Machine-learned embedded atom method

Xin Chen, Li-Fang Wang, De-Ye Lin*, 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

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), modified embedded-atom method (MEAM), bond-order potential (BoP), or angular-dependent potential (ADP), 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, genetic algorithm, 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, is still the most popular choice in modeling metallic interactions. The smooth overlap atomic positions descriptor based gaussian approximation potential (SOAP-GAP), 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) and it has been proven working on a broad range of metals and alloys.

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, PyTorch, 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 necessarity. 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 suggest that MILPs are still too expensive. At present, most MLIPs are used to examine small to medium (10^3 to 10^4) systems.

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 and details of the machine learned EAM. Section III summarizes the training results and the optimal parameters. Applications of the new potentials are discussed in Section IV.

II. METHOD

A. Theory

In the original EAM formalism, 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. In the Finnis-Sinclair model, the electron density has a slightly modified form:

$$\rho_i^{\mathrm{FS}} = \sum_{j}^{r_{ij}<r_c} \rho_{ab}(r_{ij}) \tag{3}$$

F, ρ and ϕ can be either parameterized functions or cubic splines.

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

$$\begin{aligned} E^{total} &= E^{\mathrm{EAM}} \\ &+ \frac{1}{2}\sum_{i}\sum_{\alpha}(\mu_i^{\alpha})^2 \\ &+ \frac{1}{2}\sum_{i}\sum_{\alpha}\sum_{\beta}(\lambda_i^{\alpha\beta})^2 \\ &- \frac{1}{6}\sum_{i}\nu_i^2 \end{aligned} \tag{4}$$

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

$$\mu_i^{\alpha} = \sum_{j\neq i} \mu_{ab}(r_{ij}) r_{ij}^{\alpha} \tag{5}$$

$$\lambda_i^{\alpha\beta} = \sum_{j\neq i} \omega_{ab}(r_{ij}) r_{ij}^{\alpha} r_{ij}^{\beta} \tag{6}$$

$$\nu_i = \sum_{\alpha} \lambda_i^{\alpha\alpha} \tag{7}$$

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:

$$\begin{aligned} E_i^A =& \frac{1}{2}\sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) + \frac{1}{2}\sum_{j\neq i}^{N_i^{\mathrm{AB}}} \phi_{\mathrm{AB}}(r_{ij}) \\ &+ F_{\mathrm{A}}\left(\sum_{j\neq i}^{N_i^{\mathrm{AA}}} \rho_{\mathrm{A}}(r_{ij}) + \sum_{j\neq i}^{N_i^{\mathrm{AB}}} \rho_{\mathrm{B}}(r_{ij})\right) \end{aligned} \tag{8}$$

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

$$\begin{aligned} E_j^B =& \frac{1}{2}\sum_{i\neq j}^{N_j^{\mathrm{BB}}} \phi_{\mathrm{BB}}(r_{ij}) + \frac{1}{2}\sum_{i\neq j}^{N_j^{\mathrm{BA}}} \phi_{\mathrm{AB}}(r_{ij}) \\ &+ F_{\mathrm{B}}\left(\sum_{j\neq i}^{N_j^{\mathrm{BB}}} \rho_{\mathrm{B}}(r_{ij}) + \sum_{j\neq i}^{N_j^{\mathrm{BA}}} \rho_{\mathrm{A}}(r_{ij})\right) \end{aligned} \tag{9}$$

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}:

$$N^{\mathrm{nl}} = \max\left(N_i^{\mathrm{nl}}\right) \tag{10}$$

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:

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{11}$$

then the pairwise term can be transformed to:

$$\begin{aligned} \sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) =& \sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) \cdot 1 + \sum_{j\neq i}^{N^{\mathrm{nl}}-N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(0) \cdot 0 \\ =& \phi_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) \end{aligned} \tag{12}$$

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 9 in an equivalent expression:

$$\begin{aligned} E_i^{\mathrm{A}} =& \frac{1}{2}\left(\phi_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) + \phi_{\mathrm{AB}}(\vec{\mathbf{r}}_i^{\mathrm{AB}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AB}})\right) \\ &+ F_{\mathrm{A}}\left(\rho_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) + \rho_{\mathrm{AB}}(\vec{\mathbf{r}}_i^{\mathrm{AB}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AB}})\right) \end{aligned} \tag{13}$$

