The Journal of Chemical Physics

Neural network potential from bispectrum components: A case study on crystalline silicon

Cite as: J. Chem. Phys. **153**, 054118 (2020); https://doi.org/10.1063/5.0014677
Submitted: 22 May 2020 . Accepted: 16 July 2020 . Published Online: 06 August 2020

Howard Yanxon, David Zagaceta, Brandon C. Wood, and Qiang Zhu

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Neural network potential from bispectrum components: A case study on crystalline silicon

Cite as: J. Chem. Phys. 153, 054118 (2020); doi: 10.1063/5.0014677
Submitted: 22 May 2020 • Accepted: 16 July 2020 •
Published Online: 6 August 2020

Howard Yanxon,[1,2] David Zagaceta,[1] Brandon C. Wood,[2] and Qiang Zhu[1,a)]

AFFILIATIONS

[1] Department of Physics and Astronomy, University of Nevada, Las Vegas, Nevada 89154, USA
[2] Materials Science Division, Lawrence Livermore National Laboratory, Livermore, California 94550, USA

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

ABSTRACT

In this article, we present a systematic study on developing machine learning force fields (MLFFs) for crystalline silicon. While the mainstream approach of fitting a MLFF is to use a small and localized training set from molecular dynamics simulations, it is unlikely to cover the global features of the potential energy surface. To remedy this issue, we used randomly generated symmetrical crystal structures to train a more general Si-MLFF. Furthermore, we performed substantial benchmarks among different choices of material descriptors and regression techniques on two different sets of silicon data. Our results show that neural network potential fitting with bispectrum coefficients as descriptors is a feasible method for obtaining accurate and transferable MLFFs.

I. INTRODUCTION

Atomistic modeling methods such as molecular dynamics (MD) or Monte Carlo (MC) play important roles in investigating time-dependent physical and chemical processes. In these methods, energy and forces need to be recalculated iteratively as the atomic configuration evolves. Consequently, atomistic simulations crucially depend on the accuracy of the underlying potential energy surface (PES). Modern quantum mechanical modeling based on density functional theory (DFT) can consistently generate accurate energetic descriptions for many solid systems.[1] However, MD simulations based on DFT suffer from the highly demanding computational cost. The simulations are only suitable to model a system with up to a few thousands of atoms at tens of picoseconds. On the other hand, the classical force field (FF) method is widely employed to simulate materials with millions of atoms at hundreds of nanoseconds. This method has enabled many explorations that lead to revealing interesting physical and chemical phenomena.[2–4] However, the construction of a reliable PES by the classical FF method remains problematic. In developing classical FF, a set of parameters are fitted to a few DFT and/or experimental data to compute the potential energy of a system, given an analytic functional form. Due to the constraints on the functional form and the limitation of the training dataset, the accuracy of classical FF is not dependable.

Meanwhile, *in silico* materials discovery requires an accurate yet efficient energy model to screen materials' properties in a high-throughput manner. In the past decade, discoveries of new materials have been highly driven by advanced structure prediction methods such as crystal structure prediction (CSP)[5] and data mining.[6] In both cases, the DFT method is used to perform geometry relaxation and energy evaluation. Despite the power of the current supercomputer, the computational cost for DFT simulation remains a bottleneck to many important and fascinating puzzles in materials science. Ideally, an approach that preserves DFT accuracy without sacrificing the computational cost is desirable.

To resolve the limitations described above, many efforts have been devoted toward establishing the machine learning force field (MLFF) method. Compared to the DFT method, the MLFF approach demands far lower computational cost (2–4 orders of magnitude lower) while retaining accuracy at the DFT level. The power of the MLFF method is illustrated by many applications to a range of materials.[7–10] A large amount of DFT data (structures, energy, forces, and stresses) are required to develop an accurate MLFF. The structures must be represented by appropriate descriptors (high-dimensional real valued array) in order to identify the similarities and/or dissimilarities in the atomic environments. In MLFF fitting, a variety of regression techniques are used to correlate between the descriptor and energy/forces. Several machine

learning techniques for developing MLFF had been successfully implemented: linear/polynomial regression,[11–14] Gaussian process regression,[15,16] and high-dimensional neural network potential (NNP).[17,18] A benchmark study of these machine learning methods had been carried out for performance and cost inspections to many elemental systems.[19] Nevertheless, many of the published MLFFs lack transferability/versatility, which is crucial in crystal structure prediction.

In the past few years, many researchers have attempted to improve transferability for many different systems.[10,20–25] Two approaches, including advanced sampling and structure prediction, have recently become popular. One is to force ordinary MD simulations to escape from the already explored equilibrium states,[26,27] while the other attempts to identify the low energy configurations by sampling many different basins mostly based on geometry optimizations. A heterogeneous training dataset—diversity in structural types—enhances transferability across different types of structures, curing the extrapolation problem.[24,25] Zeni *et al.*[24] achieved a good trade-off between transferability and overall accuracy by applying Gaussian process regression with a diverse dataset (including high temperature structures). Similarly, many physical properties were reproduced within 10% relative error to the DFT.[25] Hajinazar *et al.* employed a structure prediction technique to generate more diverse datasets than the common, less diverse, dataset generated with the MD-based approach.[20] In addition, it was proposed that the generation of MLFFs could be performed in conjunction with structure prediction processes. The active learning approach in constructing MLFFs on-the-fly was employed automatically to deal with extrapolation outside the training domain. Then, the MLFFs replaced the DFT gradually for structural relaxations and energy evaluations with much lower computational cost. The active learning technique had been successfully applied to predict PES reconstructions of several challenging elemental systems[21,22] and multi-component systems.[23] For instance, Deringer *et al.*[21] used Gaussian process regression combined with random structure searching (RSS) algorithms to systematically construct an interatomic potential for boron; Podryabinkin *et al.*[22] employed the evolutionary algorithm USPEX to build the machine-learning interatomic potentials for several elemental allotropes; and similar ideas were also applied to investigate the surface reconstructions[23] and nanoparticles.[28]

In this report, we will discuss about our attempts on developing accurate and transferable MLFFs for elemental silicon as the prototypical system. Many silicon MLFFs had been developed using the training datasets obtained by running MD simulations and selecting known structural prototypes manually.[10,15,19,25,29–32] These configurations from MD trajectories tend to possess strong correlations with the initial geometry. Hence, the resulting MLFFs can only describe a few energy basins of the entire PES. We believe that there are two main factors that can influence the transferability of the MLFF. First, the training dataset generated with the high-throughput structure prediction method can enhance the transferability. Here, we generate a diverse silicon dataset by using our in-house code, PyXtal[33]—a Python package for random crystal structure generation. The DFT-quality dataset spans a large space in the PES covering many energy basins, and the DFT setting is provided in Sec. II A. Second, we enable a machine learning infrastructure that allows Behler–Parrinello descriptors and bispectrum coefficient descriptors to be trained with generalized linear regression and neural networks. The details of the descriptors and the regression techniques are available in Secs. II B and II C, respectively. Finally, we will systematically construct the NNP with bispectrum coefficients as the descriptors in Sec. III.

II. COMPUTATIONAL METHODOLOGIES

A. *Ab initio* calculation

Ab initio calculations are necessary to provide the training dataset for MLFF development. In this study, we employed PyXtal[33] software to generate several thousands of structural configurations. For each configuration, the total energy and forces were calculated at the DFT level through the ASE package.[34] ASE provides interface to the Vienna Ab Initio Simulation Package (VASP) code[35] within projector augmented wave methodology[36] to perform geometry relaxations. In our calculation, we used the Perdew-Burke-Ernzerhof generalized gradient approximation (PBE-GGA)[37] as the exchange-correlation functional with an energy cutoff of 600 eV and a Γ-centered KSPACING of 0.15.

