The Journal of Chemical Physics

RESEARCH ARTICLE | JUNE 26 2019

Improve the performance of machine-learning potentials by optimizing descriptors

Hao Gao ; Junjie Wang; Jian Sun ✉

Check for updates

J. Chem. Phys. 150, 244110 (2019)
https://doi.org/10.1063/1.5097293

View Online
Export Citation
CrossMark

Articles You May Be Interested In

Microscopic structural descriptor of liquid water

J. Chem. Phys. (March 2018)

A Java Library for the Calculation of Molecular Descriptors

AIP Conference Proceedings (December 2007)

The Timbre Toolbox: Extracting audio descriptors from musical signals

J Acoust Soc Am (November 2011)

The Journal of Chemical Physics
Special Topic: Adhesion and Friction
Submit Today!
AIP Publishing

AIP Publishing

Improve the performance of machine-learning potentials by optimizing descriptors

Cite as: J. Chem. Phys. 150, 244110 (2019); doi: 10.1063/1.5097293
Submitted: 24 March 2019 • Accepted: 6 June 2019 •
Published Online: 26 June 2019

Hao Gao, Junjie Wang, and Jian Sun[a)]

AFFILIATIONS

National Laboratory of Solid State Microstructures, School of Physics and Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing 210093, China

[a)] Author to whom correspondence should be addressed: jiansun@nju.edu.cn

ABSTRACT

Machine-learning (ML) potentials are promising in atomic simulations due to their comparable accuracy to density functional theory but much lower computational cost. The descriptors to represent atomic environments are of high importance to the performance of ML potentials. Here, we implemented the descriptor in a differentiable way and found that ML potentials with optimized descriptors have some advantages compared with the ones without descriptor optimization, especially when the training dataset is small. Taking aluminum as an example, the trained potentials with proper descriptors can not only predict energies and forces with high accuracy of the first-principles calculations but also reproduce the statistical results of dynamical simulations. These predictions validate the efficiency of our method, which can be applied to improving the performance of machine learning interatomic potentials and will also strongly expand its applications.

I. INTRODUCTION

Molecular simulations have been widely applied to problems in various areas including physics, chemistry, biology, and material science. The reliability of molecule simulations is strongly dependent on the method to describe the potential energy surface. *Ab initio* methods, such as density functional theory (DFT), are popular for its good balance between accuracy and computational cost. However, DFT calculations are still quite expensive for large scale simulations. To reduce the computational cost and model large systems with relatively long time scales, empirical interatomic potentials (or force fields) have been developed to be an effective method. Using this method, the energies and forces can be efficiently calculated in explicit forms after the parameters have been fitted with the *ab initio* or experimental results. Empirical force fields have been successfully applied to many systems although they are still lack of transferability and generality.

Recently, machine learning (ML) algorithms have been implemented to fit more accurate force fields to overcome the shortcomings of empirical force fields. ML interatomic potentials mainly differ from empirical force fields in two aspects. First, the physically meaningful descriptors of atomic environments are replaced by statistical fingerprints to represent more complex environments. Second, the empirical energy functions are replaced by flexible models, such as Neural Network (NN),[1] Gaussian Process Regression (GPR),[2] or linear regression.[3–5] ML interatomic potentials have three advantages over the *ab initio* methods or empirical force fields as follows: (1) After fitting to the *ab initio* results, ML interatomic potentials can be more accurate to describe the total energy and forces of the system than empirical force fields. (2) Calculations using ML interatomic potentials are much less time-consuming than those of the *ab initio* method, so it is effective for simulating large systems and long-time dynamics processes. (3) Directly fitting the relations between the structure and energy, ML interatomic potentials can be applied to diverse systems, overcoming the disadvantages of empirical force fields in transferability and generality. ML potentials have been successfully applied to different systems such as molecules, clusters, bulk materials, surfaces, and solid–liquid interfaces.[6–9]

The fitting process framework of ML potentials is shown in Fig. 1(a), and the atomic positions of the input configuration is first transformed by descriptors to fingerprint vectors. Such fingerprints are then taken as input vectors of a regression model to compute the total energy, forces, and stresses of the atomic structure. Several types of descriptors have been introduced and widely used in recent years, including Behler-Parrinello Symmetry Functions (SFs),[10]

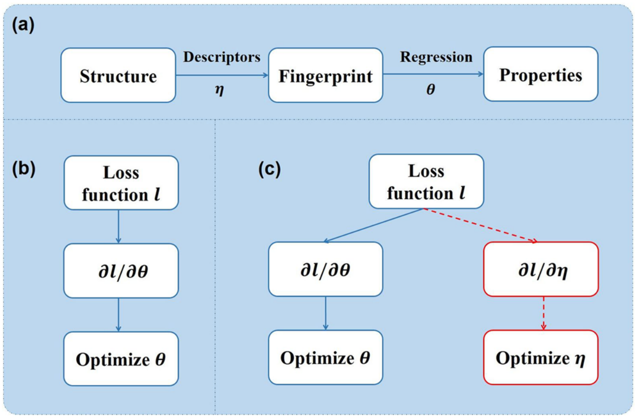

FIG. 1. (a) Prediction process of ML interatomic potentials. (b) Training process of conventional ML potentials (c) Training process in this work; the red path and boxes show the differentiation process to optimize the descriptors.