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}:

$$\begin{aligned} E_j^{\mathrm{B}} =& \frac{1}{2}\left(\phi_{\mathrm{BB}}(\vec{\mathbf{r}}_j^{\mathrm{BB}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BB}}) + \phi_{\mathrm{BA}}(\vec{\mathbf{r}}_j^{\mathrm{BA}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BA}})\right) \\ &+ F_{\mathrm{A}}\left(\rho_{\mathrm{BB}}(\vec{\mathbf{r}}_j^{\mathrm{BB}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BB}}) + \rho_{\mathrm{BA}}(\vec{\mathbf{r}}_j^{\mathrm{BA}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BA}})\right) \end{aligned} \tag{14}$$

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:

$$\mathbf{g}_i = \left[\vec{\mathbf{r}}_i^{\mathrm{AA}} \;\; \vec{\mathbf{r}}_i^{\mathrm{AB}} \;\; \vec{\mathbf{r}}_i^{\mathrm{BB}} \;\; \vec{\mathbf{r}}_i^{\mathrm{BA}}\right] \tag{15}$$

$\mathbf{g}_i$ 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) 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:

$$\rho_b(r) = \frac{f_e \exp\left[-\beta\left(r/r_e - 1\right)\right]}{1 + \left(r/r_e - \lambda\right)^{20}} \tag{16}$$

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

$$\phi_{aa}(r) = \frac{A \exp\left[-\alpha\left(r/r_e - 1\right)\right]}{1 + \left(r/r_e - \kappa\right)^{20}} - \frac{B \exp\left[-\beta\left(r/r_e - 1\right)\right]}{1 + \left(r/r_e - \lambda\right)^{20}} \tag{17}$$

where A, B, α and κ are also trainable parameters, β and κ are used in Equation 16 already. For the pairwise interaction between two atoms of different species, Zhou et al proposed an interpolation form:

$$\phi_{ab}(r) = \frac{1}{2}\left(\frac{\rho_b(r)}{\rho_a(r)}\phi_{aa}(r) + \frac{\rho_a(r)}{\rho_b(r)}\phi_{bb}(r)\right) \tag{18}$$

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

$$F(\rho) = \begin{cases} \sum_{i=0}^{3} F_{ni}\left(\frac{\rho}{\rho_n} - 1\right)^i & \rho < \rho_n \\ \sum_{i=0}^{3} F_i\left(\frac{\rho}{\rho_e} - 1\right)^i & \rho_n \leq \rho < \rho_0 \\ F_e\left[1 - \eta \ln\left(\frac{\rho}{\rho_s}\right)\right](\frac{\rho}{\rho_s})^{\eta} & \rho_0 \leq \rho \end{cases} \tag{19}$$

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 19:

$$\begin{aligned} F(\rho) =& c_1 \cdot \sum_{i=0}^{3} F_{ni}\left(\frac{\rho}{\rho_n} - 1\right)^i \\ &+ c_2 \cdot \sum_{i=0}^{3} F_i\left(\frac{\rho}{\rho_e} - 1\right)^i \\ &+ c_3 \cdot F_e\left[1 - \eta \ln\left(\frac{\rho}{\rho_s}\right)\right](\frac{\rho}{\rho_s})^{\eta} \end{aligned} \tag{20}$$

$$c_1 = \frac{1}{1 + e^{-2(\rho_n - \rho)}} \tag{21}$$

$$c_3 = \frac{1}{1 + e^{-2(\rho - \rho_0)}} \tag{22}$$

$$c_2 = 1 - c_1 - c_3 \tag{23}$$

ω_1 and ω_3 are just damping factors calculated by the sigmoid functions (Equations 21 and 22).

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

$$\mu_{ab}(r) = \left[d_1^{ab} \exp\left(-d_2^{ab} r\right) + d_3^{ab}\right] \psi\left(\frac{r - r_0}{r_h}\right) \tag{24}$$

$$\omega_{ab}(r) = \left[q_1^{ab} \exp\left(-q_2^{ab} r\right) + q_3^{ab}\right] \psi\left(\frac{r - r_0}{r_h}\right) \tag{25}$$

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

$$\psi(x) = \begin{cases} 0 & x \geq 0 \\ \frac{x^4}{1 + x^4} & x < 0 \end{cases} \tag{26}$$