B. Descriptors

Descriptors, as the unique numerical representations of atomic structures, play an essential part in constructing MLFFs. It is crucial for a descriptor to be able to distinguish the local environments of atomic structures. The most common choice of representation by atomic coordinates is convenient, but it poorly describes the structural environments. The Cartesian coordinates of a crystal structure can change through translational or rotational operation, while the energy remains invariant. Thus, physically meaningful descriptors must be unaffected by these alterations to the structural environment, and any permutation of atoms should not change the descriptors. Additionally, the descriptors must be continuously differentiable within the domain of the local atomic environment. In the last decade, the atom-centered descriptors, which probe the atomic environment by their neighboring vectors, became popular because they fit the criteria. The descriptors usually operate within a cutoff function to ensure that the descriptors smoothly vanish to zero at a given cutoff radius, R_c. A popular cutoff function choice is the so-called cosine cutoff function. The function is expressed in the following:

$$f_c(R_{ij};R_c)=\begin{cases}\frac{1}{2}\left[\cos\left(\frac{\pi R_{ij}}{R_c}\right)+1\right] & R_{ij}\le R_c\\ 0 & \text{otherwise}\end{cases}, \quad (1)$$

where R_{ij} is the distance between the center atom i and the neighbor atom j.

Among the atom-centered descriptors, Behler–Parrinello descriptors[17] and bispectrum coefficients[15] are widely used in the materials modeling community. Their definitions will be discussed briefly as follows.

1. Behler–Parrinello descriptors

Behler–Parrinello descriptors are used regularly to represent the local atomic environments of crystal structures in NNP development. Commonly used Behler–Parrinello descriptors are two-body (G^2) and three-body (G^4) symmetry functions,

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

$$G_i^4 = 2^{1-\zeta} \sum_{j\neq i} \sum_{k\neq i,j} (1+\lambda\cos\theta_{ijk})^{\zeta} \cdot \times e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} \cdot f_c(R_{ij}) \cdot f_c(R_{ik}) \cdot f_c(R_{jk}). \tag{3}$$

G^2 is mainly designed to capture the radial environment, while G^4 is used for describing the angular part by including the three-body *ijk* terms. R_s shifts the center of the Gaussian functions to a certain radius, resulting in a spherical shell with the Gaussian width of η. ζ controls the angular resolution, and λ usually takes the value of +1 and −1 for inverting the cosine function. The cutoff function (f_c) is consistent with Eq. (1). There is a set of G_i^2 and G_i^4 descriptors specifying the center atom *i* in relation to the neighboring atoms *j* in terms of radial and angular parts. For a real material system, this set of parameters need to be optimized by a more extensive search.[38–41]

2. Bispectrum coefficients

Similar to Behler–Parrinello descriptors, the SO(4) bispectrum can be used to represent the local atomic environments. It was first introduced by Bartók *et al.* for the training of machine learning FF (MLFF) on the elemental systems of Group IVA.[15] A detailed study of the SO(4) bispectrum as a descriptor along with several alternative implementations [SO(3) bispectrum, angular Fourier series, and SOAP kernel] is available in Ref. 29. Later, Thompson *et al.* proposed the spectral neighbor analysis (SNAP) method and demonstrated that the SO(4) bispectrum could achieve satisfactory accuracy based on the simple linear[11] and quadratic regressions.[12] Following the original work, the expression of the SO(4) bispectrum is formed by the expansion coefficients of 4D hyperspherical harmonics,

$$B_i^{l_1,l_2,l} = \sum_{m,m'=-l}^{l} (c^l_{m',m})^* \sum_{m_1,m_1'=-l_1}^{l_1} \sum_{m_2,m_2'=-l_2}^{l_2} \times c^{l_1}_{m_1',m_1} c^{l_2}_{m_2',m_2} H^{l,m,m'}_{l_1,m_1,m_1',l_2,m_2,m_2'}, \tag{4}$$

where $H^{l_1,l_2,l}_{m_1',m_2',m',m_1,m_2,m}$ is analog to the Clebsch–Gordan coefficients on a 3-sphere. In application, it is the product of two ordinary Clebsch–Gordan coefficients on a 2-sphere. $c^{l,m}_{l_1,m_1,l_2,m_2}$ are the expansion coefficients from the hyperspherical harmonic ($U^l_{m',m}$) functions that are projected from the atomic neighborhood density within a cutoff radius onto the surface of a four-dimensional sphere,

$$\rho = \sum_{l=0}^{+\infty} \sum_{m=-l}^{+l} \sum_{m'=-l}^{+l} c^l_{m',m} U^l_{m',m}, \tag{5}$$

where the expansion coefficients are defined as

$$c^l_{m',m} = \left\langle U^l_{m',m} | \rho \right\rangle. \tag{6}$$

In this work, our implementation of the SO(4) bispectrum or bispectrum descriptor is very similar to the SNAP method[11] that is implemented in the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) code.[42] However, we introduce another method to calculate the hyperspherical harmonics and their gradients.[43] The benefit of this method is that it allows for the removal of singularities at the north and south poles of the 3-sphere that exist in the traditional implementation. Furthermore, we also include an option to normalize the expansion coefficients from the hyperspherical harmonics, where the normalization factor is $\frac{\sqrt{2l+1}}{4\pi}$. The impacts of normalization on the MLFF training will be discussed later in Sec. III C.

C. Machine learning force field fitting

The construction of the total energy (E_{total}) of a structure can be obtained by the summation of atomic energy (E_i) evaluated from atom-centered descriptors, $\boldsymbol{X}_i$,

$$E_{\text{total}} = \sum_{i}^{\text{all atoms}} E_i(\boldsymbol{X}_i). \tag{7}$$

The atomic energy contributions depend on the local structural environment within a cutoff radius with respect to the center atom *i*. Furthermore, an accurate representation of PES is also dependent on the contributions of forces. The force that acted on atom *j* can be expressed by the negative gradient of the energy with respect to its atomic positions ($\boldsymbol{r}_j$),

$$\boldsymbol{F}_j = -\sum_{i}^{\text{all atoms}} \frac{\partial E_i(\boldsymbol{X}_i)}{\partial \boldsymbol{X}_i} \cdot \frac{\partial \boldsymbol{X}_i}{\partial \boldsymbol{r}_j}. \tag{8}$$

The functional forms of *E* and *F* are fully dependent on the regression algorithm. Generalized linear regression and neural network (NN) regression will be discussed in Secs. II C 1 and II C 2.

1. Generalized linear regression

Linear regression is the most fundamental approach in curve fitting. In this context, each atomic energy is assumed to be linearly correlated with the descriptors. Thus, the total energy can be expressed as follows:

$$E_{\text{total}} = \gamma_0 + \boldsymbol{\gamma} \cdot \sum_{i=1}^{N} \boldsymbol{X}_i, \tag{9}$$

where γ_0 and $\boldsymbol{\gamma}$ are the weights presented in scalar and vector forms, and *N* is the total number of atoms in a structure.

In general, the total energy can be described as a generalized linear regression with extended polynomial terms. The following equation is a version to the second-order (quadratic) expansion in the Taylor series:

$$E_{\text{total}} = \gamma_0 + \boldsymbol{\gamma} \cdot \sum_{i=1}^{N} \boldsymbol{X}_i + \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{X}_i^T \cdot \boldsymbol{\Gamma} \cdot \boldsymbol{X}_i, \tag{10}$$

where $\boldsymbol{A}$ is the symmetric weight matrix (i.e., $\boldsymbol{A}_{12} = \boldsymbol{A}_{21}$) describing the quadratic terms. From linear to quadratic regression, the size of weight coefficients increases from $N+1$ to $(N+1)(N+2)/2$. Indeed, the energy can be further expanded to higher order. However, we restrict it to the second-order expansion due to the drastic increase in the size of weight coefficients.

Correspondingly, the force of an atom *j* can be expressed in this form by expanding the terms in Eq. (8) with Eq. (10),

$$\boldsymbol{F}_j = \sum_{i=1}^{N} \left(-\boldsymbol{\gamma} \cdot \frac{\partial \boldsymbol{X}_i}{\partial \boldsymbol{r}_j} - \frac{1}{2} \left[\frac{\partial \boldsymbol{X}_i^T}{\partial \boldsymbol{r}_j} \cdot \boldsymbol{\Gamma} \cdot \boldsymbol{X}_i + \boldsymbol{X}_i^T \cdot \boldsymbol{\Gamma} \cdot \frac{\partial \boldsymbol{X}_i}{\partial \boldsymbol{r}_j} \right] \right). \tag{11}$$

Both energy and force terms have a linear correlation with the expanded descriptors through a set of weight coefficients $\{\gamma_0, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_N, \boldsymbol{\Gamma}_{11}, \boldsymbol{\Gamma}_{12}, \ldots, \boldsymbol{\Gamma}_{NN}\}$. For convenience, we call the set of coefficients as $\boldsymbol{w}$ from now on. To obtain the best $\boldsymbol{w}$, we solve the objective cost function following the least squares formula for both energy and force,

$$\Delta = \frac{1}{2s}\sum_{i=1}^{s}\left[\left(\frac{E_i - E_i^{\mathrm{Ref}}}{N_{\mathrm{atom}}^i}\right)^2 + \frac{\beta}{3N_{\mathrm{atom}}^i}\sum_{j=1}^{3N_{\mathrm{atom}}^i}(F_{i,j} - F_{i,j}^{\mathrm{Ref}})^2\right], \quad (12)$$

where s is the total number of structures, i loops over all structures, and j loops over all atoms for each structure i in all three directions. N_i^{atom} is the total number of atoms in the ith structure. β is the force coefficient. It balances the energy and force contributions due to the number of force components being much larger. The cost function compares the predicted values obtained from the regression (E_i and $F_{i,j}$) to the true values of E^{Ref} and $F_{i,j}^{\mathrm{Ref}}$.