Bispectrum descriptors,[11] Zernike descriptors,[12] and Moment Tensor descriptors.[4] Among them, the Behler-Parrinello SFs have the simplest forms and high flexibility, thus have been widely used. In Behler-Parrinello SFs, some descriptor parameters are required to distinguish different atomic environments. When fitting a force field for a complex system, the size of descriptor parameters should be large enough and the values of descriptor parameters are of great importance to the performance of ML force fields.

In most of the previous works, only the model parameters have been trained while the optimization of the descriptor parameters is absent, as shown in Fig. 1(b). Therefore, it is still worth to explore in parameter space to find better descriptors. Huan *et al.*[13] have used grid search to find optimal descriptor parameters. Imbalzano *et al.*[14] have performed several methods of automatic selection of fingerprints to improve the speed and accuracy of ML force fields. Gastegger *et al.*[15] have optimized descriptors by a genetic algorithm. However, the gradient-free algorithms may be inefficient compared with gradient-guided ones such as the Adam algorithm[16] used in this article.

In this work, using an automatic differentiation technique, we optimized the descriptor parameters of Behler-Parrinello SFs[10] and fitted interatomic potentials of aluminum (Al) by Gaussian approximation potential (GAP).[17] Comparing the performance of the potentials, we discussed the effect of optimization of descriptor parameters. To test the capability, we further calculated the vibrational and melting properties of Al using our newly fitted ML interatomic potentials. The results are in good agreement with those of DFT and experiments. Most interestingly, we find our strategy has better performance compared to the original ones when the training dataset is small. This makes our model useful in the cases when a big training dataset is not available.

II. METHOD

A. Symmetry functions

The Cartesian coordinates seem not appropriate to be considered as inputs of ML potentials because a translation or rotation will lead to coordinates transformation, which can change the energy output of a configuration. To solve this problem, the fingerprint vectors composed of several Behler-Parrinello SFs[1] are introduced to describe the environment of atoms. These vectors are invariant with respect to translation, rotation, and permutation of atoms with the same species.

From a geometric point of view, the environment of an atom is the configuration of atoms around the central atom in a sphere with a certain cutoff radius R_c. A cutoff function is defined as

$$f_c(R_{ij}) = 0.5 \cdot \left[\cos\left(\frac{\pi R_{ij}}{R_c}\right) + 1\right], \tag{1}$$

where R_{ij} is the distance between atoms i and j. $f_c(R_{ij})$ is continuous and differentiable so that analytical derivatives are available for calculating the forces.

Based on the cutoff function, SFs could be built by encoding radial and/or angular information of neighboring atoms. A function called G^2 has been applied in the previous works,[10] which sums the products of Gaussians and cutoff function

$$G_i^2 = \sum_j e^{-\eta\left(R_{ij}-R_s\right)^2} \cdot f_c(R_{ij}). \tag{2}$$

Here, the widths of Gaussians are defined by a parameter η and the positions are controlled by the shifting parameter R_s. Assigning different values of these parameters, a high-dimensional representation of the environment of atom i can be constructed. Therefore, the total energy in ML interatomic potentials can be written as a sum of atomic contributions,

$$E = \sum_i^N E_i(\mathbf{G}_i), \tag{3}$$

where E_i is the i atomic energy.

In previous works, the values of η components are usually sampled on a logarithmic grid and the differences between η values are too large.[16] In this paper, we have modified G^2 as

$$G_i = \sum_j \exp\left[-\frac{\exp(\eta) R_{ij}}{R_c}\right]^2 \cdot f_c(R_{ij}), \tag{4}$$

which is inspired by Botu *et al.*[18] and Khorshidi and Peterson.[12] Here, we change G^2 by transforming η to e^η. Actually, this modification will not affect the nature of the representation, but after the transformation, η values become arithmetic sequences and are distributed in a narrow domain so that η is relatively easy to be optimized.

The G^2 descriptor is too simple since it only contains radial information of atomic environments. Advanced descriptors, such as Behler-Parrinello G^4 function, consider angular terms to build better potentials. However, it is more difficult to implement those angular-dependent descriptors in a differentiable way. In this work, we only construct differentiable G^2 to test the effect of optimization of descriptor parameters.

