is superior to SO(4) in the context of linear regression. However, this is no longer the case for NN regression. With the same number of descriptors (30), both NN models yield very similar levels of accuracy. In terms of elastic properties prediction, the SO(4) model seems to be slightly better than SO(3) though SO(3) generated a slightly lower MAE value for stress tensors overall.[23] From the point view of computational cost, computing the 30 bispectrum components is less expensive than computing the same number of power spectrum components. Therefore, it is fair to conclude that two descriptors are competitive for the application of NN regression.

VI. CONCLUSION

In summary, we present a numerical implementation of computing the atom-centered descriptors derived from harmonic analysis, which include the SO(4) bispectrum components and the smooth SO(3) power spectrum. Using these descriptors to fit machine learning interatomic potentials for a small set of Ni–Mo stoichiometries within a narrow chemical composition space, we found that both descriptors are able to yield satisfactory accuracy within the framework of linear regression. However, the linear regression is not easily extended to fit a more diverse dataset from a larger chemical composition space and even then accuracy can still be lacking without hyperparameter optimization such as descriptor size, specie weights, cutoff radii, and nonuniform data weighting. Hence, we demonstrate that neural networks regression paired with the SO(4) bispectrum components or the smooth SO(3) power spectrum components can provide a better trained model without the need for large band limit descriptors or heavy hyperparameter optimization. The validity of the trained models are further supported by the accuracy of elastic property calculations. Last, the SO(3) power spectrum descriptor clearly exhibits better agreement with the total energy than the SO(4) bispectrum components, thus it is a better choice for linear regression. However, when adopted to the neural networks regression, both descriptors tend to yield the same level of accuracy. A further comparison of the performances of different types of descriptors will be the subject of future studies.

ACKNOWLEDGMENTS

We acknowledge the NSF (I-DIRSE-IL: No. 1940272) and NASA (No. 80NSSC19M0152) for financial support. The computing resources are provided by XSEDE (No. TG-DMR180040). The authors thank Dr. A. Thompson (Sandia), Dr. S. Ong (UCSD), and Dr. Y-G Li (USCD) for insightful discussions in the computation of bispectrum coefficients.

APPENDIX: ALTERNATIVE EXPRESSION TO COMPUTE *D*

We are aware that two previous works[3,6] used a different approach to compute the Wigner-*D* martices.[3,6] To start, a different set of Cayley–Klein parameters were used,

$$\begin{aligned} R_a &= \frac{1}{\sqrt{r^2 + r^2 \cot^2(\omega/2)}}(r\cot(\omega/2) + iz), \\ R_b &= \frac{1}{\sqrt{r^2 + r^2 \cot^2(\omega/2)}}(y + ix), \end{aligned} \tag{A1}$$

which can be shown to be identical to Eq. (15). However, when implemented numerically, there exists a singularity at $\omega = 0$ and $\omega = 2\pi$, so we choose to implement Eq. (15) rather than treating $\omega = 0$ as a separate case and omitting $\omega = \pi$ altogether. Moreover, they used a recursive scheme to compute the *D* matrices,

$$\begin{cases} D^j_{mm'} = \sqrt{\frac{j-m}{j-m'}} R^*_a D^{j-1/2}_{m+1/2,m'+1/2} - \sqrt{\frac{j+m}{j-m'}} R^*_b D^{j-1/2}_{m-1/2,m'+1/2}, & m' \neq j, \\ D^j_{mm'} = \sqrt{\frac{j-m}{j+m'}} R_b D^{j-1/2}_{m+1/2,m'-1/2} + \sqrt{\frac{j+m}{j+m'}} R_a D^{j-1/2}_{m-1/2,m'-1/2}, & m' \neq -j. \end{cases} \tag{A2}$$

Compared to the polynomial form Eq. (16), the recursive form requires less floating point operations, in general, and is more efficient in serial calculations. However, in parallel architectures, a polynomial form of the *D*-matrices is advantageous as no single term depends on another. During our implementation, we found that using Numba's automatic parallelization,[22] we were able to fuse all loops in the *D*-matrix calculation to achieve parallelization more so than algorithm when compared to the recursive version. This difference results in an improved scaling of the algorithm.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. The source code to analyze the data is available on https://github.com/qzhu2017/PyXtal_FF.

REFERENCES

[1] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," Phys. Rev. **140**, A1133–A1138 (1965).

[2] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood *et al.*, "Performance and cost assessment of machine learning interatomic potentials," J. Phys. Chem. A **124**, 731–745 (2020).

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

[4] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).

[5] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," Multiscale Model. Simul. **14**, 1153–1173 (2016).

[6] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, "Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials," J. Comput. Phys. **285**, 316–330 (2015).

[7] M. A. Wood and A. P. Thompson, "Extending the accuracy of the snap interatomic potential form," J. Chem. Phys. **148**, 241721 (2018).

[8] A. P. Bartók and G. Csányi, "Gaussian approximation potentials: A brief tutorial introduction," Int. J. Quantum Chem. **115**, 1051–1057 (2015).

[9] J. Behler, "Constructing high-dimensional neural network potentials: A tutorial review," Int. J. Quantum Chem. **115**, 1032–1050 (2015).

[10] R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials," Phys. Rev. B **99**, 014104 (2019).

[11] M. J. Willatt, F. Musil, and M. Ceriotti, "Atom-density representations for machine learning," J. Chem. Phys. **150**, 154110 (2019).

[12] J. Behler, "Perspective: Machine learning potentials for atomistic simulations," J. Chem. Phys. **145**, 170901 (2016).

[13] M. Ceriotti, M. J. Willatt, and G. Csányi, "Machine learning of atomic-scale properties based on physical principles," in *Handbook of Materials Modeling: Methods, Theory and Modeling*, edited by W. Andreoni and S. Yip (Springer International Publishing, Cham, 2018), pp. 1–27.

[14] H. Yanxon, D. Zagaceta, B. C. Wood, and Q. Zhu, "On transferability of machine learning force fields: A case study on silicon," preprint arXiv:2001.00972 (2020).

[15] R. Kondor, "A complete set of rotationally and translationally invariant features for images," CoRR **abs/cs/0701127** (2007), see arXiv:cs/0701127.

[16] P. J. Steinhardt, D. R. Nelson, and M. Ronchetti, "Bond-orientational order in liquids and glasses," Phys. Rev. B **28**, 784–805 (1983).

[17] D. A. Varshalovich, A. N. Moskalev, and V. K. Khersonskii, *Quantum Theory of Angular Momentum* (World Scientific, Singapore, 1988).

[18] M. Boyle, "Angular velocity of gravitational radiation from precessing binaries and the corotating frame," Phys. Rev. D **87**, 104006 (2013).

[19] N. Artrith and A. Urban, "An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO_2," Comput. Mater. Sci. **114**, 135–150 (2016).

[20] X.-G. Li, C. Hu, C. Chen, Z. Deng, J. Luo, and S. P. Ong, "Quantum-accurate spectral neighbor analysis potential models for Ni-Mo binary alloys and FCC metals," Phys. Rev. B **98**, 094104 (2018).

[21] C. Chen, Z. Deng, R. Tran, H. Tang, I.-H. Chu, and S. P. Ong, "Accurate force field for molybdenum by machine learning large materials data," Phys. Rev. Mater. **1**, 43603 (2017).

[22] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based python JIT compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM'15 (Association for Computing Machinery, New York, 2015).

[23] We note that each NN training follows a stochastic optimization process. So each time, it may generate slightly different results. Hence, the comparison is not definitive.