To prevent overfitting, it is useful to add a penalty term to account for the complexity of the entire weights (m) to Eq. (12),

$$\Delta_{\mathrm{p}} = \frac{\alpha}{2s}\sum_{i=1}^{m}(\boldsymbol{w}^i)^2, \quad (13)$$

where α is a dimensionless number that controls the degree of penalty. Adding such a penalty function in the context of machine learning is called regularization. Then, the optimum solution can be solved by finding the $\boldsymbol{w}$ leading to the zero partial derivative of Δ with respect to each element in $\boldsymbol{w}$. Accordingly, we use the *numpy.linalg.lstsq*[44] solver for generalized linear regression problems.

2. Neural network regression

In this section, the high-dimensional NN (Fig. 1) is introduced. The regression based on NN can be considered as an extension of the linear regression model. For a crystal structure that consists of N atoms, there are N positions ($\boldsymbol{R}_{\mathrm{N}}$) for the atoms to arrange themselves. N atom-centered descriptors ($\boldsymbol{X}_i$) for the structure can be mapped based on this atomic configuration. Each of the atom-centered descriptors is, then, fed into a NN architecture [Fig. 1(b)]. The NN architecture consists of input, hidden, and output neurons. These neurons are organized in layers as shown. The neurons in the first layer (input layer) are occupied by the atom-centered descriptors. The neuron at the output layer defines the atomic energy, E_i. Hidden layers lie between the input and output layers. In the case of Fig. 1(b), there are two hidden layers. In particular, we will call this NN architecture 2-3-3. 2 represents two neurons in the input layers. 3-3 represents two hidden layers with 3 neurons each. It is redundant to repeatedly mention the output layer as the node is always 1. The neurons in hidden layers represent no physical meaning. They act as a functional form to predict the atomic energy. There is no limit to the number of hidden layers. However, the flexibility of NNP will depend on the number of neurons present in the NN architecture. The connectivity in between the neurons are the weight parameters (fitting parameters). Mathematically, one can calculate the value of a neuron in this form,

$$X_{n_i}^l = a_{n_i}^l\left(b_{n_i}^{l-1} + \sum_{n_j=1}^{N} W_{n_j,n_i}^{l-1,l}\cdot X_{n_j}^{l-1}\right). \quad (14)$$

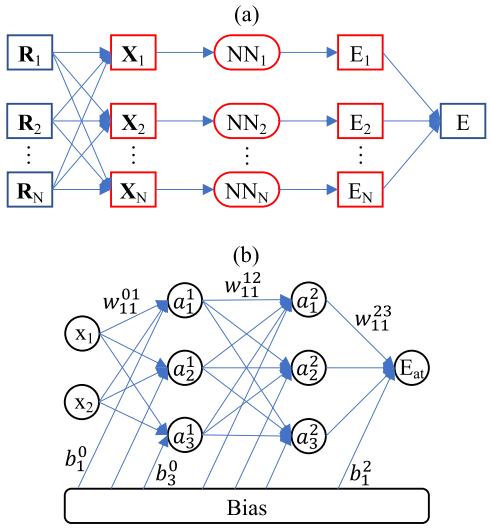

FIG. 1. (a) A schematic diagram of the high-dimensional neural networks. The red diagrams are parts of (b) the neural network architecture. Each atom in a structure is first mapped into atom-centered descriptors according to the atomic environment of the structure. The atom-centered descriptors serve as inputs in the neural network architecture that outputs the atomic energy. Finally, the collection of the atomic energies is the total energy of the structure.

The value of a neuron ($X_{n_i}^l$) at layer l can be determined by the relationships between the weights ($W_{n_j,n_i}^{l-1,l}$), the bias ($b_{n_i}^{l-1}$), and all neurons from the previous layer ($X_{n_j}^{l-1}$). $W_{n_j,n_i}^{l-1,l}$ specifies the connectivity of neuron n_j at layer $l-1$ to the neuron n_i at layer l. $b_{n_i}^{l-1}$ represents the bias of the previous layer that belongs to the neuron n_i. These connectivities are summed based on the total number of neurons (N) at layer $l-1$. Finally, an activation function ($a_{n_i}^l$) is applied to the summation to induce non-linearity to the neuron ($X_{n_i}^l$). X_{n_i} at the output layer is equivalent to an atomic energy, and it represents an atom-centered descriptor at the input layer. Since the atomic energy has no reference value to the DFT energy, each atomic energy is collected as in Eq. (7) to obtain the total energy of a crystal structure. The accuracy of NNP will rely on the accuracy of the NN architecture to predict the energy.

To train the NNP, we can consistently use the cost function in Eqs. (12) and (13). The minimization problem is then solved by our in-house stochastic gradient descent and Adaptive Moment Estimation (ADAM)[45] optimizer. Alternatively, we interfaced our in-house code with the SciPy package,[46] so it is possible to use the Limited-memory Broyden-Fletcher -Goldfarb-Shanno (L-BFGS) method[47] for this study.

III. RESULTS

In this section, we discuss about the development of accurate and transferable MLFFs. First, we introduce two types of datasets—a localized dataset and a diverse dataset. Second, we will validate our machine learning framework with the localized dataset as the baseline. Third, we explore the interplay between bispectrum coefficients and the two machine learning regressions (generalized linear regression and NN) on the localized dataset. This subsection is dedicated

to further validate the localized dataset with a new NNP fitting strategy. Finally, we will develop a transferable silicon MLFF based on the new strategy.

A. Data sets

Here, we present two silicon datasets. Set #1 is the localized dataset, obtained from Ref. 19. Set #1 contains 244 structures in total (219 for training and 25 for test), which includes the ground state of crystalline structure, strained structures, slabs, vacancy, and liquid configurations from MD simulations. To generate the diverse dataset, we utilized our in-house PyXtal code[33] to produce thousands of silicon structures with various numbers of atoms in the unit cell from 1, 2, 4, 6, 8 to 16. Random space group (1–230) assignment was applied to these silicon structures. For each random structure, we performed four consecutive geometry optimization steps at the level of DFT with a steady increase in precision. The maximum numbers for each ionic step were 10, 25, 50, and 50. The relaxed images were then selected to our training pool to represent the shape of PES toward the energy minima. With this scheme, we ensured that not only the minima but also the configurations around the minima would be captured during the energy fitting. Afterward, we performed single-point DFT calculations for all configurations in the training pool using the parameters described in Sec. II A. Finally, 5352 silicon structures (Set #2) were selected by removing structures with energies that are higher than −4.000 eV/atom (i.e., 1.400 eV/atom higher than the ground states). In total, Set #1 has 15 078 atoms, and Set #2 has 31 004 atoms. We note that the energy cutoff (600 eV) used in our DFT calculation is slightly higher than the one (520 eV) used in Ref. 19. However, this resulted in negligible differences according to our test for the same structures. Therefore, we will use these two datasets for direct comparison in Secs. III D– III F.