B. Potential based on Gaussian process regression

In the framework of Gaussian process regression potentials, the atomic energy contribution E_* is described as a Gaussian random variable depending on $\mathbf{G}_*$. The energy prediction is $\mu(E_*)$, the mean of E_*, which is configured as a linear combination of covariance between E_* and the training set,

$$\mu(E_*(\boldsymbol{G}_*)) = \sum_t \alpha_t \, cov(E_t, E_*) = \sum_t \alpha_t K(\boldsymbol{G}_t, \boldsymbol{G}_*), \tag{5}$$

where t runs over the training set. The covariance $cov(E_*, E_t)$ is computed by the covariance kernel function K which evaluates the similarity of the two input configurations. Here, we use the form of squared exponential kernel

$$K(\boldsymbol{G}_i, \boldsymbol{G}_j) = \exp\left[-\frac{(\boldsymbol{G}_i - \boldsymbol{G}_j)^2}{2\theta^2}\right]. \tag{6}$$

Thanks to the fact that GPR allows derivative information in the training set,[17] we can also calculate covariance between E_i and $\frac{\partial E_i}{\partial \boldsymbol{G}_i}$,

$$cov\left(\frac{\partial E_i}{\partial \boldsymbol{G}_i}, E_j\right) = \frac{\partial K(\boldsymbol{G}_i, \boldsymbol{G}_j)}{\partial \boldsymbol{G}_i}, cov\left(\frac{\partial E_i}{\partial \boldsymbol{G}_i}, \frac{\partial E_j}{\partial \boldsymbol{G}_j}\right) = \frac{\partial^2 K(\boldsymbol{G}_i, \boldsymbol{G}_j)}{\partial \boldsymbol{G}_i \partial \boldsymbol{G}_j}. \tag{7}$$

Equation (5) is only suitable for local energies, but, in practice, DFT codes compute total energies as a sum of atomic energies [Eq. (3)]. Since the observations (total energies, forces, and stresses) are linear combinations of atomic energies and their derivatives, we can compute the covariance between them by using the properties of Gaussian random variables (a is a constant and X, Y, Z are Gaussians),

$$cov(aX, Y) = a \cdot cov(X, Y), cov(X + Y, Z) = cov(X, Z) + cov(Y, Z). \tag{8}$$

Thus, the predictive value of the target observation O_* is expressed as

$$\mu(O_*) = \sum_t \alpha_t cov(O_t, O_*) = \boldsymbol{K}_*^T \boldsymbol{\alpha}, \tag{9}$$

where O_t represents total energies, forces, or stresses in the training set.

The coefficients $\boldsymbol{\alpha}$ are given by

$$\boldsymbol{\alpha} = \boldsymbol{K}_{tt}^{-1} \boldsymbol{y}, \tag{10}$$

where $\boldsymbol{K}_{tt}$ is the covariance matrix composed between configurations in the training set and $\boldsymbol{y}$ is a set of reference data. More detailed formulations of GAP are available in previous literature.[8,19]

C. Differentiable descriptor

When the training set is fixed, the performance of a ML interatomic potential is controlled by descriptor parameters η and model parameters θ. Most of previous works only optimize model parameters in the training process, which, however, reduces the flexibility of the models. The reason for only optimizing θ is that it is difficult to compute the derivative of loss function l with respect to η. Here, we solve this problem by automatic differentiation, which has been recently used to calculate the complex gradients in quantum chemistry.[20] In view of automatic differentiation, a function is composed of a series of simple analytical functions, whose calculation procedure can be represented by a computational graph. The derivatives can be evaluated by the chain rule when the computational graph has been constructed. In this work, we built a computational graph correctly to represent the calculation process of the fingerprints and the derivatives should be computed by automatic differentiation with a ML library, such as MXNet.[21] With this differentiable descriptor, we are able to build an end-to-end ML interatomic potential. Both continuous parameters θ and η in this model could be automatically tuned in the training process as shown in Fig. 1(c). We employ the Adam algorithm,[16] which is an efficient algorithm suitable for optimization problems with large datasets and high-dimensional parameter spaces.

D. Dataset preparation

To validate the effectiveness of differentiable descriptors employed in ML interatomic potentials, we started by generating several datasets, including two kinds of Al configurations: fcc bulk and grain boundaries. All the datasets were built from *ab initio* molecular dynamics (AIMD) based on DFT. We used Perdew-Burke-Ernzerhof (PBE) functional implemented in the Vienna *ab initio* simulation (VASP) code with the plane-wave basis of 500 eV cutoff energy. For bulk fcc Al, trajectories of AIMD started with a 2 × 2 × 2 supercell of 4.16 Å lattice constant. Two trajectories used as bulk datasets have been generated by NPT AIMD calculations under zero pressure at temperatures of 1000 K and 2000 K, respectively. The AIMD simulations were performed for 1000 steps with a time step of 0.5 fs, sampling the Brillouin zone by a 2 × 2 × 2 Monkhorst-Pack k-point mesh. For grain boundaries, we use the models built by Botu *et al.*[18] as initial configurations of AIMD simulations, including $\Sigma 5(210)$ and $\Sigma 13(510)$. The trajectories were sampled in the NVT ensemble at 1000 K for 1000 steps with a time step of 0.5 fs. Only the Γ point was used for sampling the Brillouin zone with larger cells of grain boundaries.

