Machine-learned embedded atom method

Xin Chen, Li-Fang Wang, Xing-Yu Gao, Ya-Fan Zhao, De-Ye Lin*, Wei-Dong Chu, Hai-Feng Song*

Institute of Applied Physics and Computational Math, Beijing 100088, China and
CAEP Software Center for High Performance Numerical Simulation, Beijing 100088, China

Interatomic potential or force-field plays a vital role in simulating materials. Embedded atom method (EAM) and its variant angular-dependent potential (ADP) have proven successful in modeling fcc and bcc metals qualitatively. In the past few years, machine learning has became the central approach for modeling interatomic interactions. Powered by big data and efficient optimizers, machine learning interatomic potentials can generally approximate first-principle accuracy. In this works, we successfully integrated machine learning with EAM and ADP. Machine learning can significantly boost their performances. For energy and force predictions, machine leaned EAM and ADP can be almost as accurate as the spectral neighbor analysis potential (SNAP) on the fcc Ni, bcc Mo and Mo-Ni alloy systems, while the later approach is nealy 1000 times slower. ML-EAM and ML-ADP can also give first-principle accuracy key material properties, such as elastic constants, melting temperatures. Our results suggest a new and systematic route for developing machine learning interatomic potentials. Our

I. INTRODUCTION

Atomistic modeling plays a vital role in materials science. *ab initio* calculation or force-field based molecular dynamics simulation (MD) are effective ways to study, understand or predict chemical and physical properties of materials. *ab initio* approaches are generally much more precise but they are rarely used on large-scale metallic systems due to their extremely-high computation expenses. Physical model based empirical potential (force-field), such as the embedded-atom method (EAM) [1–5], modified embedded-atom method (MEAM) [6–9], bond-order potential (BoP) [10–13], or angular-dependent potential (ADP) [14–18], still plays the major role in long-time simulations and these empirical methods can achieve reasonable accuracy with much lower computation costs. Empirical potentials generally have very few learnable parameters and both microscopic observables (energy, forces, virial, etc.) and macroscopic observables (melting point, surface energy, etc.) can be used to tune these parameters. Finding optimal parameters of empirical potentials is always a challenging task. Global optimization (GO) approaches (Basin-Hopping [19, 20], genetic algorithm [21, 22], etc) are traditionally used to find the best possible parameters. However, the gradients of the losses with respect to model parameters are difficult or even impossible to calcualte. Hence, GO optimizations are generally not that effective.

In the last decade, machine learning (ML) has become one of the hottest topics in many research areas. In the materials science, researchers have made great efforts on developing ML models to describe atomic interactions. Such ML models are considered as machine learning interaction potentials (MLIPs). Until now, hundreds of MLIPs have been proposed. Among them, the symmetry-function based atomistic neural network (ANN) model, published by Parinello and Behler in 2007 [23–27], is still the most popular choice in modeling metallic interactions [28–30]. The smooth overlap atomic positions descriptor based gaussian approximation potential (SOAP-GAP) [31–34], developed by Bartòk et al, can give extremely accurate prediction results, although it's a bit computationally expensive. Recently, Thompson and co-workers proposed another quantum-accurate MLIP named the spectral neighbor analysis potential (SNAP) [35] and it has been proven working on a broad range of metals and alloys [36–38].

In many cases, MILPs can easily outperform state-of-art empirical potentials. Compared with empirical potentials, MLIPs generally have orders of magnitudes more model paramters. The redundant parameter space greatly reduces the difficulty of fitting complicated potential energy surfaces. But, to effectively train a MLIP and avoid overfitting, a large high-quality (versatile) training dataset is probably needed. However, MILPs **can really** take advantages of "big data" for two reasons. First, MLIPs typically only have *basic* or *simple* arithmetic operations. Thus, MILPs can be implemented within modern deep learning frameworks (TensorFlow [39], PyTorch [40], etc) so that the gradients of the total loss with respect to fitting parameters can be can obtained with the backpropagation algorithm automatically and efficiently. Second, MLIPs are mostly vectorizable. Hence, GPUs can be utilized to significantly accelerate training and using of MILPs.

However, MLIPs also have challenges. The large parameter space and lack of physical background makes the "big data" a necessity. The cost of dataset is non-negligible. Besides, even "big data" can only cover a small portion of real physical environments (temperature, external pressure, etc). Outside the training zone, the performances of MILPs may not that stable. For long-time molecular dynamics (MD) simulations of large-scale (10^5 or more) systems, computation efficiency also becomes a major concern. Recent benchmark tests [41] suggest that MILPs are still too expensive. At present, most MLIPs are used to examine small to medium (10^3 to 10^4) systems [28–30, 34, 37].

In this work, instead of designing new atomic descriptors, we chose a new route to develop MLIP: combining

machine learning with empirical potentials. We successfully implemented EAM and its variant ADP within TensorFlow so that machine learning approaches can be used directly to tune EAM and ADP potentials. Our results suggest ML-EAM or ML-ADP can be as precise as the SNAP machine learning method.

This paper is organized as follows. Section II describes the theoretical background of this work, including the formalism of the embedded atom method and algorithms of the machine learned EAM. Section III summarizes the results of ML-EAM and ML-ADP. Further discussions are given in section IV.

II. METHOD

A. Theory

In the original EAM formalism [1], the total energy, E^{total}, is the sum of atomic energies:

$$
\begin{aligned}
E^{total} &= \sum_i^N E_i \\
&= \sum_i^N F_a(\rho_i) + \frac{1}{2} \sum_i^N \sum_{j \neq i}^{r_{ij} < r_c} \phi_{ab}(r_{ij})
\end{aligned} \tag{1}
$$

where r_c is the cutoff radius, a and b represents species of atoms i and j, $\phi_{ab}(r_{ij})$ is energy of the pairwise interaction between i and j, $F_a(\rho_i)$ is the embedding energy and ρ_i is the local electron density of atom i. ρ_i can be calculated with the following equation:

$$
\rho_i = \sum_j^{r_{ij} < r_c} \rho_b(r_{ij}) \tag{2}
$$

where ρ_b is the electron density function of specie b. F, ρ and ϕ can be either parameterized functions or cubic splines.

The original EAM does not inlcude angular-dependent interactions. To fix this problem, Baskes modified the original EAM and got MEAM (modified embedded-atom method) [6], Lenosky proposed an alternative spline-based interpretion of MEAM [42] while Mishin developed the angular-dependent potential (ADP) [14–17, 43]. The ADP formalism introduces three additional terms to the total energy:

$$
\begin{aligned}
E^{total} = E^{\text{EAM}} \\
+ \frac{1}{2} \sum_i \sum_\alpha (\mu_i^\alpha)^2 \\
+ \frac{1}{2} \sum_i \sum_\alpha \sum_\beta (\lambda_i^{\alpha\beta})^2 \\
- \frac{1}{6} \sum_i \nu_i^2
\end{aligned} \tag{3}
$$

These terms represent non-central bonding contributions and they can be computed with the following equations:

$$
\mu_i^\alpha = \sum_{j \neq i} \mu_{ab}(r_{ij}) r_{ij}^\alpha \tag{4}
$$

$$
\lambda_i^{\alpha\beta} = \sum_{j \neq i} \omega_{ab}(r_{ij}) r_{ij}^\alpha r_{ij}^\beta \tag{5}
$$

$$
\nu_i = \sum_\alpha \lambda_i^{\alpha\alpha} \tag{6}
$$

where $\mu_{ab}(r)$ and $\omega_{ab}(r)$ can be viewed as measures of the strengths of dipole and quadrupole interactions.