As shown in Fig. 2, Set #2 covers more diverse atomic environments in terms of energy, force, and density. Set #1 includes 244 structures that span from −4.560 eV/atom to −5.425 eV/atom in energy and 17.56 Å^3/atom to 40.89 Å^3/atom in density. The energy of Set #2 ranges from −4.000 eV/atom to −5.425 eV/atom, and the density ranges from 8.295 Å^3/atom to 52.81 Å^3/atom. The force distribution in Set #2 is wider than that in Set #1. In order to probe the similarity between the two datasets, we further assessed them with the principal component analysis (PCA) technique. While the projection of two most dominating principal components is shown in Fig. 2, the principal components were fitted with the bispectrum coefficients mapped from the Set #2 structures. The inset shows that the data points of Set #1 cover mostly the empty space in the concentrated area. In other words, it appears to be that Set #1 and Set #2 rarely overlap. This indicates that two datasets encompass different atomic environments, which is expected since two different strategies were employed in generating the atomic configurations. Therefore, the two datasets are complementary and can be used to cross-validate each other in the MLFF development.

It is important to note that these two sets of data were obtained through entirely different approaches. Set #1 was not designed to generate an accurate force field for Si but rather to compare different MLFFs on a small, standardized dataset applicable to several elemental systems (for example, only 60 snapshots from *ab initio* MD were included into it). In a typical MLFF development, a few thousand or more configurations will be needed for both the Gaussian process[21,48] and NN regressions.[20,27] Therefore, the training results from Set #1 are expected to gain some improvement by employing a larger version of Set #1 with the same strategy (e.g., adding more MD snapshots). However, many other features in the PES will remain missing. Compared to Set #1, Set #2 covers more energy basins in the PES since it was obtained from an unbiased

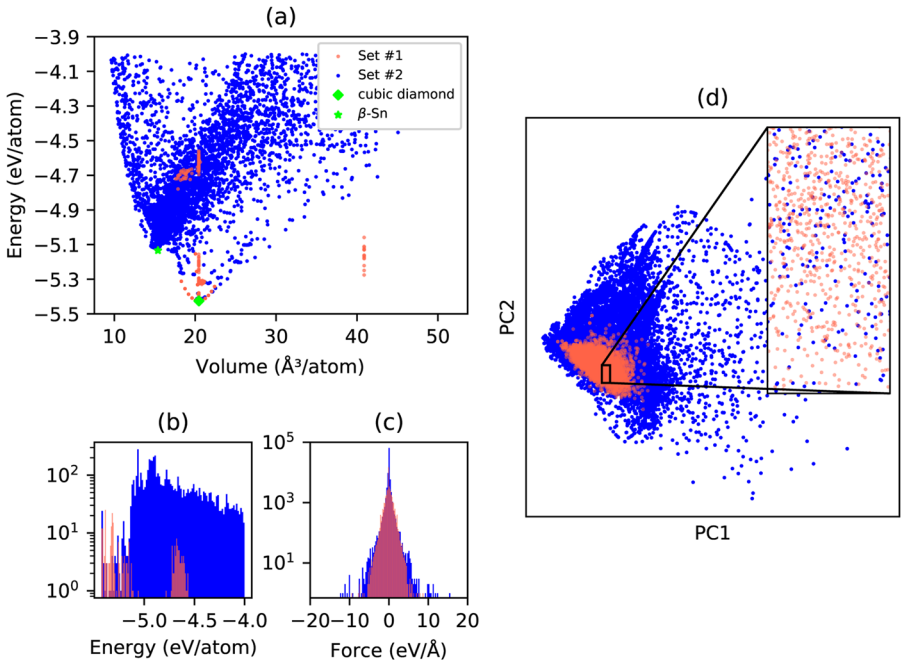

FIG. 2. (a) The energy vs volume plot for training Set #1 and Set #2. The histograms of energy and forces are presented in (b) and (c), respectively. (d) The projection of two most dominating principal components of the atomic bispectrum coefficients. The inset illustrates a zoomed-in view of the concentrated area. In the area, Set #1 is highly concentrated, whereas Set #2 is more widely spread.

and more uniform sampling. For instance, we found that Set #2 contains the high pressure β-Sn phase of silicon and many other phases with five- and six-coordinated silicon atoms. While such atomic environments can also exist in silicon grain boundaries and other types of defects generated from high temperate MD simulations, the MLFF training may not describe these atomic energetics accurately when it attempts to fit the total energy of the system. Therefore, we believe that a MLFF with better coverage of the PES landmarks by small structures is more effective for an accurate modeling of rare events under various conditions (e.g., phase transitions, pronounced deformations, and chemical reactions). As we will discuss in Sec. III E, fitting on Set #2 is considerably more challenging than Set #1. While many relatively simple models can yield satisfactory errors for Set #1, the overall accuracy for Set #2 is notably lower, regardless of the machine learning methods. Therefore, our goal of this work is to fit a Si-MLFF, which can describe Set #2 reasonably well while retaining a similar level of accuracy for Set #1. Prior to this, we will verify our MLFF implementation with Set #1 in Sec. III B.

B. Verification with the localized data set

In Ref. 19, the authors presented an extensive benchmark for silicon (as well as several other elemental systems) with different MLFF approaches. This provided us a foundation to verify our MLFF implementations by using their data for training and testing. With Set #1, we attempted to reproduce the results based on the NNP, SNAP, and quadratic SNAP (qSNAP) methods. To compute the descriptors, we employed the same parameter setting as reported in Ref. 19, which is summarized in Table I. In the original literature, there were 9 G^2 and 18 G^4 descriptors. We made a deeper inspection on the histogram of the computed symmetry functions of the entire Set #1. We identified that descriptors with large η values span in a very narrow range. Narrow-range descriptors were less likely to discriminate different local atomic environments, and they could introduce numerical noise. Therefore, we reduced the parameter set, which included only 6 G^2 and 8 G^4 descriptors for this study. The reduced parameter sets are marked with asterisk symbols. For convenience, we are naming the full Behler–Parrinello descriptors as G27 and the reduced Behler–Parrinello descriptors as G14. For bispectrum coefficients, the expansion is limited to several finite orders since the higher indices of l can only be beneficial in detecting subtle signals on the neighbor density map. In this study, we only considered the band limit ($l_{\max}$) of up to 8, with focus on 3, 4, and 5 (30, 55, and 91 bispectrum coefficients). They are denoted as B30, B55, and B91. Furthermore, we investigated the case of B with normalization, and they are denoted as $\hat{B}30$, $\hat{B}55$, and $\hat{B}91$. Correspondingly, the labelings with the regression techniques are NNP+G27 for the NN regression with G27 descriptors, LR+B55 for linear regression with B55 descriptors, and QR+B55 for quadratic regression with B55 descriptors.

For the cases of linear and quadratic regressions, the results are deterministic as long as the force coefficient in Eq. (12) is given. Figure 3 displays the gradual changes in mean absolute error (MAE) values for energy and forces by varying the force coefficient (β) from 1×10^{-6} to 1×10 for both LR+B55 and QR+B55. For each regression, these points seem to form a Pareto front. Namely, there is no single point that can beat the other points in both energy and force MAE values. Here, we choose a range from the Pareto front that leads to an approximately even change on other sides. This point corresponds to the force coefficient closest to 1×10^{-4}. When $\beta = 1 \times 10^{-4}$, B55+LR yields the MAE values of 6.94 (6.28) meV/atom for energy and 0.11 (0.12) eV/Å for force in the training (test) dataset. For B55+QR, the results gain significant improvement. The final energy MAE value is 2.50 (2.21) meV/atom, and the force MAE value is 0.06 (0.08) eV/Å. The results are expected since the quadratic form allows the coupling of bispectrum coefficients.[12] However, the number of weight parameters also increases notably from 56 to 1596,

TABLE I. The setting used to compute the atom-centered descriptors in this study. The Behler–Parrinello descriptors are consistent with Ref. 19, except that R_c was set to 4.8 Å for the quadratic regression in the previous literature. Moreover, we considered bispectrum coefficients with the band limit $l_{\max}$ up to 8. The asterisk symbol denotes the reduced parameter set for Behler–Parrinello descriptors.