E. Two phase approach

A natural method to calculate the melting temperature T_m is called *heat until it melts* (HUM), which is performed by heating the solid phase until melting occurs. However, for a solid, metastable solid phases can be maintained leading to the overestimation of T_m by HUM. To overcome the shortcoming, a more accurate method called *two-phase approach* (TPA) has been introduced, which is based on the simulation supercells where solid and liquid phases coexist. If such solid-liquid coexistence is maintained in the equilibrium state, T_m could be calculated by averaging temperatures in the simulation period. In TPA, one should build a supercell containing both solid and liquid phases. However, heating such two phases independently and placing them together may cause overlapping between atoms and phase boundary effect. Instead, we use a systemic process as described by Dozhdikov *et al.*[23] to build a coexisting supercell. The equilibrium state is then obtained by a NVE molecule dynamics to evaluate the melting temperature.

F. Development environment

ML interatomic potentials in this work have been implemented based on flexible and efficient ML library MXNet.[21,24] The atomic simulation environment[25] (ASE) package is used to analyze the results from the DFT results by VASP and perform molecular dynamics simulations by the ML potentials. The phonon band structures and density of states (DOS) of bulk Al were calculated using the PHONOPY[26] code. All the other calculations and analyses are performed with home-made codes.

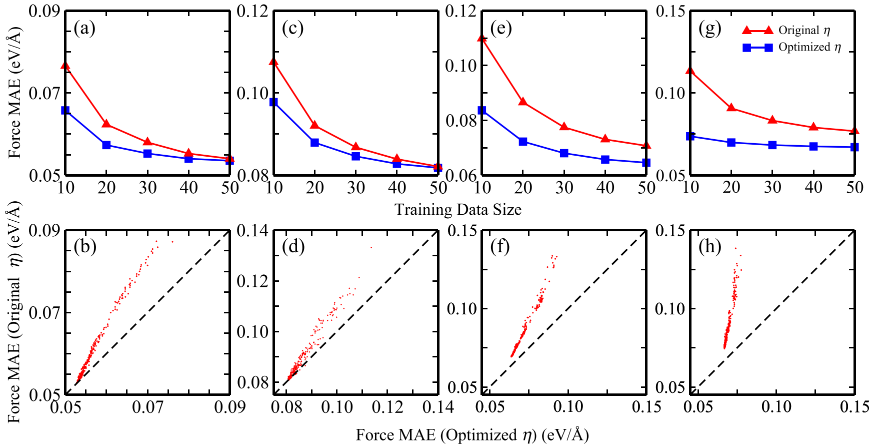

FIG. 2. Comparison of force predicted using original and optimized η for different systems: [(a) and (b)] the 1000 K bulk Al dataset, [(c) and (d)] the 2000 K bulk Al dataset, [(e) and (f)] Σ5(210), and [(g) and (h)] Σ13(510). For every training data size, 50 independent subsets were selected for training and the total MAE are shown in (a), (c), (e), and (g). The scatter figures (b), (d), (f), and (h) show the force MAE using original and optimized η for every training dataset.

III. RESULTS

A. Performances of optimized parameters

In this article, η contains 8 components and the values were sampled in a range of [ln0.1, ln2] uniformly, which is similar to the parameters used by Botu *et al.*[18] while the forms of descriptors are different. The cutoff radius is set to 8 Å. For example, we applied the differential descriptor ML potentials to model bulk and grain boundaries of Al.

For every dataset of bulk Al, 50 configurations (5% of the whole dataset) were chosen randomly to adjust the model parameter θ at first, and then, the values of η were optimized with fixed θ. Finally, we compared the accuracy of predicting forces by original and optimized η. To prove the reliability of our results, we performed 50 independent tests with randomly chosen training sets for different size i (i = 10, 20, 30, 40, 50) and the rest (1000 − i) configurations were used as the test set. The results were shown in Fig. 2. As shown in the scatter figures, almost all the points are distributed in the top-left part of the figures. It means that the forces predicted by the optimized η are always more accurate than the forces calculated from the original η. Optimization of η obviously improves the accuracy of force prediction, especially for small training datasets. When only 10 configurations are sampled in the training dataset, the optimized η reduces the mean absolute error (MAE) by 14% and 9% for 1000 K and 2000 K trajectories, respectively. For a large training dataset, the configurations for training are sufficient to represent the total trajectory in the bulk system, so the difference in MAE becomes small.