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) 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

$$E(x) = E_0 \left[1 + \alpha x + \beta \alpha^3 x^3 \frac{2x+3}{(x-1)^2}\right] e^{-\alpha x} \tag{27}$$

is used because the original form tends to underestimate energies under high pressures. In Equation 27, 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

$$\alpha = \sqrt{-\frac{9V_0 B}{E_0}} \tag{28}$$

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 $\mathbf{L}^{\mathrm{Rose}}$ is measured as the 2-norm of the energy differences between $E(x)^{\mathrm{dft}}$ and their corresponding predicted $E(x)$:

$$\mathbf{L}^{\mathrm{Rose}} = \sum_{\Omega} \sqrt{\sum_{i} \left(E(x_i) - E(x_i)^{\mathrm{dft}}\right)^2} \tag{29}$$

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:

$$V \cdot \epsilon = -\mathbf{F}^T \mathbf{R} + \left(\frac{\partial E^{total}}{\partial \mathbf{h}}\right)^T \mathbf{h} \tag{30}$$

$$c_{ijkl}|_{\epsilon \to 0, \mathbf{F} \to 0} = \frac{1}{V} \left[\left(\frac{\partial \epsilon_{ij}}{\partial \mathbf{h}}\right)^{\mathrm{T}} \mathbf{h}\right]_{kl} \tag{31}$$

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 $\mathbf{L}^{\mathrm{elastic}}$ contributed by the elastic tensor is also measured by the RMSE between c_{ijkl} and c_{ijkl}^{dft}:

$$\mathbf{L}^{\mathrm{elastic}} = \sum_{\Omega} \left(\omega_\Omega \cdot \mathbf{RMSE}_\Omega + ||\epsilon|| + ||\mathbf{F}||\right) \tag{32}$$

$$\omega_\Omega = \mathbf{ReLU}(\mathbf{MAE}_\Omega - \tau) \tag{33}$$

$$\mathbf{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{34}$$

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, $\mathbf{L}^{\mathrm{elastic}}$ 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 35) and virial stress (Equation 30) can be derived by the AutoGrad module of TensorFlow and calcualted automatically and efficiently.

$$\mathbf{F} = -\frac{\partial E^{total}}{\partial \mathbf{R}} \tag{35}$$

This direct computation graph also plays a key role in computing analytical elastic tensor (Equation 31) and the Rose loss (Equation 27).

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 36 demonstrates the loss function used in this work:

$$\begin{aligned}\textbf{Loss} =& \sqrt{\frac{1}{N_b}\sum_{i=1}^{N_b}\left(E_i - E_i^{\text{dft}}\right)^2} \\ &+ \chi_{\text{f}}\sqrt{\frac{1}{3\sum_i^{N_b} N_i}\sum_i^{N_b}\sum_j^{N_i}\sum_\alpha\left(f_{ij\alpha} - f_{ij\alpha}^{\text{dft}}\right)^2} \\ &+ \chi_{\text{s}}\sqrt{\frac{1}{6N_b}\sum_i^{N_b}\sum_j^{6}\left(\epsilon_j^{\text{voigt}} - \epsilon_j^{\text{voigt,dft}}\right)^2} \\ &+ \chi_{\text{rose}}\mathbf{L}^{\text{Rose}} + \chi_{\text{elastic}}\mathbf{L}^{\text{elastic}} \qquad (36)\end{aligned}$$

where N_b is the batch size and N_i is the number of atoms in structure i. χ_{f}, χ_{s}, χ_{rose} and χ_{elastic} are overall weights of force, virial stress, rose EOS and elastic contributions. The superscript 'voigt' means that virial stress tensors should be converted to Voigt form. The Adam optimizer is used to minimize Equation 36. 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 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 with the PBE functional and the projector augmented-wave approach.

The melting points T_m of bcc Mo and fcc Ni were measured with the solid-liquid coexistence approach. For fcc Ni, $30 \times 10 \times 10$ supercells were used and for bcc the supercells were $30 \times 15 \times 15$. The time step was set to 1 fs. The simulation for each temperature was carried out for 300 ps.

The optimized parameters are listed in the appendix.

A. Ni

We first use the elementary fcc Ni dataset to show our approach. This dataset has 461 structures in total. 400 were randomly selected to form the training subset. In this experiment, χ_{f}, χ_{rose} and χ_{elastic} were set to 1, 3 and 0.05 respectively. The batch size was 25. Only the fcc Ni was included in Ω (Equation 29 and 32).