Descriptors	Parameters	Values
G^2	R_c (Å)	5.2
	R_s (Å)	0
	η (Å^{-2})	0.036*, 0.071*, 0.179*, 0.357*, 0.714*, 1.786*, 3.571, 7.142, 17.855
G^4	R_c (Å)	5.2
	λ (Å)	−1, 1
	ζ	1
	η (Å^{-2})	0.036*, 0.071*, 0.179*, 0.357*, 0.714, 1.786, 3.571, 7.142, 17.855
B	R_c (Å)	4.9
	$l_{\max}$	2, 3, 4, 5, 6, 7, 8
	Normalization	True and false

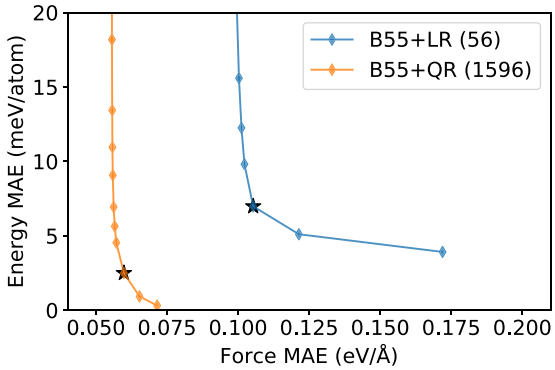

FIG. 3. The comparison of fitting between linear and quadratic regressions based on the B55 descriptors ($l_{\max}$ = 4) applied to Set #1. For each regression, the energy MAE and force MAE values were collected by gradually varying the force coefficients from 1×10^{-6} to 1. The numbers of weight parameters are given in parentheses. The marked black asterisks correspond to the results when the force coefficient is at 1×10^{-4}.

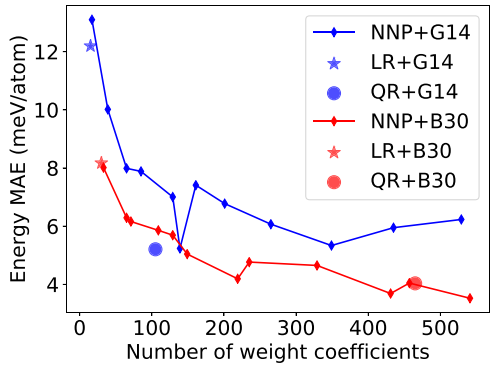

FIG. 4. The performance of NN regression on G14 and B30 as a function of weight parameters. For comparison, the results from linear and quadratic regressions are also included.

which increases the computational cost for both FF training and prediction.

For NNP+G27, we tested the NNP fitting with the NN architecture of 27-24-24. The predicted MAE values are 5.65 meV/atom in the training dataset and 5.60 meV/atom in the test dataset. The metrics are close to the previously reported values: 5.88 meV/atom and 5.60 meV/atom in Ref. 19. Our force MAE values are 0.095 eV/Å and 0.106 eV/Å, agreeing with the previous report as well. Furthermore, we employed reduced Behler–Parrinello descriptors to the NNP fitting (NNP+G14). It is found that the training with NNP+G14 also yielded comparable metrics. This indicated that the removed Behler–Parrinello descriptors were indeed redundant, and they can cause numerical noise during the NNP training. Correspondingly, we adjusted our NNP training strategy toward G14 to investigate the impacts of hyperparameters on NNP training. In contrast to linear regression, the NNP training is much less vulnerable to the choice of force coefficient since the NNP can compromise for more flexible functional forms. It is rather reliant to the hidden layer size. Figure 4 shows the energy MAE values scanning across the hidden layer sizes for NNP+G14 with β fixed at 0.03. The overall picture suggests that NNP performances tend to improve as the NNP model becomes more flexible. However, the NNP accuracy will saturate at some point. Beyond the saturation point, increasing the hidden layer size will only raise the computational cost and lower the chance of finding optimal weight parameters. We also mention that the results from QR+B14 yield better performance than NNP with the same number of parameters. In principle, NNP should be able to self-learn a model similar to QR with the same number of weight parameters. However, different NNP trainings from different initial random guesses may yield somewhat less optimal solutions. This practice suggests that quadratic regression can be an alternative approach when the descriptor size is relatively small.

TABLE II. The comparison of mean absolute error (MAE) values between this work and Ref. 19 for the same 244 Si dataset (Set #1). The results from Ref. 19 are shown in parentheses. For LR+B55 and QR+B55, the results are shown when the force coefficient is at 1×10^{-4}. For the NNP fitting, we used NN architectures of 27-24-24 and 14-12-12.

Fitting method	Train energy (meV/atom)	Test energy (meV/atom)	Train force (eV/Å)	Test force (eV/Å)
LR+B55	6.94 (6.38)	6.28 (6.89)	0.11 (0.21)	0.12 (0.22)
QR+B55	2.50 (3.98)	2.21 (3.81)	0.06 (0.18)	0.08 (0.17)
NNP+G27	5.65 (5.88)	5.60 (5.60)	0.09 (0.12)	0.11 (0.11)
NNP+G14	5.95	6.33	0.10	0.11

The results of verification with different training strategies are summarized in Table II. Compared to Ref. 19, our results are close or maybe slightly better, especially in the force performances for generalized linear regression. Therefore, we proceed to make further investigations on Set #1 by using different strategies.

C. Bispectrum coefficients and algorithms interplay

In this section, MLFF fitting with bispectrum coefficients will be discussed in detail by using both generalized linear and NN regressions on Set #1. First, the performances of generalized linear regression can be improved based on the normalization factor of bispectrum coefficients prior to the MLFF fitting. In the original implementation of SNAP,[11] the bispectrum coefficients are not normalized prior to the MLFF fitting. However, Fig. 5 shows the

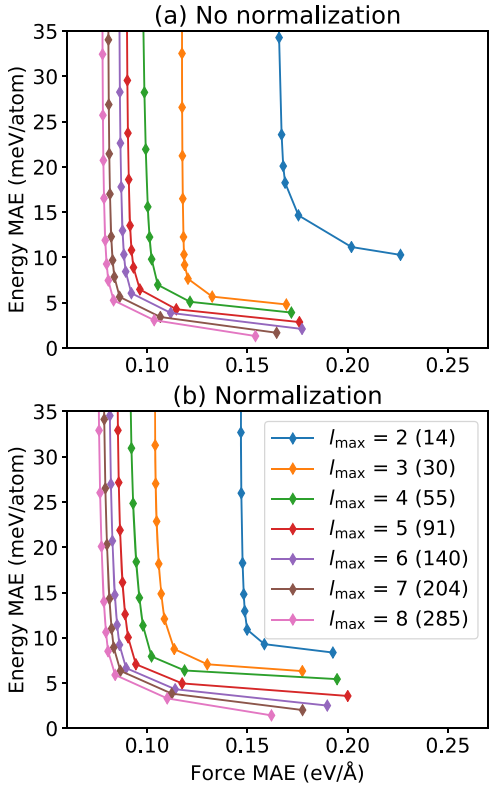

FIG. 5. The performance of linear regression based on the bispectrum coefficients without (a) and with normalization (b). In each plot, l_{max} values from 2 to 8 were considered. The number of descriptors is given in parenthesis.

benefits of normalization prior to the MLFF fitting. Linear regression achieves better performances for both energy and forces as l_{max} increases. At $l_{max} > 5$, there are no significant gains in the MAE values as the computational cost increases. The insignificance of normalization can be due to the limitation of linear regression ability to express the complexity.

Second, Fig. 4 shows the overall NNP fitting with bispectrum coefficients as the inputs to the neural network architecture. The results of NNP+B30 are trained with different hidden layer sizes. The best accuracy is achieved with the hidden layer size of [24, 24]. The 30-24-24 architecture consists of 1369 parameters in total. The training MAE values are 3.18 meV/atom and 0.07 eV/Å, and the test MAE values are 3.54 meV/atom and 0.08 eV/Å. These metrics reach comparable values to that from QR+B55 (see Table II) with few bispectrum coefficients. For reference, linear regression and quadratic regression results with the corresponding number of bispectrum coefficients are also marked in Fig. 4. NNP with bispectrum coefficients can gain notable improvements in comparison to linear regression and quadratic regression. The improvements are expected since NN allows more flexible functional forms to describe the deviation from linearity. Meanwhile, quadratic regression achieves significant improvement in accuracy compared to linear regression due to the extended polynomial forms. However, similar accuracy can be attained with NNP fitting with a smaller number of weight parameters.