It is more difficult to build ML potentials of grain boundaries because the structures change largely in MD under 1000 K. After optimizing the values of η based on 5% of the total dataset (50 configurations), we also tested independently 50 times for different training set sizes with both original and optimized η to compare their accuracy. As shown in Fig. 2, the performance of optimized η is superior to that of the original one, no matter how many configurations are contained in the training set. In the case that the training set is composed of 10 configurations (1%), the optimized η reduces MAE by 24% and 35% for Σ5(210) and Σ13(510) datasets, respectively. There are some outliers in scatter figures with high errors predicted by original η because the randomly selected training sets are insufficient to cover the possible configurations. Applying optimized η for prediction can reduce the error significantly. To our surprise, for the complex system Σ13(510), the prediction error of optimized η based on only 10 training configurations is even lower than the error of original η with 50 training configurations. These results point out that the hand designed descriptors are not suitable for complex configurations and the optimization of descriptor parameters can effectively enhance the capability of ML potentials.

As shown in Table I, the optimized ηs are obviously different from the original ones. The distances between original and optimized ηs have relation with the performance of ML potential. The

TABLE I. The components of different η in this work.

	η_i							
Dataset	1	2	3	4	5	6	7	8
Original	−2.302 59	−1.874 62	−1.446 66	−1.018 7	−0.590 74	−0.162 78	0.265 185	0.693 147
Bulk 1000 K	−1.854 02	−1.833 6	−1.454 66	−1.158 12	0.297 696	0.303 112	0.399 569	0.433 958
Bulk 2000 K	−1.875 26	−1.870 33	−1.484 3	−1.221 97	0.269 014	0.362 559	0.363 699	0.373 096
Σ5(210)	−1.895 89	−1.839 87	−1.795 52	−1.560 35	−1.399 35	−1.286 37	−0.536 64	0.328 343
Σ13(510)	−1.917 22	−1.861 8	−1.840 83	−1.717 21	−1.526 89	−1.324 32	−1.011 62	−0.796 84

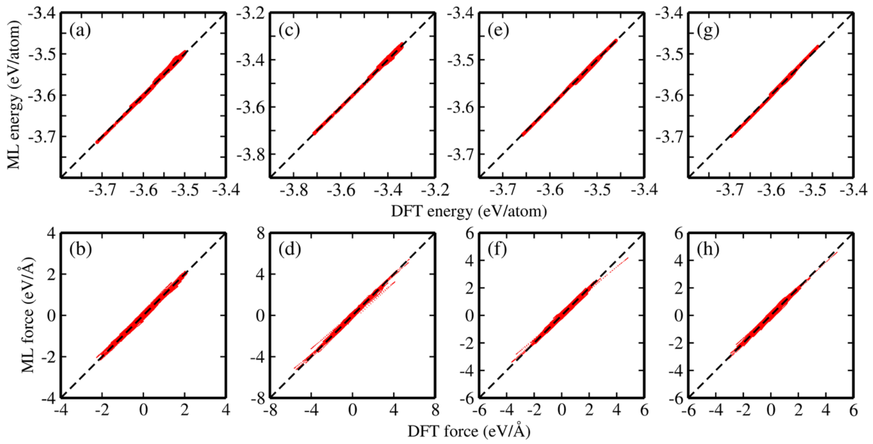

FIG. 3. Comparison of energy and force predicted by ML potentials and DFT, for (a) and (b) the 1000 K bulk Al dataset, (c) and (d) the 2000 K bulk Al dataset, (e) and (f) Σ5(210), and (g) and (h) Σ13(510). For every dataset, 50 configurations have been randomly taken as the training set and the rest 950 configurations are used as the test set.

η far away from the original one can significantly improve the accuracy of predictions on the relative dataset, such as Σ13(510). Furthermore, the optimal parameters for the datasets also reflect the relation between them. For example, the optimized ηs for the two bulk datasets are approximately similar, but the differences of η between the bulk and grain boundary datasets are comparatively large. The relation between the descriptor parameters and configurations is not surprising. Different configurations occupy different regions of the potential energy surface (PES) with various local shapes. The ML potentials can be viewed as low dimensional representations of the local PES. So different configurations lead to different descriptor parameters. To test the quality of the our ML potentials with optimized η, we have also shown the comparison of DFT and ML results in predicting energies and forces in Fig. 3. The energy MAEs of the force fields are 3.1 meV/atom, 3.8 meV/atom, 1.4 meV/atom, and 1.9 meV/atom for 1000 K bulk, 2000 K bulk, Σ5(210), and Σ13(510). The force MAEs are 0.053 eV/Å, 0.082 eV/Å, 0.064 eV/Å, and 0.067 eV/Å, respectively. The accuracy is close to the converge criteria in usual DFT relaxation calculations. These results show that our approach is indeed valid and can be used to generate accurate ML potentials, even with small training datasets.