B. Transformation

To integrate EAM/ADP with machine learning, the original total energy expression (Equation 1) must be transformed to a vectorizable form. Without loss of generality, we take the binary alloy, AB, to demonstrate how to do the transformation.

Suppose the cutoff radius r_c is fixed, the energy of atom i of specie A can be calculated with the following expanded equation:

$$
\begin{aligned}
E_i^A = &\frac{1}{2} \sum_{j \neq i}^{N_i^{\text{AA}}} \phi_{\text{AA}}(r_{ij}) + \frac{1}{2} \sum_{j \neq i}^{N_i^{\text{AB}}} \phi_{\text{AB}}(r_{ij}) \\
&+ F_{\text{A}} \left(\sum_{j \neq i}^{N_i^{\text{AA}}} \rho_{\text{A}}(r_{ij}) + \sum_{j \neq i}^{N_i^{\text{AB}}} \rho_{\text{B}}(r_{ij}) \right)
\end{aligned} \tag{7}
$$

where N_i^{AA} represents the number of A-type neighbors of atom i and N_i^{AB} represents the number of B-type neighbors. For atom j of specie B, we can also write a similar form.

$$
\begin{aligned}
E_j^B = &\frac{1}{2} \sum_{i \neq j}^{N_j^{\text{BB}}} \phi_{\text{BB}}(r_{ij}) + \frac{1}{2} \sum_{i \neq j}^{N_j^{\text{BA}}} \phi_{\text{AB}}(r_{ij}) \\
&+ F_{\text{B}} \left(\sum_{j \neq i}^{N_j^{\text{BB}}} \rho_{\text{B}}(r_{ij}) + \sum_{j \neq i}^{N_j^{\text{BA}}} \rho_{\text{A}}(r_{ij}) \right)
\end{aligned} \tag{8}
$$

When r_c is fixed, N_i^{AA}, N_i^{AB}, N_j^{BB} and N_j^{BA} are all constants and N_i^{nl} is the maximum of these numbers. Finally, we can pre-determine the maximum neighor list size N^{nl}:

$$
N^{\text{nl}} = \max \left(N_i^{\text{nl}} \right) \tag{9}
$$

N^{nl} is also a constant in the training phase because both the training dataset and r_c are fixed.

Next, assume $H(x)$ represents the heaviside step function:

$$
H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{10}
$$

then the pairwise term can be transformed to:

$$
\sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) = \sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) \cdot 1 + \sum_{j\neq i}^{N^{\mathrm{nl}}-N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(0) \cdot 0
$$
$$
= \phi_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) \tag{11}
$$

where $\vec{\mathbf{r}}_i^{\mathrm{AA}}$ is a N^{nl}-length column vector whose last $N^{\mathrm{nl}}-N_i^{\mathrm{AA}}$ elements are zeros.

We can write Equation 8 in an equivalent expression:

$$
E_i^{\mathrm{A}} = \frac{1}{2}\left(\phi_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) + \phi_{\mathrm{AB}}(\vec{\mathbf{r}}_i^{\mathrm{AB}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AB}})\right)
$$
$$
+ F_{\mathrm{A}}\left(\rho_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) + \rho_{\mathrm{AB}}(\vec{\mathbf{r}}_i^{\mathrm{AB}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AB}})\right) \tag{12}
$$

Here $\vec{\mathbf{r}}_i^{\mathrm{AB}}$ is also a N^{nl}-length vector. For atom j of specie B, we can also derive its energy E_j^{B}:

$$
E_j^{\mathrm{B}} = \frac{1}{2}\left(\phi_{\mathrm{BB}}(\vec{\mathbf{r}}_j^{\mathrm{BB}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BB}}) + \phi_{\mathrm{BA}}(\vec{\mathbf{r}}_j^{\mathrm{BA}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BA}})\right)
$$
$$
+ F_{\mathrm{A}}\left(\rho_{\mathrm{BB}}(\vec{\mathbf{r}}_j^{\mathrm{BB}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BB}}) + \rho_{\mathrm{BA}}(\vec{\mathbf{r}}_j^{\mathrm{BA}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BA}})\right) \tag{13}
$$

Since $\vec{\mathbf{r}}_i^{\mathrm{AA}}$, $\vec{\mathbf{r}}_i^{\mathrm{AB}}$, $\vec{\mathbf{r}}_i^{\mathrm{BB}}$ and $\vec{\mathbf{r}}_i^{\mathrm{BA}}$ all have the same length (N^{nl}), we can use a (redundant) matrix, $\mathbf{g}_i$, to describe all neighbors of atom i:

$$
\mathbf{g}_i = \begin{bmatrix} \vec{\mathbf{r}}_i^{\mathrm{AA}} & \vec{\mathbf{r}}_i^{\mathrm{AB}} & \vec{\mathbf{r}}_i^{\mathrm{BB}} & \vec{\mathbf{r}}_i^{\mathrm{BA}} \end{bmatrix} \tag{14}
$$

$\mathbf{g}_i$ is a $N^{\mathrm{nl}} \times 4$ matrix. If the specie of atom i is A, only the first two columns have non-zero values. Similarly, the last two columns will have non-zeros values if atom i is a B-type atom. In fact, $\mathbf{g}_i$ can be viewed as the EAM descriptors for atom i. Hence, each structure can be expressed with a 3D matrix, $\mathbf{G}$, of shape $N \times N^{\mathrm{nl}} \times 4$.

During the training phase, the maximum appearances of element A and B in any structure ($N_{\mathrm{A}}^{\max}$ and $N_{\mathrm{B}}^{\max}$) are also constants. Thus, any $\mathbf{G}$ can be expanded to a $(N_{\mathrm{A}}^{\max} + N_{\mathrm{B}}^{\max}) \times N^{\mathrm{nl}} \times 4$ matrix $\mathbf{G}'$ by zero-padding. In summary, arbitrary structure in the training dataset can be converted to a fixed-shape descriptor matrix $\mathbf{G}'$.

For the ADP formalism, the corresponding transformation is almost the same except that $\mathbf{g}_i^{\mathrm{adp}}$ should also include the XYZ components of $\vec{\mathbf{r}}_i$.

C. Functions

In this work, we use the EAM potential published by Zhou, Johnson and Wadley (Zjw04) [44–46] as an example to validate our machine learning approach. Zjw04 is a quite popular EAM potential. In the Zjw04 potential, the electron density function has the following form:

$$
\rho_b(r) = \frac{f_e \exp\left[-\beta\left(r/r_e - 1\right)\right]}{1 + (r/r_e - \lambda)^{20}} \tag{15}
$$

where r_e is a constant equal to equilibrium spacing between nearest neighbors, f_e, β, λ are adjustable parameters. The pairwise potential between the same species can be computed with:

$$
\phi_{aa}(r) = \frac{A\exp\left[-\alpha\left(r/r_e - 1\right)\right]}{1 + (r/r_e - \kappa)^{20}} - \frac{B\exp\left[-\beta\left(r/r_e - 1\right)\right]}{1 + (r/r_e - \lambda)^{20}} \tag{16}
$$

where A, B, α and κ are also trainable parameters, β and κ are used in Equation 15 already. For the pairwise interaction between two atoms of different species, Zhou et al proposed an interpolation form:

$$
\phi_{ab}(r) = \frac{1}{2}\left(\frac{\rho_b(r)}{\rho_a(r)}\phi_{aa}(r) + \frac{\rho_a(r)}{\rho_b(r)}\phi_{bb}(r)\right) \tag{17}
$$