D. Transferability of the MLFF from a localized data set

Our in-house code has the ability to apply various descriptors and regression techniques to train MLFF with satisfactory accuracy (energy MAE of <10 meV/atom and force MAE of <0.15 eV/Å) on Set #1. From computational perspective, bispectrum coefficients can cover more orthogonal sets and are easier to be expanded. Therefore, we focus on the use of bispectrum coefficients as the main descriptors from now on. Using the MLFF trained on Set #1, we tried to validate the prediction power on Set #2 (the more diverse dataset). The models include NNP with the 30-10-10 architecture (431 parameters, with β at 0.03), linear regression (31 parameters), and quadratic regression (528 parameters). The three scenarios use normalized bispectrum coefficients with l_{max} of 3 as normalized bispectrum coefficients suggest slight accuracy in improvement. Table III summarizes the results. In general, the prediction power of the MLFF on Set #2, especially in energy, is still poor, though the force errors are acceptable. It is not surprising as the machine learning ability in extrapolation is known to be poor. The performance of the MLFF yields great accuracy based on the given training dataset. The characteristic of atomic environments of Set #2 is too broad, and most of the data points lay outside of Set #1. Therefore, the predicted energy and force are no longer reliable.

TABLE III. The MAE values of the predicted energy and forces of Set #2 by training on Set #1. The NN architecture of 30-10-10 is used for providing comparable weight parameters as the quadratic regression. The numbers inside parentheses are the test MAEs.

	NN	LR	QR
Energy (meV/atom)	4.7 (70)	7.5 (110)	4.0 (265)
Force (eV/Å)	0.08 (0.13)	0.12 (0.15)	0.08 (0.21)
Number of parameters	431	31	496

TABLE IV. Comparison of different machine learning potentials of Si shown in RMSE values.

	QR	NN	GPR[29]	DL[49]
Energy (meV/atom)	9.7	14.8	20.2	N/A
Force (eV/Å)	0.22	0.16	0.25	0.12

Despite the unsatisfactory accuracy, some insights can be gained from this numerical experiment. NN regression can achieve better transferability in comparison to linear and quadratic regressions. Although the quadratic regression yields the best accuracy in training (3.99 meV/atom energy and 0.08 eV/Å force), it also produces the largest error on the test set. On the contrary, NN regression achieves a similar level of accuracy in training (4.70 meV/atom energy and 0.08 eV/Å force). However, the errors on the test set (69.8 meV/atom energy MAE and 0.13 eV/Å force MAE) are much smaller. This can be partially explained by the fact that NN adopts more flexible functional forms during fitting.

E. Training with data from random structure generator

For the sake of data diversity, it is more natural to train the MLFF based on Set #2 and test its performance on Set #1. To train reliable MLFF on Set #2, we decided to use more bispectrum coefficients and a larger NN architecture and test on Set #1. In addition, polynomial fittings were included again for the purpose of comparison. For polynomial regression, l_{max} at 5 with a cutoff radius of 4.9 Å was applied. According to Fig. 5, normalizing the bispectrum coefficients had a negligible effect on the results. Hence, normalization was ignored. The β value was fixed at 1×10^{-4} for quadratic regression and 1×10^{-3} for linear regression. The NN architecture of 91-34-34 was used to give comparable weight parameters as the quadratic regression.

Figure 6 summarizes the results of Set #2 training. In terms of energy, quadratic regression performs the best accuracy (5.90 meV/atom), whereas NNP can predict less accurate energy (9.81 meV/atom) but better forces (0.08 eV/Å). It should be emphasized that Set #2 contains a smaller unit cell (1–16 atoms in a unit cell) than Set #1 (up to 64 atoms in a unit cell). This transition from smaller to larger cells can introduce long-range effects that were not accounted for in the training.[49] Therefore, the MAE values on the test set are consistently larger than the training set. Furthermore, Set #1 may contain some manually selected atomic configurations. These configurations may not be fully covered by our random generated structures. While linear regression predicts well on the forces, it guides the energy predictions to unsatisfactory results. This may be due to the limitation of the regression technique as the smaller number of parameters fails to describe the true PES. Therefore, our recommendation is to use either quadratic regression (similar to the recently proposed qSNAP method[12]) or NN for a better fitting of

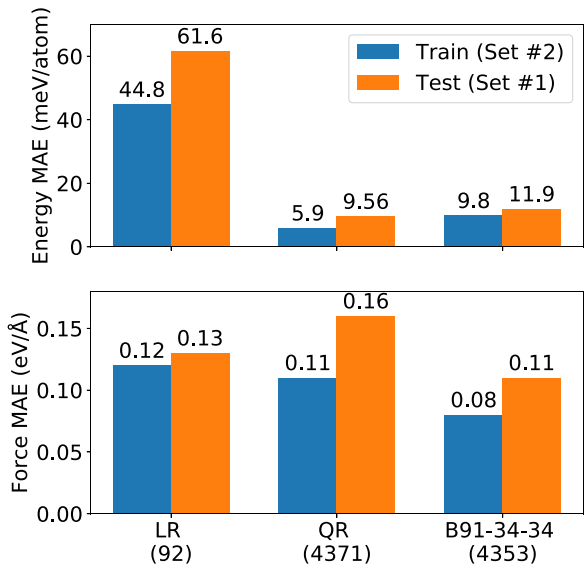

FIG. 6. The performance of trained MLFF on Set #2.

a diverse dataset. Compared to the quadratic regression, NN is our preferred choice due to its flexibility.

Table IV shows our results in comparison to studies.[29,49] The authors juxtaposed among several atom-centered descriptors, including bispectrum coefficients, which were coupled with Gaussian process regression. In particular, the root mean square errors (RMSEs) for energy and forces with l_{max} at 5 are 20.2 meV/atom and 0.25 eV/Å. Meanwhile, the training RMSEs of our quadratic regression yield 9.7 meV/atom and 0.22 eV/Å and the training RMSEs of the 91-34-34 architecture are 14.8 meV/atom and 0.16 eV/Å. In another study, Kuritz *et al.* focused on training atomic forces using a deep learning model with the environmental distances as the descriptors. The force predictions are performed at a scaling from 16 atoms to 128 atoms, yielding a MAE of 0.12 eV/Å,[49] given that the NN nodes are in the order of 10^3/layer. This phenomenon proves that the choice of descriptor can reduce the complexity of MLFF.

F. Physical properties

One of the critical requirements for MLFFs is to predict basic material properties, including but not limited to lattice parameters, elastic constants, and bulk moduli of diamond cubic Si. To obtain the elastic constants, we computed the stress–strain relation and fitted the relation to a set of linear equations built from the symmetry. For each applied deformation, the geometry of the structure was optimized to gain a net force of zero. The summary of the properties is tabulated in Table V.

First, it is crucial to validate our code on Set #1. In the column of Set #1 in Table V, the performances of the MLFFs are presented with different training strategies: energy-force linear regression (EF-LR), energy-force-stress linear regression (EFS-LR), energy-force quadratic regression (EF-QR), energy-force-stress quadratic regression (EFS-QR), energy-force NN (EF-NN), and energy-force-stress NN (EFS-NN). All of the training involved bispectrum coefficients as the descriptor. Linear and quadratic regressions used bispectrum coefficients with l_{max} of 4, whereas NN used l_{max} of 3. Here, we used the NN architecture of 30-10-10. Moreover, EF was trained with DFT energy and forces only as the reference values, while EFS included the DFT stress information in the training. Without stress involvement, the quadratic regression performances are the closest to the DFT values. Seemingly, linear and NN regressions fail to extrapolate the C_{12}. However, the C_{12} values tend to get closer to the DFT with tiny sacrifice in the accuracy of C_{11}, when stress is involved.

Second, without stress information, linear and quadratic regressions are considered to be more transferable in predicting the physical properties of Set #2. Evidently, linear regression gains no prominent refinement without trade-off between elastic constants as stress information is added. However, the values are the closest to the experimental values. On the other hand, quadratic regression exhibits accuracy boosts in C_{11} and the lattice constants in comparison to the DFT. As stress training is employed, NNP seems to benefit the most in terms of transferability. Consequently, it is crucial to include stress tensors during the training of NNP.

TABLE V. The experimental elastic constants[50] of cubic-diamond silicon are shown at zero-Kelvin values, while the DFT data are obtained from Ref. 19. In comparison to the Gaussian approximation potential (GAP) of Si,[25] the GAP results are shown below. The numbers of weight parameters are displayed in parentheses. EF and EFS stand for energy-force and energy-force-stress training. LR, QR, and NN are linear, quadratic, and neural network regressions, respectively. The NN architecture is 30-10-10 for Set #1 and 91-34-34 for Set #2.