B. Optimize with different size

In the above tests, we used 5% configurations to adjust the descriptor parameters η. Here, we would like to investigate the effect of training set size to η, so we compared the performance of three different types of η: original η, η optimized by 5% configurations (called broad fit η here for short), and η optimized by 1% configurations (called local fit η). In this test, we used two datasets: 1000 K bulk and Σ13(510) grain boundaries and the results are shown in Fig. 4. The broad fit η outperforms others in both datasets. For the bulk dataset, the performance of local fit η is better than original η and very close to broad fit η with small training data size. When the training size is large, the error of local fit η becomes higher than the original one. It means that the η is optimized to a local minimum using such a small training dataset. However, the descriptors have different performances for the grain boundary dataset. In this case, the local fit η is significantly superior to the original one in all conditions. The reason is that the original η is far from the optimal value in this system and optimization with only several configurations can effectively improve the prediction accuracy.

C. Vibrational and melting properties of aluminum

Besides simply evaluating the error of static calculations, it is more important to test the ability of the ML potential to reproduce the properties of dynamical processes. If the ML potential has good performance, we can apply it to those methods in need of simulating large cells containing thousands of or even more atoms in long time scales, which are beyond the capability of DFT due to computational cost. In the following parts, we use the trained ML potentials to evaluate the vibrational and melting properties and compare the results with DFT calculations.

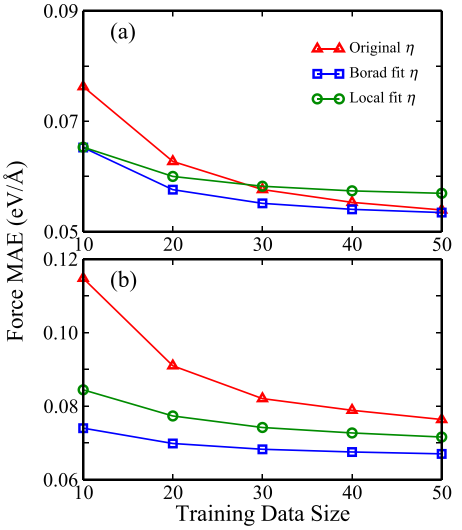

FIG. 4. MAE of force predictions using different η in (a) 1000 K bulk dataset and (b) Σ13(510) grain boundary dataset.

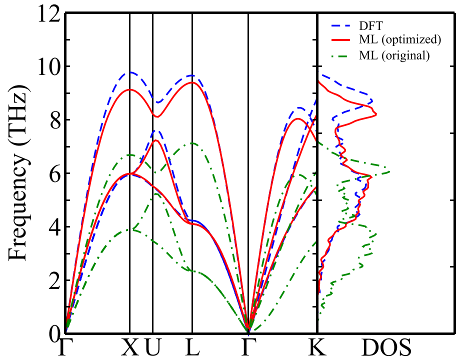

FIG. 5. Phonon band structure and density of states (DOS) of fcc Al using DFT code (blue dashed lines), optimized (red solid lines), and origin (green dashed lines) ML interatomic potentials.

Figure 5 shows the phonon dispersion curves and density of states (DOS) of fcc Al calculated by ML potential and VASP. The ML potential is trained with optimized η using 50 configurations randomly chosen from the 1000 K trajectory. The ML potential and DFT are in good agreement with vibrational properties at zero temperature although a slight difference exists in the high frequency region. The results show that the ML potential can describe the vibrational behavior of materials correctly. In addition, we have also calculated the phonon properties by ML potentials with original η. The original potential dramatically underestimates the frequencies, which reveals the importance of descriptor optimization.

The simulations of liquid phase often require large cells which are demanding for DFT calculations. Here, we computed the melting properties of Al using a trained ML potential and compared the results with *ab initio* and experimental data. To model the interactions in the melting process of Al, we generated a NPT trajectory at 3000 K for 2 × 2 × 2 fcc Al, and other parameters were the same to 1000 K and 2000 K simulations. A training set containing 50 configurations randomly sampled from 2000 K to 3000 K trajectories has been used to construct a ML interatomic potential with optimization of the descriptor parameter η.