Table I shows the MAEs of those models. The original EAM model already has an accepetable performace on this fcc dataset. Its energy MAE is 10.6 meV/atom and the force MAE is 0.06 eV/Å. The MEAM performs slightly worse on this dataset with an energy MAE of 17.6 meV/atom and force MAE of 0.08 eV/Å. Our machine learning approach can further reduce the energy MAE to 4.1 meV/atom and the force MAE to 0.05 eV/Å, which are much closer to the elementary SNAP Ni model.

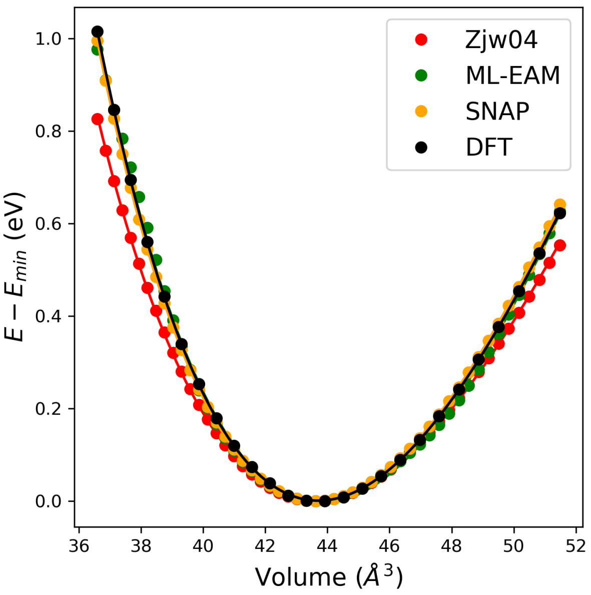

FIG. 1. The energy-volume curves of the original EAM, SNAP, ML-EAM and DFT for the fcc Ni.

Figure 1 plots the equation of state curves obtained with the DFT, SNAP, EAM and ML-EAM models. The whole fitting region ranges from 84% to 118% of the equilibrium volume. The original EAM significantly underestimates the energy at both tensile and compress strains. However, our machine learning approach together with the Rose constraint can fix this problem. The energy-volume curve of ML-EAM overalps with the DFT curve very well for the entire fitting region.

Table II summerizes the predicted material properties of fcc Ni, including the melting point, the elastic constants, the vacancy formation energy E_v, the migration energy E_m and the activation energy E_a. The original EAM already performs well on these properties. ML-EAM further improves the elastic constants. The errors were significantly reduced to around 1-2%.

MAE	SNAP	EAM	MEAM	ML-EAM
Energy (meV/atom)	1.2	10.6	17.8	4.1
Force (eV/Å)	0.05	0.06	0.08	0.05

TABLE I. Comparion of the energy and force accuracy of the elementary SNAP Ni model, the original EAM, the MEAM and our ML-EAM.

Figure 2 compares the $rho(r)$, $F(\rho)$ and $\phi(r)$ functions of ML-EAM with original Zjw04 EAM. 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

	DFT	SNAP	EAM	MEAM	ML-EAM	Exp.
T_m(K)		1785	1520	1765	1520	1728
c_{11} (GPa)	276	276 (0.0%)	248 (-10.1%)	260 (-5.8%)	274 (-0.7%)	261
c_{12} (GPa)	159	159 (0.0%)	147 (-7.5%)	151 (0.0%)	163 (2.5%)	151
c_{44} (GPa)	132	132 (0.0%)	125 (-5.3%)	131 (0.0%)	131 (-0.8%)	132
E_v (eV)	1.46	1.68 (15.1%)	1.68 (15.1%)	1.16 (-20.5%)	1.71 (17.1%)	1.54–1.80
E_m (eV)	1.12	1.07 (-4.5%)	0.90 (-19.6%)	1.46 (30.5%)	0.87 (-22.3%)	1.01–1.48
$E_a = E_v + E_m$ (eV)	2.58	2.75 (6.6%)	2.58 (0.0%)	2.62 (1.6%)	2.58 (0.0%)	2.77–2.95

TABLE II. 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.

equilibrium spacing between nearest neighbors.

B. Mo

The second test example is bcc Mo. This dataset has 284 structures. 34 of them were used as the testing subset and the rest were used for training. The training settings were the same with fcc Ni. An ADP potential was also optimized for bcc Mo.

Table III summerizes the energy and force MAEs on the test subset. The original Zjw04 EAM performs significantly worse. The energy

MAE	SNAP	EAM	ML-EAM	ML-ADP
Energy (meV/atom)	13.2	58.9	21.7	22.8
Force (eV/Å)	0.25	0.31	0.28	0.27

TABLE III. Comparion of the energy and force accuracy of the elementary SNAP Ni model, the original EAM, the MEAM and our ML-EAM.