				Set #1						Set #2					
	Expt.	DFT	GAP	EF-LR (56)	EFS-LR (56)	EF-QR (1596)	EFS-QR (1596)	EF-NN (431)	EFS-NN (431)	EF-LR (91)	EFS-LR (91)	EF-QR (4371)	EFS-QR (4371)	EF-NN (4353)	EFS-NN (4353)
a (Å)	5.429	5.469	...	5.467	5.466	5.462	5.467	5.473	5.468	5.415	5.469	5.503	5.468	5.509	5.467
C_{11} (GPa)	167	156	153	153	151	149	152	157	154	137	167	173	158	167	153
C_{12} (GPa)	65	65	56	100	62	60	57	96	58	76	73	55	55	128	57
C_{44} (GPa)	81	76	72	69	70	75	75	66	68	73	85	81	71	43	76
B_{VRH} (GPa)	99	95	89	118	92	90	89	117	90	96	104	94	89	141	89

IV. DISCUSSION

A. Training dataset

In general, MLFF lacks extrapolative ability, unlike the traditional force field method. The training dataset plays an extremely important role in MLFF development. A more complete dataset can grant the trained MLFF with more powerful predictive ability. The use of randomly pre-symmetrized crystal structures is able to produce a dataset with highly diverse atomic distribution.[21,33,51] In this work, we prepared the training dataset from crystal structure prediction techniques. However, several recent works demonstrated that the MLFF can also be trained on-the-fly.[21,22] In addition to generating random structures, advanced sampling techniques such as metadynamics[10,52] and stochastic surface walking[27] have also been used to provide the training data for MLFF development. In general, each method focuses on different aspects of the PES. For instance, the surface walking method may work better in describing the transition path between different low energy configurations, while random structure generation offers more energy basins of the PES. On the other hand, the metadynamics method excels at describing the liquid and amorphous states. At the moment, it remains challenging in obtaining a universal MLFF to fully replace DFT simulation for general purpose.[25] Given the increasing power of regression techniques such as deep learning,[53,54] it will be interesting to know if a MLFF can ultimately achieve the DFT accuracy by considering all training configurations from different sampling techniques.

B. Limitation of the objective function

A typical DFT calculation outputs the total energy for each configuration. Thus, the MLFF is trained to describe the total energy of a structure. However, it is possible that the MLFF fails to distinguish the atomic energies for the trained structures.[55] To prevent the incorrect energy decomposition, one can either develop the approach to extract the site energy from the DFT simulation[56] or intentionally prepare the structures with nearly identical atomic environments in the training. Random crystal structure generation with pre-symmetrization follows the latter. Therefore, Set #2 includes many structures with smaller unit cells to allow for better descriptions of the PES. Hence, this can help the performance in predicting the total energy. In addition, Set #2 can be further extended to consist of more varieties of atomic environments to enhance the capability of the current NNP. For example, it was shown above that adding stress tensors can help improve elastic constant predictions.

C. Descriptors

As the complexity of a system's PES increases, different atomic descriptors can yield different accuracy in MLFF development.[29] For instance, thousands of nodes are needed to achieve similar accuracy in NNP transferability,[49] compared to 34 nodes in this study. The key to extract reliable descriptors is by reconstructing the atomic neighbor density function. The expansion of bispectrum coefficients as the descriptor is more straightforward to be applied than the Behler–Parrinello descriptors. Nevertheless, it is important to take account of the relation between computational cost and accuracy in MLFF training. The current MLFF is developed through the reconstruction of the neighbor density function, which is described by the Dirac δ function. The full description of the true neighbor density can only be partially represented by the finite spherical harmonic expansion. In addition, it is numerically unstable to compare the differences between two δ functions. A better design of the descriptor uses smooth Gaussian functions to express the atomic neighbor density, as recently developed in the SOAP method.[29] The comparison between SO(4) bispectrum and SOAP descriptors for NNP development will be made in the future code development. Moreover, other similar types of descriptors, such as moment tensor potential (MTP),[14] will be investigated in the future.

D. Fitting scheme

Linear regression, as the simplest method in curve fitting, has been used in developing several MLFFs.[11,14] In particular, the MTP approach[19] can predict energy and forces with great accuracy while maintaining acceptable computational cost. The advantage of the linear regression method lies in its simple algorithm, which provides easy and fast computation. Despite the simplicity, linear/quadratic regressions are usually sensitive to the noise in the dataset. In this work, we focused on NN regression since it has more flexibility, which can yield better accuracy. Compared to the linear/quadratic regressions, including stress training in NNP is critical to promote the transferability. Beside NN, some non-parametric regression techniques, such as Gaussian process regression, have also been proved to be efficient in MLFF development.[16] However, this is beyond the scope of the current study.

E. Applicability

For the purpose of MD simulation around the equilibrium state, fitting the MLFF with a localized dataset generated from MD simulation is, perhaps, sufficient. However, the primary goal of this work is to generate high quality silicon MLFF for a more general purpose, which requires a complete description of PES for a given chemical system. As discussed above, the MLFF trained with Set #2 is generally capable of describing the entire PES of the crystalline system better. We expect that the MLFF generated in this work can be used to replace DFT simulation in predicting the structures of crystalline silicon, given that similar works have been done in several elemental systems.[21,22] Yet, one needs to keep in mind that the quality still depends on the coverage of training dataset. For instance, additional data are needed to enable the prediction for surfaces and clusters.[25] Moreover, the trained MLFF may not be able to describe the high energy configurations well since Set #2 only contains structures with energy less than 1.400 eV/atom from the ground state. It was found that some nonphysical configurations (e.g., short distances and overly clustered) may be favored under high temperature MD simulations. In this case, it is useful to either add a few explicit two-body and three-body terms to prevent the nonphysical configurations[48] or include some highly strained configuration in the training. We will consider the combination of physical and machine learning terms in the training and investigate the applicability.

V. CONCLUSIONS

In summary, we present a systematic investigation of MLFF fitting for elemental silicon using our in-house code. The silicon

MLFFs are developed by implementing different regression techniques based on Behler–Parrinello and bispectrum coefficients as the descriptors. The MLFFs trained with Set #1 (the localized dataset) can be described accurately in both energy and forces using generalized linear regression and NN based on both descriptor choices. Among the MLFFs, fitting NNP with the bispectrum coefficients is the most favorable option. This is due to the expansion of bispectrum coefficients being more straightforward than Behler–Parrinello descriptors. In addition, NNP provides a more flexible framework in which the functional form can be easily adjusted by adding/reducing the size of weight parameters. For Set #2 generated from random symmetric structures, the NNP fitting with bispectrum coefficients achieves accuracy at 9.8 meV/atom for energy and 80 meV/Å for force, which is comparable to the current state of the art based on other approaches. A thorough study on the applicability of Set #2 silicon MLFF on more challenging simulations such as crystal structure search will be the subject of our future work.

ACKNOWLEDGMENTS

The authors acknowledge the NSF (I-DIRSE-IL, Grant No. 1940272) and NASA (Grant No. 80NSSC19M0152) for financial support. H.Y. was also supported by the Science Graduate Student Research (SCGSR) program, which is administered by the Oak Ridge Institute for Science and Education (ORISE) for the DOE under Contract No. DE-SC0014664. The computing resources are provided by XSEDE (No. TG-DMR180040). A portion of this work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344. The authors thank Dr. A. Thompson at Sandia for insightful discussions on the computation of bispectrum coefficients. They also thank the anonymous referees for excellent suggestions during the revision.