The embedding function has a more complicated form as it requires to fit a much wider range of electron density values:

$$
F(\rho) = \begin{cases} \sum_{i=0}^{3} F_{ni}\left(\frac{\rho}{\rho_n} - 1\right)^i & \rho < \rho_n \\ \sum_{i=0}^{3} F_i\left(\frac{\rho}{\rho_e} - 1\right)^i & \rho_n \le \rho < \rho_0 \\ F_e\left[1 - \eta\ln\left(\frac{\rho}{\rho_s}\right)\right]\left(\frac{\rho}{\rho_s}\right)^{\eta} & \rho_0 \le \rho \end{cases} \tag{18}
$$

where F_{ni}, F_i, ρ_e, ρ_s, η and F_e are trainable parameters, $\rho_n = 0.85\rho_e$ and $\rho_0 = 1.15\rho_e$. For each metal, there are 15 adjustable parameters. The original embedding potential is a stepwise function. Thus, the minimization requires some tricks to ensure its continuity. To make it simpler, we slightly modified Equation 18:

$$
F(\rho) = c_1 \cdot \sum_{i=0}^{3} F_{ni}\left(\frac{\rho}{\rho_n} - 1\right)^i
$$
$$
+ c_2 \cdot \sum_{i=0}^{3} F_i\left(\frac{\rho}{\rho_e} - 1\right)^i
$$
$$
+ c_3 \cdot F_e\left[1 - \eta\ln\left(\frac{\rho}{\rho_s}\right)\right]\left(\frac{\rho}{\rho_s}\right)^{\eta} \tag{19}
$$

$$
c_1 = \frac{1}{1 + e^{-2(\rho_n - \rho)}} \tag{20}
$$

$$
c_3 = \frac{1}{1 + e^{-2(\rho - \rho_0)}} \tag{21}
$$

$$
c_2 = 1 - c_1 - c_3 \tag{22}
$$

ω_1 and ω_3 are just damping factors calculated by the sigmoid functions (Equations 20 and 21).

The dipole (μ_{ab}) and quadrupole (λ_{ab}) functions have the same form (developed by Mishin [14]):

$$
\mu_{ab}(r) = \left[d_1^{ab}\exp\left(-d_2^{ab}r\right) + d_3^{ab}\right]\psi\left(\frac{r - r_0}{r_h}\right) \tag{23}
$$

$$
\omega_{ab}(r) = \left[q_1^{ab}\exp\left(-q_2^{ab}r\right) + q_3^{ab}\right]\psi\left(\frac{r - r_0}{r_h}\right) \tag{24}
$$

where d_i, q_i, r_0 and r_h are trainable parameters and $\psi(x)$ is a damping function:

$$
\psi(x) = \begin{cases} 0 & x \ge 0 \\ \frac{x^4}{1+x^4} & x < 0 \end{cases} \tag{25}
$$

D. Physical constraints

Physical constraints are quite common in fitting traditional empirical potentials. Physical constraints are typically static (cohesive energy, elastic constants, etc) collected from experiments and they can be very effective when training data is limited. For example, the cohesive energy, bulk modulus, vacancy formation energy and other constraints were used to develop the original Zjw04 potential. Mishin et al adopted the Rose universal equation of state to ensure the performaces of his ADP potentials in the high pressure region. However, such constraints are really rare in developing MILPs. One possible explanation may be these constraints are generally derived properties and implementing them in the loss function are technically difficult.

In this work, we successfully integrated two constraints into the total loss: the Rose equation of state (EOS) [47–49] constraint and the elastic tensor constraint. These two losses will be discussed later. The details of their implementations will be described in another paper.

The Rose constraint incoporates the universal equation of state (Rose et al) into the total loss function. The Mishin-modified equation

$$E(x) = E_0 \left[1 + \alpha x + \beta \alpha^3 x^3 \frac{2x+3}{(x-1)^2} \right] e^{-\alpha x} \qquad (26)$$

is used because the original form tends to underestimate energies under high pressures. In Equation 26, E_0 is the energy of the equilibrium structure, $x = a/a_0 - 1$ is the relative isotropic scaling factor (a is the lattice constant), β is a chosen parameter and

$$\alpha = \sqrt{-\frac{9V_0 B}{E_0}} \qquad (27)$$

where V_0 is the equilibrium volume and B is the bulk modulus. The adoption of the Rose constraint guarantees the exact predictions of bulk modulus and the energy-volume curve. The loss of the Rose EOS constraint $\mathbf{L}^{\text{Rose}}$ is measured as the 2-norm of the energy differences between $E(x)^{\text{dft}}$ and their corresponding predicted $E(x)$:

$$\mathbf{L}^{\text{Rose}} = \sum_{\Omega} \sqrt{\sum_i \left(E(x_i) - E(x_i)^{\text{dft}} \right)^2} \qquad (28)$$

Here Ω denotes all selected crystals. In this work, for each included crystal, we use the same choices of x: $x_i = x_0 + \Delta x \cdot i, x_0 = -0.1, N_t^{\max} = 20, \Delta x = 0.01$. In this work, β^{Ni} is 0.005 and β^{Mo} is 0.150.

Elastic tensor is also a popular constraint for tuning empirical potentials but rarely used directly in optimizing MILPs. Shyue Ping Ong used this constraint in the outer loop (the global optimization based property-matching step) to find optimal parameters of SNAP potentials.

In this work, we successfully find a way to use elastic tensor directly as a constraint. Given an equilibrium crystal structure, its elastic constant c_{ijkl} can be derived from E^{total} directly [50]:

$$V \cdot \epsilon = -\mathbf{F}^T \mathbf{R} + \left(\frac{\partial E^{total}}{\partial \mathbf{h}} \right)^T \mathbf{h} \qquad (29)$$

$$c_{ijkl}|_{\epsilon \to 0, \mathbf{F} \to 0} = \frac{1}{V} \left[\left(\frac{\partial \epsilon_{ij}}{\partial \mathbf{h}} \right)^{\text{T}} \mathbf{h} \right]_{kl} \qquad (30)$$

where V is the volume, ϵ is the 3×3 virial stress tensor, $\mathbf{h}$ is the *row-major* 3×3 lattice tensor, $\mathbf{F}$ and $\mathbf{R}$ are $N \times 3$ matrices representing the total forces and atomic positions.

In this work, the loss $\mathbf{L}^{\text{elastic}}$ contributed by the elastic tensor is also measured by the RMSE between c_{ijkl} and c_{ijkl}^{dft}:

$$\mathbf{L}^{\text{elastic}} = \sum_{\Omega} \left(\omega_{\Omega} \cdot \mathbf{RMSE}_{\Omega} + ||\epsilon|| + ||\mathbf{F}|| \right) \qquad (31)$$

$$\omega_{\Omega} = \mathbf{ReLU}(\mathbf{MAE}_{\Omega} - \tau) \qquad (32)$$

$$\mathbf{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \qquad (33)$$

where $\sum_{\Omega}$ loops through all included crystals, $\mathbf{MAE}_{\Omega}$ and $\mathbf{RMSE}_{\Omega}$ are the mean absolute error and root mean squared error between predicted elastic constants and DFT elastic constants. τ is a pre-selected gate parameter. When $\mathbf{MAE}$ is below τ, $\mathbf{L}^{\text{elastic}}$ will not contribute to the total loss. In this work, τ is set to 2 GPa.

E. Implementation

The implementation of the machine learned embedded atom mathod and the physical constraints are beyond the scope of this work. The details will be discussed in another paper. Here, we will give a brief description.