First, the radial distribution function (RDF) of liquid Al at 1000 K has been simulated by ML potentials (original and optimized) and DFT. The simulation cell for the ML potential is a 10 × 10 × 10 supercell containing 4000 atoms, while for DFT, the supercell is 4 × 4 × 4 with 256 atoms. The results are shown in Fig. 6(d) and compared with the experimental data at 1023 K.[27] The optimized ML potential is successful in reproducing the DFT result, including the heights and widths of peaks, and as expected, its performance is better than that of the original one. Both DFT and optimized ML RDF curves are in good agreement with the experimental data.[27]

Next, a much more demanding task, melting temperature calculation by a two phase approach (TPA), has been performed by the ML potential to test the ability to simulate the two-phase system. We produced a 4 × 4 × 10 coexisting supercell containing 640 Al atoms by several short MD simulations at first. To evaluate the melting temperature of Al, we conducted an NVE MD for 60 ps with a time step of 3 fs. The curves of the kinetic energy, the potential energy, and the total energy of the coexisting system are shown in Fig. 6(e). The total energy keeps stable in such long simulation due to the energy conservation of our ML potential. We confirmed that the system has reached thermal equilibrium and the coexistence of solid and liquid phases has never been broken by comparing the starting and end configurations. To verify the state of coexistence further, the density profile and the mean square displacement (MSD) of both solid and liquid phases have been computed and displayed in Figs. 6[(a)–(c)]. The density in the solid area shows periodic

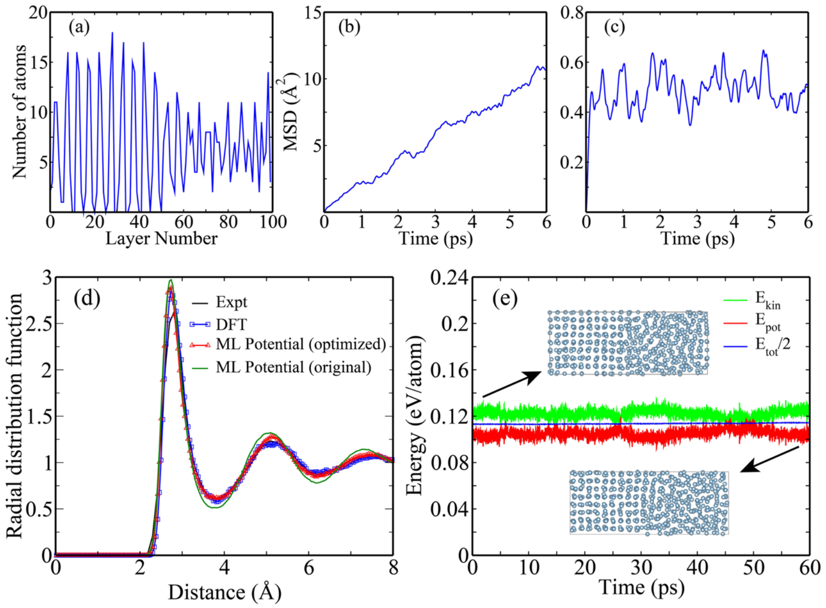

FIG. 6. (a) Density profile in a simulation cell where solid and liquid Al coexist. The system is divided by 100 layers parallel to the solid-liquid interface, and the graph shows the number of atoms in each layer. (b) Mean square displacement of a group of Al atoms in the liquid state. (c) Mean square displacement of a group of Al atoms in the solid state. (d) Calculated and experimental radial distribution function of Al at about 1000 K. (e) The kinetic energy, the potential energy, and the total energy of the coexisting supercell in the NVE simulation. The inset shows the beginning and last configurations.

oscillation vs the layer number, while in the liquid phase, the amplitude becomes much lower and the periodicity is lost. The MSD of the liquid phase shows linear dependence on time and solid MSD keep stable with small fluctuations as widely known. The NVE simulation of the coexisting supercell results in an average temperature of 944 K under 0.9 kbar. The temperature is very close to the *ab initio* result 950 K[28] with a difference of 0.6% which is even lower than the variance from different sizes of the simulation cell by DFT.[28] Furthermore, our result is also in excellent agreement with experimental data of 933 K.

D. Energy prediction

In this article, we focus on the prediction of forces by ML potentials because the MD trajectory of an atomic system is not dependent on the total energy explicitly and computing the forces solely is enough to simulate many dynamical processes. That is the reason why those energy-free ML force fields[18,29–31] without calculating total energy are worth to study. The calculation of an N-atom structure only provides one value of energy but $3N$ force components so the proportion of forces in loss function is much larger than energy. In some cases, the decrease in force error may cause an increase in energy error. We have made energy prediction using 5% configurations with original and optimized η. The prediction error of optimized η is almost similar to that of the original one.