DATA AVAILABILITY

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

REFERENCES

[1] K. Lejaeghere, G. Bihlmayer, T. Björkman, P. Blaha, S. Blügel, V. Blum, D. Caliste, I. E. Castelli, S. J. Clark, A. Dal Corso *et al.*, "Reproducibility in density functional theory calculations of solids," Science **351**, aad3000 (2016).
[2] S. Berber, Y.-K. Kwon, and D. Tománek, "Unusually high thermal conductivity of carbon nanotubes," Phys. Rev. Lett. **84**, 4613 (2000).
[3] V. Yamakov, D. Wolf, S. R. Phillpot, A. K. Mukherjee, and H. Gleiter, "Dislocation processes in the deformation of nanocrystalline aluminium by molecular-dynamics simulation," Nat. Mater. **1**, 45 (2002).
[4] V. Yamakov, D. Wolf, S. R. Phillpot, A. K. Mukherjee, and H. Gleiter, "Deformation-mechanism map for nanocrystalline metals by molecular-dynamics simulation," Nat. Mater. **3**, 43 (2004).
[5] A. R. Oganov, C. J. Pickard, Q. Zhu, and R. J. Needs, "Structure prediction drives materials discovery," Nat. Rev. Mater. **4**, 331–348 (2019).
[6] S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, "The high-throughput highway to computational materials design," Nat. Mater. **12**, 191 (2013).
[7] A. P. Bartók, M. J. Gillan, F. R. Manby, and G. Csányi, "Machine-learning approach for one-and two-body corrections to density functional theory: Applications to molecular and condensed water," Phys. Rev. B **88**, 054104 (2013).
[8] N. Artrith, T. Morawietz, and J. Behler, "High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide," Phys. Rev. B **83**, 153101 (2011).
[9] R. Z. Khaliullin, H. Eshet, T. D. Kühne, J. Behler, and M. Parrinello, "Nucleation mechanism for the direct graphite-to-diamond phase transition," Nat. Mater. **10**, 693 (2011).
[10] J. Behler, R. Martoňák, D. Donadio, and M. Parrinello, "Metadynamics simulations of the high-pressure phases of silicon employing a high-dimensional neural network potential," Phys. Rev. Lett. **100**, 185501 (2008).
[11] 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).
[12] M. A. Wood and A. P. Thompson, "Extending the accuracy of the snap interatomic potential form," J. Chem. Phys. **148**, 241721 (2018).
[13] S. Pozdnyakov, A. R. Oganov, A. Mazitov, I. Kruglov, and E. Mazhnik, "Fast general two-and three-body interatomic potential," arXiv:1910.07513 [physics.comp-ph] (2019).
[14] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," Multiscale Model. Simul. **14**, 1153–1173 (2016).
[15] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).
[16] A. P. Bartók and G. Csányi, "Gaussian approximation potentials: A brief tutorial introduction," Int. J. Quantum Chem. **115**, 1051–1057 (2015).
[17] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).
[18] J. Behler, "Constructing high-dimensional neural network potentials: A tutorial review," Int. J. Quantum Chem. **115**, 1032–1050 (2015).
[19] 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).
[20] S. Hajinazar, J. Shao, and A. N. Kolmogorov, "Stratified construction of neural network based interatomic models for multicomponent materials," Phys. Rev. B **95**, 014114 (2017).
[21] V. L. Deringer, C. J. Pickard, and G. Csányi, "Data-driven learning of total and local energies in elemental boron," Phys. Rev. Lett. **120**, 156001 (2018).
[22] E. V. Podryabinkin, E. V. Tikhonov, A. V. Shapeev, and A. R. Oganov, "Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning," Phys. Rev. B **99**, 064114 (2019).
[23] T. L. Jacobsen, M. S. Jørgensen, and B. Hammer, "On-the-fly machine learning of atomic potential in density functional theory structure optimization," Phys. Rev. Lett. **120**, 026102 (2018).
[24] C. Zeni, K. Rossi, A. Glielmo, Á. Fekete, N. Gaston, F. Baletto, and A. De Vita, "Building machine learning force fields for nanoclusters," J. Chem. Phys. **148**, 241739 (2018).
[25] A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, "Machine learning a general-purpose interatomic potential for silicon," Phys. Rev. X **8**, 041048 (2018).
[26] J. E. Herr, K. Yao, R. McIntyre, D. W. Toth, and J. Parkhill, "Metadynamics for training neural network model chemistries: A competitive assessment," J. Chem. Phys. **148**, 241710 (2018).
[27] S.-D. Huang, C. Shang, P.-L. Kang, and Z.-P. Liu, "Atomic structure of boron resolved using machine learning and global sampling," Chem. Sci. **9**, 8644–8655 (2018).
[28] E. L. Kolsbjerg, A. A. Peterson, and B. Hammer, "Neural-network-enhanced evolutionary algorithm applied to supported metal nanoparticles," Phys. Rev. B **97**, 195424 (2018).
[29] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**, 184115 (2013).
[30] Z. Li, J. R. Kermode, and A. De Vita, "Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces," Phys. Rev. Lett. **114**, 096405 (2015).

[31]H. Babaei, R. Guo, A. Hashemi, and S. Lee, "Machine-learning-based interatomic potential for phonon transport in perfect crystalline Si and crystalline Si with vacancies," Phys. Rev. Mater. **3**, 074603 (2019).
[32]L. Bonati and M. Parrinello, "Silicon liquid structure and crystal nucleation from *ab initio* deep metadynamics," Phys. Rev. Lett. **121**, 265701 (2018).
[33]S. Fredericks, D. Sayre, and Q. Zhu, "PyXtal: A python library for crystal structure generation and symmetry analysis," arXiv:1911.11123 [cond-mat.mtrl-sci] (2019).
[34]A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus *et al.*, "The atomic simulation environment—A python library for working with atoms," J. Phys. Condens. Matter **29**, 273002 (2017).
[35]G. Kresse and J. Furthmüller, "Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set," Phys. Rev. B **54**, 11169 (1996).
[36]P. E. Blöchl, "Projector augmented-wave method," Phys. Rev. B **50**, 17953 (1994).
[37]J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," Phys. Rev. Lett. **77**, 3865–3868 (1996).
[38]M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, "wACSF—Weighted atom-centered symmetry functions as descriptors in machine learning potentials," J. Chem. Phys. **148**, 241709 (2018).
[39]G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, "Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials," J. Chem. Phys. **148**, 241730 (2018).
[40]T. D. Huan, R. Batra, J. Chapman, S. Krishnan, L. Chen, and R. Ramprasad, "A universal strategy for the creation of machine learning-based atomistic force fields," npj Comput. Mater. **3**, 37 (2017).
[41]H. Gao, J. Wang, and J. Sun, "Improve the performance of machine-learning potentials by optimizing descriptors," J. Chem. Phys. **150**, 244110 (2019).
[42]S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," J. Comput. Phys. **117**, 1–19 (1995).
[43]M. Boyle, "Angular velocity of gravitational radiation from precessing binaries and the corotating frame," Phys. Rev. D **87**, 104006 (2013).
[44]T. E. Oliphant, *A Guide to NumPy* (Trelgol Publishing, USA, 2006), Vol. 1.
[45]D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in paper presented at 3rd International Conference for Learning Representations, San Diego, 2015, 2014.
[46]P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0—Fundamental algorithms for scientific computing in Python," Nat. Meth. **17**, 261–272 (2020).
[47]C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," ACM Trans. Math. Software **23**, 550–560 (1997).
[48]V. L. Deringer and G. Csányi, "Machine learning based interatomic potential for amorphous carbon," Phys. Rev. B **95**, 094203 (2017).
[49]N. Kuritz, G. Gordon, and A. Natan, "Size and temperature transferability of direct and local deep neural networks for atomic forces," Phys. Rev. B **98**, 094109 (2018).
[50]J. D. Schall, G. Gao, and J. A. Harrison, "Elastic constants of silicon materials calculated as a function of temperature using a parametrization of the second-generation reactive empirical bond-order potential," Phys. Rev. B **77**, 115209 (2008).
[51]A. O. Lyakhov, A. R. Oganov, H. T. Stokes, and Q. Zhu, "New developments in evolutionary structure prediction algorithm uspex," Comput. Phys. Commun. **184**, 1172–1182 (2013).
[52]H. Niu, L. Bonati, P. M. Piaggi, and M. Parrinello, "*Ab initio* phase diagram and nucleation of gallium," Nat. Commun. **11**, 2654 (2020).
[53]K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, "SchNetPack: A deep learning toolbox for atomistic systems," J. Chem. Theory Comput. **15**, 448–455 (2019).
[54]L. Zhang, J. Han, H. Wang, W. Saidi, R. Car, and E. Weinan, "End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2018), pp. 4436–4446.
[55]D. Yoo, K. Lee, W. Jeong, D. Lee, S. Watanabe, and S. Han, "Atomic energy mapping of neural network potential," Phys. Rev. Mater. **3**, 093802 (2019).
[56]Y. Huang, J. Kang, W. A. Goddard, and L.-W. Wang, "Density functional theory based neural network force fields from energy decompositions," Phys. Rev. B **99**, 064103 (2019).