Both ML-EAM and ML-ADP are implemented within Google's TensorFlow. The virtual-atom approach (VAP) is adopted so that we can construct a computation graph from atomic positions to total energy directly. Thus, atomic forces (Equation 34) and virial stress (Equation 29) can be derived by the AutoGrad module of TensorFlow and calcualted automatically and efficiently.

$$\mathbf{F} = -\frac{\partial E^{total}}{\partial \mathbf{R}} \qquad (34)$$

This direct computation graph also plays a key role in computing analytical elastic tensor (Equation 30) and the Rose loss (Equation 26).

To find optimal parameters of the funcitons in section II C, the overall loss function shall be defined. Just like other machine learning tasks, the mini-batch stochastic gradient descent algorithm is used to minimize the loss function. Equation 35 demonstrates the loss function
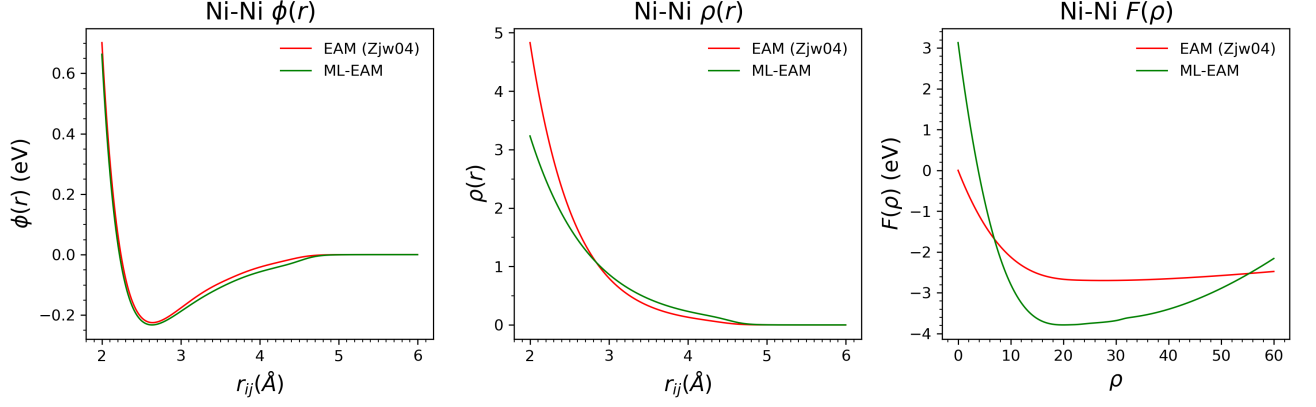
FIG. 1. $\rho(r)$, $F(\rho)$ and $\phi(r)$ of the original Zjw04 EAM and the machine learned EAM (ML-EAM).

	DFT	SNAP [37]	EAM	ML-EAM	Exp.
T_m(K)		1785	1520 [46]	1520	1728
c_{11} (GPa)	276	276 (0.0%)	248 (-10.1%)	274 (-0.7%)	261 [51]
c_{12} (GPa)	159	159 (0.0%)	147 (-7.5%)	163 (2.5%)	151 [51]
c_{44} (GPa)	132	132 (0.0%)	125 (-5.3%)	131 (-0.8%)	132 [51]
B_{VRH} (GPa)	198	198 (0.0%)	181 (-8.6%)	195 (0.0%)	188
G_{VRH} (GPa)	95	95 (0.0%)	87 (-8.4%)	93 (-2.1%)	93
μ	0.29	0.29 (0.0%)	0.29 (0.0%)	0.29 (0.0%)	0.29
E_v (eV)	1.46	1.68 (15.1%)	1.68 (15.1%)	1.71 (17.1%)	1.54–1.80 [52]
E_m (eV)	1.12	1.07 (-4.5%)	0.90 (-19.6%)	0.87 (-22.3%)	1.01–1.48 [52]
$E_a = E_v + E_m$ (eV)	2.58	2.75 (6.6%)	2.58 (0.0%)	2.58 (0.0%)	2.77–2.95 [52]

TABLE I.   Comparison of the calculated and experimental material properties of elementary fcc Ni. T_m is the melting point. c_{ij} represents elastic constants. E_v and E_m are vacancy formation energy and migration energy, respectively. SNAP refers to the elementary Ni SNAP model.

used in this work:

$$\mathbf{Loss} = \sqrt{\frac{1}{N_b} \sum_{i=1}^{N_b} \left(E_i - E_i^{\mathrm{dft}} \right)^2}$$
$$+ \chi_{\mathrm{f}} \sqrt{\frac{1}{3 \sum_i^{N_b} N_i} \sum_i^{N_b} \sum_j^{N_i} \sum_\alpha \left(f_{ij\alpha} - f_{ij\alpha}^{\mathrm{dft}} \right)^2}$$
$$+ \chi_{\mathrm{rose}} \mathbf{L}^{\mathrm{Rose}} + \chi_{\mathrm{elastic}} \mathbf{L}^{\mathrm{elastic}} \tag{35}$$

where N_b is the batch size and N_i is the number of atoms in structure i. χ_{f}, χ_{rose} and χ_{elastic} are overall weights of force, rose EOS and elastic contributions. The Adam optimizer [53] is used to minimize Equation 35. In most cases, we use 0.01 as the initial learning rate and the batch size ranges from 20 to 50. Since there are very few adjustable parameters, the optimization typically needs very few epochs to converge. When the optimization is finished, the corresponding Lammps setfl potential file will be exported.

III.   RESULTS

The publicly Ni-Mo dataset [37] is used to evaluate our approach. This dataset is provided by Shyue Ping Ong. It contains 3973 different Ni-Mo solids. All calculations were done by VASP [54] with the PBE [55] functional and the projector augmented-wave approach [56].

The optimized parameters are listed in the appendix.

A.   Elementary Ni and Mo

We first use the two elementary datasets (Ni and Mo) to show our approach. These two datasets has 461 and 284 structures, respectively. 61 and 34 were randomly selected to form the test subsets. In these two experiments, χ_{f}, χ_{rose} and χ_{elastic} were set to 1, 3 and 0.05 respectively. The batch size was fixed to 25.

The original Zjw04 potentials were fitted to experimentally obtained quantities. Thus, we use the relative
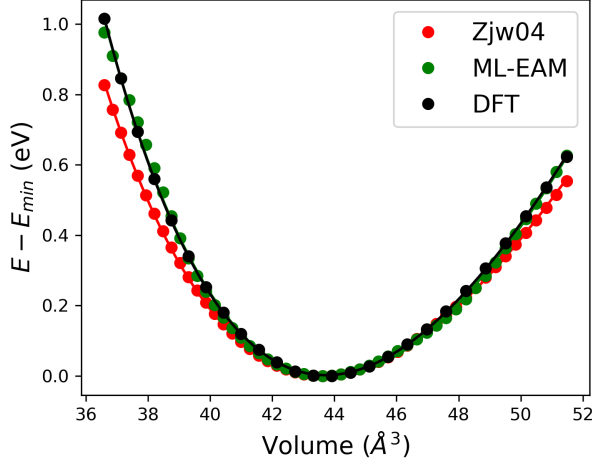
FIG. 2.   The energy-volume curves of the original EAM, ML-EAM and DFT for the fcc Ni.

FIG. 3.   The energy-volume curves of the original EAM, ML-EAM and DFT for the fcc Ni.