C. Mo-Ni

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 as accurate as the SNAP method. However, EAM/ADP are around 10^2 to 10^3 times faster than SNAP.

The many-body expansion (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 $k \leq 1$:

$$E = \sum_{i}^{\mathrm{C}_1^\mathrm{N}} E_i^{(1)} + \sum_{i,j}^{\mathrm{C}_2^\mathrm{N}} E_{ij}^{(2)} + \sum_{i,j,k}^{\mathrm{C}_3^\mathrm{N}} E_{ijk}^{(3)} + \cdots \tag{37}$$

where $\mathrm{C}_\mathrm{k}^\mathrm{N}$ 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:

$$\begin{aligned} E &= \sum_{i}^{\mathrm{N}} \left(E_i^{(1)} + \frac{1}{2!} \sum_{j \neq i}^{\mathrm{N}} E_{ij}^{(2)} + \frac{1}{3!} \sum_{j \neq i}^{\mathrm{N}} \sum_{k \neq i,j}^{\mathrm{N}} E_{ijk}^{(3)} + \cdots \right) \\ &= \sum_{i}^{\mathrm{N}} E_i \end{aligned} \tag{38}$$

where E_i is the atomic energy of atom i.

In fact, the EAM formalism can be considered as a variant of MBE truncated at $k = 2$ (Equation 1). The embedding contribution $F(\rho)$ serves as the one-body term. The 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 uses a much more complicated two-body term by introducing the pairwise dipole and quadrupole terms.

Although the SNAP method is quite complicated, the current formalism can still be viewed as a variant of MBE truncated at $k = 2$. The atomic energy

V. CONCLUSIONS

ACKNOWLEDGMENTS

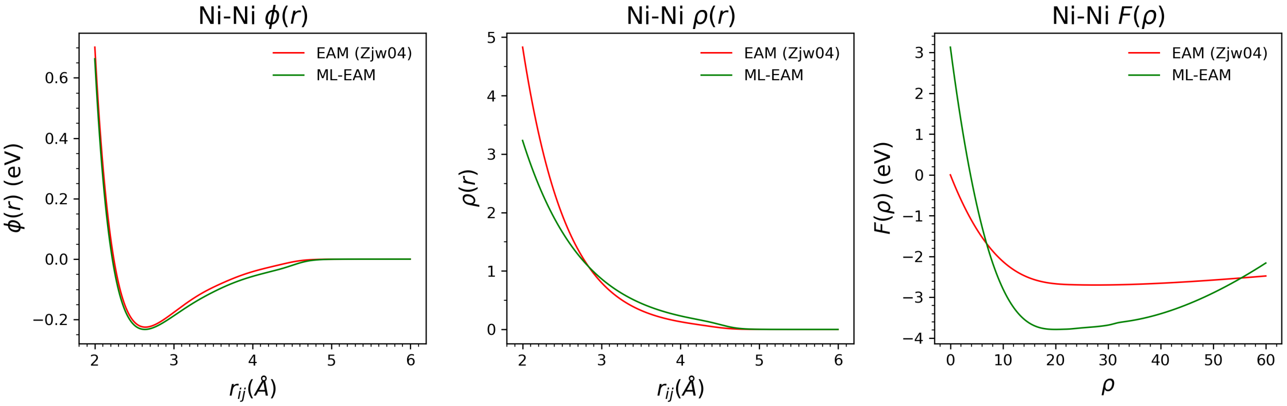

FIG. 2. $\rho(r)$, $F(\rho)$ and $\phi(r)$ of the original Zjw04 EAM and the machine learned EAM (ML-EAM).