IV. CONCLUSIONS

In summary, we have developed a method to build differentiable descriptors by an automatic differentiation technique so the optimization of descriptor parameters becomes accessible. The optimization can effectively improve the performance of ML interatomic potentials in different systems such as bulk and grain boundaries. The trained potential models show the ability to reproduce the properties computed by DFT including phonon curves, liquid behaviors, and melting point. Our results show that the hand designed descriptors sometimes are not suitable for complex configurations, thus optimizing the descriptor parameters becomes necessary and can effectively enhance the capability of ML potentials. Most importantly, we find that the ML potentials from our method have better performance compared to the original ones when the training dataset is small. This makes our model useful in the cases when a big training dataset is not available, such as crystal structure search with *ab initio* calculations. A nice example has been shown recently by Xia *et al.*[32] to predict a superhard tungsten nitride using the ML accelerated crystal structure search method. Furthermore, the application of our strategy is not limited to GPR-based potentials used in this article. For instance, those NN-based potentials using Behler-Parrinello SFs can also be enhanced by our method.

ACKNOWLEDGMENTS

We gratefully thank Gábor Csányi at the University of Cambridge for fruitful discussions. We acknowledge the financial support from MOST of China (Grant Nos. 2016YFA0300404 and 2015CB921202), the National Natural Science Foundation of China (Grant Nos. 11574133 and 11834006), the NSF Jiangsu province in China (No. BK20150012), the Science Challenge Project (No. TZ2016001), Calculations were performed on supercomputers at the High Performance Computing Center of Collaborative Innovation Center of Advanced Microstructures, at the high-performance supercomputing center of Nanjing University and "Tianhe-2" at NSCC-Guangzhou.

REFERENCES

[1] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[2] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Phys. Rev. Lett. **104**, 136403 (2010).
[3] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, J. Comput. Phys. **285**, 316 (2015).
[4] A. Shapeev, Multiscale Model. Simul. **14**, 1153 (2016).
[5] A. Seko, A. Takahashi, and I. Tanaka, Phys. Rev. B **92**, 054113 (2015).
[6] J. Behler, Int. J. Quantum Chem. **115**, 1032 (2015).
[7] J. Behler, Angew. Chem., Int. Ed. **56**, 12828 (2017).
[8] A. P. Bartók and G. Csányi, Int. J. Quantum Chem. **115**, 1051 (2015).
[9] H. Zong, G. Pilania, X. Ding, G. J. Ackland, and T. Lookman, Npj Comput. Mater. **4**, 48 (2018).
[10] J. Behler, J. Chem. Phys. **134**, 074106 (2011).
[11] A. P. Bartók, R. Kondor, and G. Csányi, Phys. Rev. B **87**, 184115 (2013).
[12] A. Khorshidi and A. A. Peterson, Comput. Phys. Commun. **207**, 310 (2016).
[13] T. D. Huan, R. Batra, J. Chapman, S. Krishnan, L. Chen, and R. Ramprasad, Npj Comput. Mater. **3**, 37 (2017).
[14] G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, J. Chem. Phys. **148**, 241730 (2018).
[15] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, J. Chem. Phys. **148**, 241709 (2018).
[16] D. P. Kingma and J. Ba, in *International Conference for Learning Representations* (2015).
[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, Mass, 2006).
[18] V. Botu, R. Batra, J. Chapman, and R. Ramprasad, J. Phys. Chem. C **121**, 511 (2017).
[19] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, Sci. Adv. **3**, e1701816 (2017).
[20] T. Tamayo-Mendoza, C. Kreisbeck, R. Lindh, and A. Aspuru-Guzik, ACS Cent. Sci. **4**, 559 (2018).
[21] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, in *Neural Information Processing Systems* (2016).
[22] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).
[23] V. S. Dozhdikov, A. Yu. Basharin, and P. R. Levashov, J. Chem. Phys. **137**, 054502 (2012).
[24] M. Seeger, A. Hetzel, Z. Dai, and N. D. Lawrence, e-print arXiv:1710.08717 [Cs, Stat] (2017).
[25] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, J. Phys.: Condens. Matter **29**, 273002 (2017).
[26] A. Togo and I. Tanaka, Scr. Mater. **108**, 1 (2015).
[27] Numerical values of the structure factor, g(r), for liquid metals at various temperatures, available at: http://res.tagen.tohoku.ac.jp/~waseda/scm/LIQ/gr.html.
[28] J. Bouchet, F. Bottin, G. Jomard, and G. Zérah, Phys. Rev. B **80**, 094102 (2009).
[29] Z. Li, J. R. Kermode, and A. De Vita, Phys. Rev. Lett. **114**, 096405 (2015).
[30] A. Glielmo, P. Sollich, and A. De Vita, Phys. Rev. B **95**, 214302 (2017).
[31] I. Kruglov, O. Sergeev, A. Yanilkin, and A. R. Oganov, Sci. Rep. **7**, 8512 (2017).
[32] K. Xia, H. Gao, C. Liu, J. Yuan, J. Sun, H.-T. Wang, and D. Xing, Sci. Bull. **63**, 817 (2018).