MAE:

$$\mathbf{rMAE} = \frac{1}{N} \sum_{i=1}^{N} |\frac{(E_i - E_i^{dft})}{N_i} - (E_{eq} - E_{eq}^{dft})| \quad (36)$$

to measure their performaces. Here N_i is the number of atoms in structure i, E_i is the Zjw04 energy of structure i while E_i^{dft} represents its corresponding DFT energy, E_{eq} and E_{eq}^{dft} refers to the equilibrium energy of Zjw04 and DFT, respectively.

Figure 5 compares the $\rho(r)$, $F(\rho)$, $\phi(r)$, $\mu(r)$ and $\omega(r)$ functions. The differences between ML-EAM and ML-ADP $\rho(r)$, $\phi(r)$ and $F(\rho)$ are quite small. Both the angular-dependent dipole $\mu(r)$ and quadrupole $\omega(r)$ quickly descrease to zero at 2.5 Å, which is lower than the equilibrium spacing (2.7 Å) between two adjacent Mo atoms. Hence, the dipole and quadrupole terms only contribute a little to the total energy for this test case.

For the Ni dataset, the original Zjw04 already has a reasonable performace. The energy **rMAE** is 10.6 meV/atom and the force MAE is just 0.06 eV/Åon the entire dataset. Our machine learning approach can further enhance the performace. ML-EAM achieves 4.1 meV/atom and 0.05 eV/Åon the test set and 3.9 meV/atom and 0.05 eVÅon the entire dataset. ML-EAM is nearly as accurate as SNAP (1.2 meV/atom, 0.05 eVÅ), but later has around three orders of magnitude more computational expense. Figure 1 demonstrates the $\rho(r)$, $F(\rho)$ and $\phi(r)$ functions before (Zjw04) and after machine learning optimization. The pairwise potential $\phi(r)$ almost remains the same while the embedding function $F(\rho)$ and the electron density function $\rho(r)$ changes significantly. Both $\rho(r)$ and $\phi(r)$ converge to zero at around $r = 4.8$ Å, which is rougly 2 times of the equilibrium spacing between nearest neighbors.

For the Mo dataset, the situation is quite different. The original Zjw04 has extremely large MAEs: the energy **rMAE** is $\sim$ 98 meV/atom and the force MAE
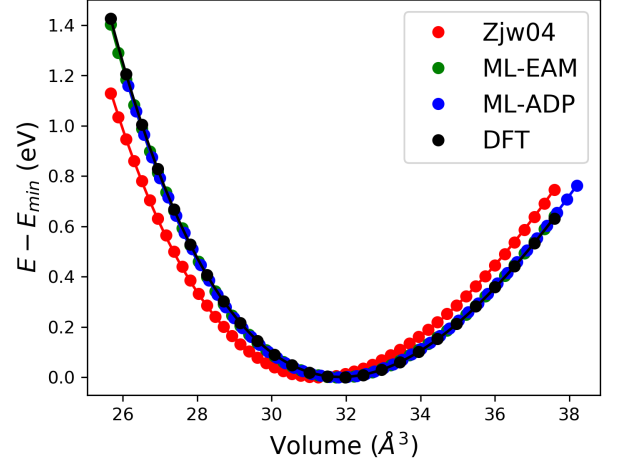
is more than 0.40 eV/Å. Our method can significantly reduce the error: ML-EAM can achieve test MAEs of 21.9 meV/atom and 0.28 eV/Åand overall MAEs of 28.1 meV/atom and 0.32 eV/Å. We also optimized an ADP potential for this dataset. Interestingly, ML-ADP achieves almost exactly the same MAEs with ML-EAM: 22.1 meV/atom and 0.27 eVÅon the test set and 27.9 meV/atom and 0.33 eV/Åon the entire dataset. On the contrast, the SNAP method (13.2 meV/atom, 0.25 eV/Å) behaves better. But considering the computational cost, the optimized EAM and ADP are totally accepetable.

Figure 2 and 3 plot the equation of state curves. The fitting region ranges from 84% to 118% of the corresponding equilibrium volume. For both crystals, the original EAM significantly underestimate the energy at both tensile and compress strains. However, our constrained machine learning approach can fix this problem. The energy-volume curves of ML-EAM and ML-ADP overalp with the DFT curves very well.

Table I and II summarize the predicted material properties of fcc Ni and bcc Mo, including the melting point T_m, the elastic constants (c_{11}, c_{12} and c_{44}), the vacancy formation energy E_v, the migration energy E_m and the activation energy E_a. The original Zjw04 potentials were developed by fitting material properties so they already perform well on these metrics. Our ML-EAM and ML-ADP not only inherit this feature but also gain significant improvement on predicting elastic constants: the relative errors of the elastic constants as well as the Voigt-Reuss-Hill bulk modulus, shear modulus and homogeneous Poisson's ratio, for both bcc Mo and fcc Ni, are reduced from $\sim$10% to 0-2%.

We further examined our ML-EAM and ML-ADP by calculating the melting temperature of fcc Ni and bcc Mo using the solid-liquid coexistence approach. For fcc Ni, $30 \times 10 \times 10$ supercells were used and for bcc the su-

	DFT	SNAP [36]	EAM [46]	ML-EAM	ML-ADP	Exp.
T_m(K)		3000	3750	2940	2880	2890
c_{11} (GPa)	472	473 (0.2%)	457 (-3.2%)	472 (0.0%)	473 (0.2%)	479 [57]
c_{12} (GPa)	158	152 (-3.8%)	168 (6.3%)	158 (0.0%)	160 (1.2%)	165 [57]
c_{44} (GPa)	106	107 (0.9%)	116 (9.4%)	106 (0.0%)	105 (-0.9%)	108 [57]
B_{VRH} (GPa)	263	259 (-1.5%)	264 (0.4%)	262 (-0.4%)	264 (0.4%)	270 [57]
G_{VRH} (GPa)	124	126 (1.6%)	127 (2.4%)	125 (0.8%)	124 (0.0%)	125 [57]
μ	0.30	0.29 (-3.3%)	0.29 (-3.3%)	0.30 (0.9%)	0.30 (0.0%)	0.30
E_v (eV)	2.87	2.61 (-9.1%)	3.02 (5.2%)	1.16 (-20.5%)	1.71 (17.1%)	
E_m (eV)	1.12	1.39 (24.1%)	1.54 (37.5%)	1.46 (30.5%)	0.87 (-22.3%)	
$E_a = E_v + E_m$ (eV)	3.99	4.00 (-0.1%)	4.56 (14.3%)	2.62 (1.6%)	2.58 (0.0%)	4.00 [58]

TABLE II. Comparison of the calculated and experimental material properties of elementary bcc Mo. SNAP refers to the elementary Mo SNAP model.
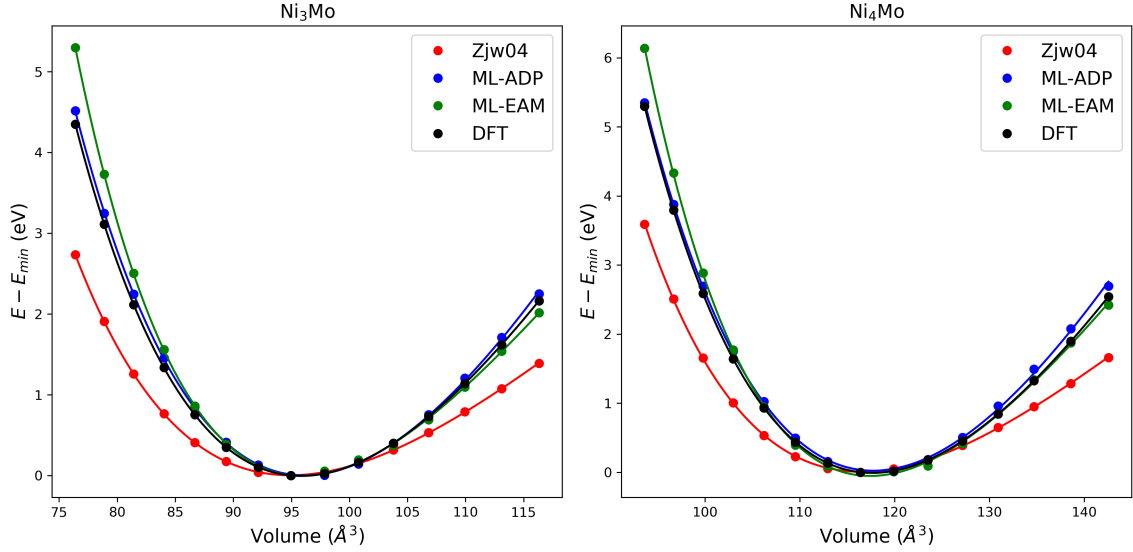