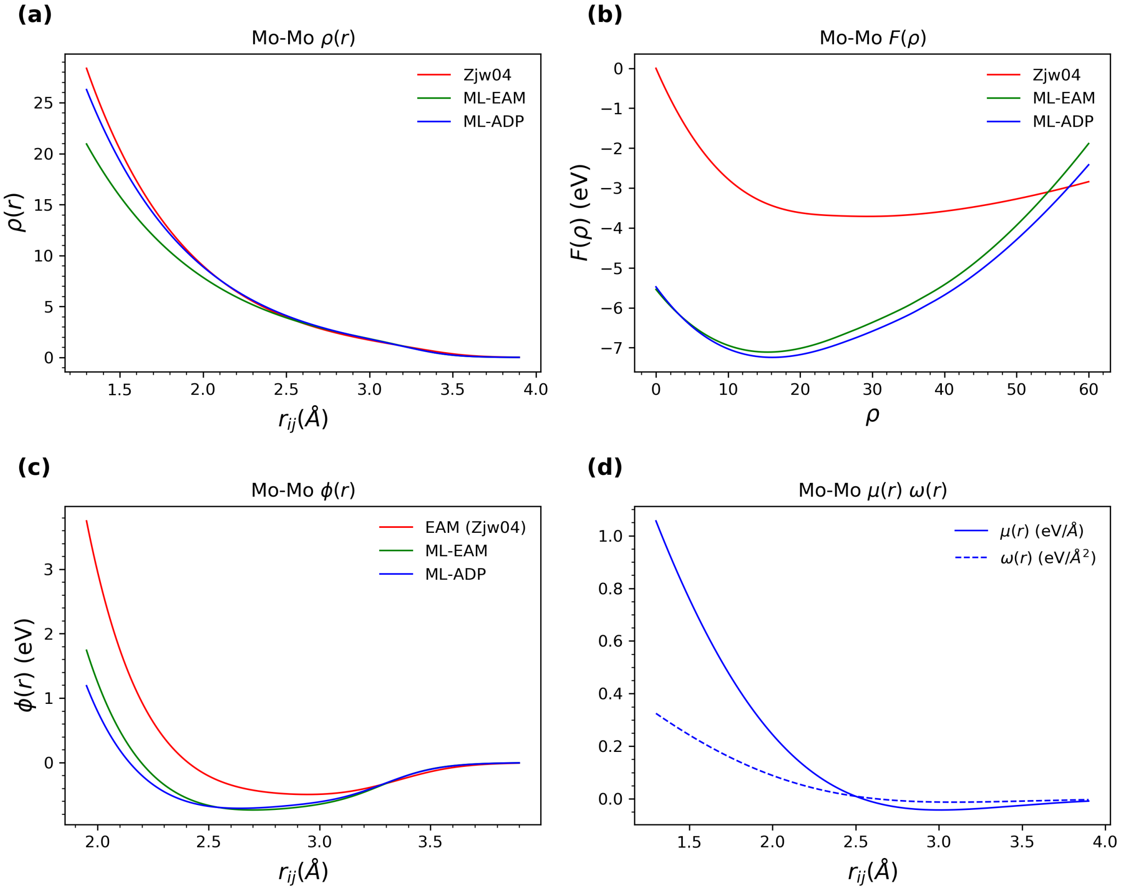

FIG. 3. $\rho(r)$, $F(\rho)$, $\phi(r)$ of the original Zjw04 EAM, ML-EAM and ML-ADP. **(d)** shows the $\mu(r)$ and $\omega(r)$ of ML-ADP.

	DFT	SNAP	EAM	ML-EAM	ML-ADP	Exp.
T_m(K)		3000	3750	2940	2880	2890
c_{11} (GPa)	472	473 (0.2%)	456 (-3.4%)	260 (-5.8%)	274 (-0.7%)	479
c_{12} (GPa)	158	152 (-3.8%)	167 (5.7%)	151 (0.0%)	163 (2.5%)	165
c_{44} (GPa)	106	107 (0.9%)	115 (8.5%)	131 (0.0%)	131 (-0.8%)	108
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

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

	Model	Mo	Ni_4Mo	Ni_3Mo	Mo_{Ni}	Ni_{Mo}	Ni	Overall
Energy (meV/atom)	SNAP	16.2	4.0	5.2	22.7	33.9	7.9	22.5
	EAM	58.9	211.2	255.6	46.5	147.6	10.6	117.2
	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	0.29	0.14	0.16	0.13	0.55	0.11	0.23
	EAM	0.40	0.20	0.19	0.21	0.57	0.06	0.26
	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 V. 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	EAM	ML-EAM	ML-ADP
Ni_3Mo					
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 (13.7%)	272 (107.6%)	214 (63.3%)
c_{33}	402	408 (-10.4%)	295 (13.7%)	474 (17.9%)	398 (-0.5%)
c_{44}	94	84 (-10.4%)	36 (13.7%)	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_4Mo					
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 VI. Comparion 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_3Mo and Ni_4Mo.