FIG. 4. $\rho(r)$, $F(\rho)$, $\phi(r)$ of the original Zjw04 EAM, ML-EAM and ML-ADP. **(d)** shows the $\mu(r)$ and $\omega(r)$ of ML-ADP.

percells were $30 \times 15 \times 15$. The time step was set to 1 fs. The simulation for each temperature was carried out for at least 300 ps. For fcc Ni, T_m remains the same for ML-EAM, 200 K lower than the experiment T_m. However, the calculated T_m of Mo is greatly improved. The original Zjw04 T_m is around 3750 K, 800 K higher than experiment (2890 K). Both ML-EAM and ML-ADP can give very accurate melting temperature.

B. Mo-Ni

Secondly, we tested our method on the entire binary Mo-Ni dataset. 300 structures were randomly splitted to form the test dataset and the rest 3673 were used to train the potentials. χ_f, χ_{rose} and χ_{elastic} were set to 1, 1 and 0.05 respectively. The batch size was increased to 50. Only the equilibrium fcc Ni and bcc Mo are included in Ω of Equation 31 and 28.

The original EAM uses a simplified interpolation form to describe the pairwise interaction of A-B (Equation 17). However, we did a minor modification: by adding more fitting parameters we use Equation 16 directly. Thus, in total, there are 44 learnable parameters for ML-EAM and 68 for ML-ADP. One should note that in this section, SNAP, ML-EAM and ML-ADP all refer to their corresponding binary form.

Table III summarizes the energy and force MAEs of SNAP, ML-EAM and ML-ADP on the different subsets. Amazingly, ML-ADP becomes the overally best. Both the energy MAE (19.6 meV/atom) and the force MAE (0.20 eV/Å) are smaller than those of the binary SNAP (22.5 meV/atom and 0.23 eV/Å). The pure Mo is still a challenge for ML-ADP as the energy and force MAEs are noticeably large (still much smaller than the original Zjw04 EAM). The binary SNAP predicts well on pure Ni and Mo. But ML-ADP outperforms the SNAP model on binary phases (Ni$_3$Mo, Ni-doped Mo and especially the
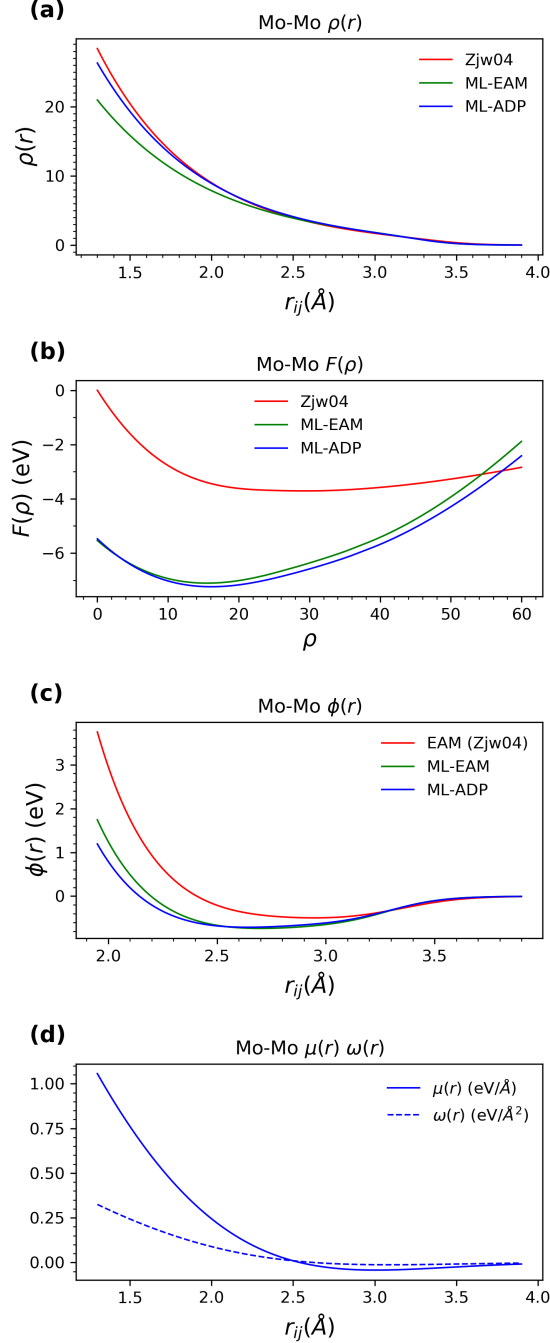
FIG. 5. $\rho(r)$, $F(\rho)$, $\phi(r)$ of the original Zjw04 EAM, ML-EAM and ML-ADP. **(d)** shows the $\mu(r)$ and $\omega(r)$ of ML-ADP.

Mo-doped Ni). On the contrast, ML-EAM performs significantly worse compared with ML-ADP. Interestingly, our results suggest that Mo-Ni dipole and quadrupole interactions are completely negligible (Figure S1 in the appendix). The Ni-Ni dipole and quadrupole interactions are an order of magnitude weaken than those of Mo-Mo.

Figure 4 demonstrates the energy-volume curves of Ni_3Mo and Ni_4Mo. The original Zjw04 EAM can not

properly describe the energy-volume relations of these two crystals. The ML-ADP curves fit perfectly with the DFT curves. ML-EAM fits well with tensile strains but overestimates the energy under compress strains.

Table IV summarizes the elastic constants predictions of Ni_3Mo and Ni_4Mo. One should note that these two crystals were not included in Ω of Equation 31 and 28 but their elastic properties were used to tune the binary SNAP model. The original Zjw04 EAM performs obviously poorly on predicting elastic constants. Some absolute percentage errors can even exceed 100%. With our machine learning correction, ML-EAM, especially ML-ADP, can give much more accurate predictions. For the Ni_3Mo crystal, the average absolute percentage error is reduced from 40% (Zjw04) to 20% (ML-ADP). ML-ADP even beats SNAP on 4 (c_{11}, c_{12}, c_{22} and c_3) of the 7 elastic metrics. For the Ni_4Mo crystal, the average absolute percentage error is also descreased.

IV. DISCUSSIONS

The above results suggest that with machine learning and essential physical constraints, for the fcc and bcc solids, the traditional empirical potentials, EAM/ADP, can be improved to be almost as accurate as the SNAP method. But, EAM/ADP are around 10^2 to 10^3 times faster than SNAP. To understand this, we can start from the many-body expansion (MBE) scheme [59].

MBE is one of the most widely used schemes for fitting potential energy surfaces. In the many-body expansion scheme, the total energy of a system with N atoms can be expressed as the sum of all k-body terms where $k \leq 1$:

$$E = \sum_i^{C_1^N} E_i^{(1)} + \sum_{i,j}^{C_2^N} E_{ij}^{(2)} + \sum_{i,j,k}^{C_3^N} E_{ijk}^{(3)} + \cdots \quad (37)$$

where C_k^N is the binomial coefficient and $E^{(k)}$ represents the k-body contribution. In many cases, higher-order terms like E_{ij} or E_{ijk} are symmetric: $E_{ij} = E_{ji}$, $E_{ijk} = E_{ikj} = E_{jik} = \cdots$. Thus, Equation 37 can be further transformed to:

$$E = \sum_i^N \left(E_i^{(1)} + \frac{1}{2!} \sum_{j \neq i}^N E_{ij}^{(2)} + \frac{1}{3!} \sum_{j \neq i}^N \sum_{k \neq i,j}^N E_{ijk}^{(3)} + \cdots \right)$$
$$= \sum_i^N E_i \quad (38)$$

where E_i is the atomic energy of atom i.

In fact, both the EAM and the SNAP formalisms can be considered as variants of MBE truncated at $k = 2$. For the EAM method, the embedding contribution $F(\rho)$ serves as the one-body term and the overall pairwise contribution $\frac{1}{2} \sum_{j \neq i} \phi(r_{ij})$ is just a plain transcription of the two-body term in Equation 38. The ADP formalism, which introduces extra pairwise dipole and quadrupole

	Model	Mo	Ni$_4$Mo	Ni$_3$Mo	Mo$_{Ni}$	Ni$_{Mo}$	Ni	Overall
Energy (meV/atom)	SNAP [37]	16.2	4.0	5.2	22.7	33.9	7.9	22.5
	ML-EAM	30.4	11.1	8.6	29.8	33.9	11.6	26.2
	ML-ADP	36.9	5.1	4.8	21.7	22.8	13.3	19.6
Force (eV/Å)	SNAP [37]	0.29	0.14	0.16	0.13	0.55	0.11	0.23
	ML-EAM	0.33	0.24	0.19	0.25	0.35	0.08	0.23
	ML-ADP	0.45	0.16	0.15	0.14	0.35	0.08	0.20

TABLE III. Comparion of the MAEs in predicted energies (mev/atom) and forces (eV/Å) relative to the DFT on the entire Mo-Ni dataset.

	DFT	SNAP [37]	EAM [46]	ML-EAM	ML-ADP
Ni$_3$Mo					
c_{11}	385	420 (9.1%)	195 (-49.4%)	403 (4.7%)	374 (-2.9%)
c_{12}	166	197 (18.7%)	98 (-41.0%)	208 (25.3%)	160 (-3.6%)
c_{13}	145	162 (11.7%)	98 (-32.4%)	230 (58.6%)	176 (21.4%)
c_{22}	402	360 (10.7%)	351 (-12.7%)	443 (10.2%)	386 (-4.0%)
c_{23}	131	145 (-10.4%)	107 (-18.3%)	272 (107.6%)	214 (63.3%)
c_{33}	402	408 (1.5%)	295 (-26.6%)	474 (17.9%)	398 (-0.5%)
c_{44}	94	84 (-10.4%)	36 (-61.7%)	25 (-73.4%)	52 (-44.6%)
B_{VRH}	230	243 (5.7%)	156 (-32.2%)	302 (31.3%)	250 (8.7%)
G_{VRH}	89	100 (12.4%)	61 (-31.5%)	57 (36.0%)	66 (25.8%)
μ	0.33	0.32 (-3.0%)	0.33 (0.0%)	0.41 (24.2%)	0.38 (15.1%)
Ni$_4$Mo					
c_{11}	300	283 (-5.7%)	172 (-42.7%)	282 (-6.0%)	342 (14.0%)
c_{12}	186	179 (-3.8%)	158 (-15.1%)	125 (-32.8%)	217 (16.7%)
c_{22}	313	326 (4.2%)	158 (-49.5%)	282 (-10.0%)	342 (9.3%)
c_{23}	166	164 (-1.2%)	80 (-51.8%)	156 (-6.0%)	232 (39.8%)
c_{44}	130	126 (-3.1%)	125 (-3.8%)	136 (4.6%)	117 (-10.0%)
B_{VRH}	223	220 (-1.3%)	161 (-27.8%)	178 (-20.1%)	257 (15.2%)
G_{VRH}	91	95 (4.4%)	-156 (-162%)	80 (-12.1%)	79 (-13.2%)
μ	0.33	0.31 (-6.1%)	0.70 (112%)	0.31 (-6.1%)	0.36 (9.1%)

TABLE IV. Comparison of elastic constants (c_{ij}, GPa), Voigt-Reuss-Hill bulk modulus (B_{VRH}, GPa), Voigt-Reuss-Hill shear modulus (G_{VRH}, GPa) and homogeneous Poisson's ratio (μ) for fcc Ni, fcc Mo and binary alloys Ni$_3$Mo and Ni$_4$Mo.

interactions, does not change this nature. For the SNAP method, the atomic energy is calculated with the following equation:

$$E_i = \beta_0^{\alpha_i} + \sum_k \beta_k^{\alpha_i} B_k^i \qquad (39)$$

where $\beta_0^{\alpha_i}$ and $\beta_k^{\alpha_i}$ are learnable scalar parameters, α_i indicates the element type of atom i. B_k^i is a bispectrum coefficient. The calculation of B_k^i is very complicated but it is still a two-body term. The overall two-body contribution is a linear combination of B_k^i. Thus, the SNAP formalism can be viewed as an assemble of a simplified one-body term with a complicated two-body term. On the other hand, the EAM formalism is built up with a 'complicated' one-body term and a rather simple two-body term. Hence, for the fcc and bcc systems, both methods perform similarly.

As we analysed in Section II B, the fundamental atomic descriptor of EAM, the $N \times N^{nl} \times 4$ matrix $\mathbf{G}$, is composed of just raw interatomic distances. However, interatomic distances alone may not uniquely describe a structure. To avoid this problem, the cutoff was raised up to 6.5 Å. As a comparison, the overall r_{cut} is 4.6 Åfor the binary SNAP. But such descriptor is still too rough to describe complicated potential energy surfaces (e.g. bcc Mo). To further enhance EAM/ADP, perhaps we shall consider adding real three-body terms (e.g. Tersoff).

V. CONCLUSIONS

To conclude, we have successfully combined empirical potentials (EAM/ADP) with machine learning. The machine learning approaches (big data, SGD, etc), together with physical constraints (Rose EOS, elastic constants) can significantly boost the performaces of EAM/ADP.

For the fcc Ni, bcc Mo and Mo-Ni alloys, ML-EAM and ML-ADP can be as accurate as SNAP, while later method is orders of magnitude slower. Our work also indicates a new route to design and develop machine learning interatomic potentials —the machine learning enhanced empirical potential approach.

[1] M. S. Daw and M. I. Baskes, Phys. Rev. Lett **50**, 1285 (1983).
[2] M. S. Daw and M. I. Baskes, Phys. Rev. B **29**, 6443 (1984).
[3] M. I. Baskes, J. S. Nelson, and A. F. Wright, Phys. Rev. B **40**, 6085 (1989).
[4] R. A. Johnson, Phys. Rev. B **39**, 12554 (1989).
[5] M. I. Baskes, Phys. Rev. Lett **59**, 2666 (1987).
[6] M. I. Baskes, Phys. Rev. B **46**, 2727 (1992).
[7] M. I. Baskes and R. A. Johnson, Modelling Simul. Mater. Sci. Eng. **2**, 147 (1994).
[8] B.-J. Lee, J.-H. Shim, and M. I. Baskes, Phys. Rev. B **68**, 10.1103/PhysRevB.68.144112 (2003).
[9] B. Jelinek, S. Groh, M. F. Horstemeyer, J. Houze, S. G. Kim, G. J. Wagner, A. Moitra, and M. I. Baskes, Phys. Rev. B **85**, 10.1103/PhysRevB.85.245102 (2012).
[10] D. G. Pettifor, M. W. Finnis, D. Nguyen-Manh, D. A. Murdick, X. W. Zhou, and H. N. G. Wadley, Mater. Sci. Eng. A **365**, 2 (2004).
[11] D. A. Murdick, X. W. Zhou, H. N. G. Wadley, D. Nguyen-Manh, R. Drautz, and D. G. Pettifor, Phys. Rev. B **73**, 10.1103/PhysRevB.73.045206 (2006).
[12] B. A. Gillespie, X. W. Zhou, D. A. Murdick, H. N. G. Wadley, R. Drautz, and D. G. Pettifor, Phys. Rev. B **75**, 10.1103/PhysRevB.75.155207 (2007).
[13] D. K. Ward, X. W. Zhou, B. M. Wong, F. P. Doty, and J. A. Zimmerman, Phys. Rev. B **85**, 10.1103/PhysRevB.85.115206 (2012).
[14] Y. Mishin and A. Y. Lozovoi, Acta Mater. **54**, 5013 (2006).
[15] A. Hashibon, A. Y. Lozovoi, Y. Mishin, C. Elsässer, and P. Gumbsch, Phys. Rev. B **77**, 10.1103/PhysRevB.77.094131 (2008).
[16] F. Apostol and Y. Mishin, Phys. Rev. B **82**, 10.1103/PhysRevB.82.144115 (2010).
[17] F. Apostol and Y. Mishin, Phys. Rev. B **83**, 10.1103/PhysRevB.83.054116 (2011).
[18] Y. Mishin, M. J. Mehl, and D. A. Papaconstantopoulos, Acta Mater. **53**, 4029 (2005).
[19] D. J. Wales and J. P. K. Doye, J. Phys. Chem. A **101**, 5111 (1997).
[20] D. J. Wales and H. A. Scheraga, Science **285**, 1368 (1999).
[21] E. Jones, T. Oliphant, P. Peterson, *et al.*, SciPy: Open source scientific tools for Python (2001–), [Online; accessed ¡today¿].
[22] H. Mühlenbein, M. Schomisch, and J. Born, Parallel Computing **17**, 619 (1991).
[23] J. Behler and M. Parrinello, Phys. Rev. Lett **98**, 146401 (2007).
[24] J. Behler, Phys. Chem. Chem. Phys **13**, 17930 (2011).
[25] J. Behler, J. Chem. Phys **134**, 074106 (2011).
[26] J. Behler, J. Phys. Condens. Matter **26**, 183001 (2014).
[27] J. Behler, Int. J. Quantum Chem. **115**, 1032 (2015).
[28] S. Hajinazar, J. Shao, and A. N. Kolmogorov, Phys. Rev. B **95**, 10.1103/PhysRevB.95.014114 (2017).
[29] R. Kobayashi, D. Giofré, T. Junge, M. Ceriotti, and W. A. Curtin, Phys. Rev. M **1**, 10.1103/PhysRevMaterials.1.053604 (2017).
[30] B. Onat, E. D. Cubuk, B. D. Malone, and E. Kaxiras, Phys. Rev. B **97**, 10.1103/PhysRevB.97.094106 (2018).
[31] A. P. Bartok, M. C. Payne, R. Kondor, and G. Csanyi, Phys. Rev. Lett **104**, 136403 (2010).
[32] A. P. Bartók and G. Csányi, Int. J. Quantum Chem. **115**, 1051 (2015).
[33] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, Sci. Adv **3**, 10.1126/sciadv.1701816 (2017).
[34] A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, Phys. Rev. X **8**, 10.1103/PhysRevX.8.041048 (2018).
[35] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, J. Comput. Phys **285**, 316 (2015).
[36] C. Chen, Z. Deng, R. Tran, H. Tang, I.-H. Chu, and S. P. Ong, Phys. Rev. M **1**, 10.1103/PhysRevMaterials.1.043603 (2017).
[37] X. G. Li, C. Hu, C. Chen, Z. Deng, J. Luo, and S. P. Ong, Phys. Rev. B **98**, 094104 (2018).
[38] M. A. Wood, M. A. Cusentino, B. D. Wirth, and A. P. Thompson, Phys. Rev. B **99**, 184305 (2019).
[39] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* (USENIX Association, Savannah, GA, 2016) pp. 265–283.
[40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, in *NIPS-W* (2017).
[41] C. Trott and A. P. Thompson, *A Brief Description of the Kokkos implementation of the SNAP potential in ExaMiniMD*, Report (2017).
[42] T. J. Lenosky, B. Sadigh, E. Alonso, V. V. Bulatov, T. D. d. l. Rubia, J. Kim, A. F. Voter, and J. D. Kress, Modelling Simul. Mater. Sci. Eng. **8**, 825 (2000).
[43] G. P. Purja Pun, K. A. Darling, L. J. Kecskes, and Y. Mishin, Acta Mater. **100**, 377 (2015).
[44] H. N. G. Wadley, X. W. Zhou, R. A. Johnson, and M. Neurock, Prog. Mater. Sci **46**, 329 (2001).
[45] X. W. Zhou, H. N. G. Wadley, R. A. Johnson, D. J. Larson, N. Tabat, A. Cerezo, A. K. Petford-Long, G. D.

Smith, P. H. Clifton, R. L. Martens, and T. F. Kelly, Acta Mater. **49**, 4005 (2001).

[46] X. W. Zhou, R. A. Johnson, and H. N. G. Wadley, Phys. Rev. B **69**, 10.1103/PhysRevB.69.144113 (2004).

[47] J. H. Rose, J. R. Smith, F. Guinea, and J. Ferrante, Phys. Rev. B **29**, 2963 (1984).

[48] R. R. Zope and Y. Mishin, Phys. Rev. B **68**, 10.1103/PhysRevB.68.024102 (2003).

[49] Y. Mishin, Acta Mater. **52**, 1451 (2004).

[50] N. C. Admal, J. Marian, and G. Po, J. Mech. Phys. Solids **99**, 93 (2017).

[51] G. Alers, J. Neighbours, and H. Sato, Journal of Physics and Chemistry of Solids **13**, 40 (1960).

[52] E. H. Megchiche, S. Pérusin, J.-C. Barthelat, and C. Mijoule, Phys. Rev. B **74**, 064111 (2006).

[53] D. P. Kingma and J. Ba, CoRR **abs/1412.6980** (2014), arXiv:1412.6980.

[54] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).

[55] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett **77**, 3865 (1996).

[56] P. E. Blöchl, Phys. Rev. B **50**, 17953 (1994).

[57] G. Simmons and H.Wang, Single crystal elastic constants and calculated aggregate properties, MIT Press, Cambridge, MA (1971).

[58] J. Askill and D. H. Tomlin, The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics **8**, 997 (1963), https://doi.org/10.1080/14786436308214459.

[59] X. Chen, M. S. Jorgensen, J. Li, and B. Hammer, J. Chem. Theory Comput. **14**, 3933 (2018).