Machine learning enhanced empirical potentials for metals and alloys

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

[a]Institute of Applied Physics and Computational Mathematics, Beijing 100088, China
[b]CAEP Software Center for High Performance Numerical Simulation, Beijing 100088, China

Abstract

Interatomic potential (i.e. force-field) plays a vital role in atomistic simulation of materials. Empirical potentials like the embedded atom method (EAM) and its variant angular-dependent potential (ADP) have proven successful in modeling many metals qualitatively. In the past few years, machine learning has became a compelling approach for modeling interatomic interactions. Powered by big data and efficient optimizers, machine learning interatomic potentials can generally approximate to the the accuracy of the first-principles calculations based on the quantum mechanics theory. In this works, we successfully developed a route to express EAM and ADP within machine learning framework in highly-vectorizable form and further incorperated several physical constraints into the training. As it is proved in this work, the performances of empirical potentials can be significantly boosted with few traning data. For energy and force predictions, machine learned 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 molecular dynamics simulations performed by our potentials are nearly 1000 times faster. Machine learned EAM and ADP can also reproduce some key materials properties, such as elastic constants, melting temperatures and surface energies, close to the first-principles accuracy. Our results suggest a new and systematic route for developing machine learning interatomic potentials. All the new algorithms have been implemented in our program TensorAlloy.

Keywords: Machine Learning, Alloy, Materials Modeling, Computational Physics, Empirical Potentials

Corresponding author. E-mail address: lindeye0716@163.com, song_haifeng@iapcm.ac.cn

1. 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, 2, 3, 4, 5], modified embedded-atom method (MEAM) [6, 7, 8, 9], bond-order potential (BoP) [10, 11, 12, 13], or angular-dependent potential (ADP) [14, 15, 16, 17, 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, global 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, 24, 25, 26, 27], is still the most popular choice in modeling metallic interactions [28, 29, 30]. The smooth overlap atomic positions descriptor based gaussian approximation potential (SOAP-GAP) [31, 32, 33, 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, 37, 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 requisite. 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 [37, 28, 29, 30, 34].

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 Tensor-Flow 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 2 describes the theoretical background of this work, including the formalism of the embedded atom method and algorithms of the machine learned EAM. Section 3 describes the implementation details. Section 4 summarizes the results of ML-EAM and ML-ADP. Further discussions are given in section 4.3.

All the new algorithms have been implemented in our program TensorAlloy [42].

2. Method

2.1. Theory

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

$$E^{total} = \sum_i^N E_i$$

$$= \sum_i^N F_a(\rho_i) + \frac{1}{2} \sum_i^N \sum_{j \neq i}^{r_{ij} < r_c} \phi_{ab}(r_{ij}) \tag{1}$$

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 [43] while Mishin developed the angular-dependent potential (ADP) [14, 15, 16, 17, 44]. The ADP formalism introduces three additional terms to the total energy:

$$E^{total} = E^{EAM}$$

$$+ \frac{1}{2} \sum_i \sum_\alpha (\mu_i^\alpha)^2$$

$$+ \frac{1}{2} \sum_i \sum_\alpha \sum_\beta (\lambda_i^{\alpha\beta})^2$$

$$- \frac{1}{6} \sum_i \nu_i^2 \tag{3}$$

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.

2.2. Transformation

To integrate EAM/ADP with machine learning strategy, 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:

$$E_i^A = \frac{1}{2} \sum_{j \neq i}^{N_i^{AA}} \phi_{AA}(r_{ij}) + \frac{1}{2} \sum_{j \neq i}^{N_i^{AB}} \phi_{AB}(r_{ij})$$
$$+ F_A \left(\sum_{j \neq i}^{N_i^{AA}} \rho_A(r_{ij}) + \sum_{j \neq i}^{N_i^{AB}} \rho_B(r_{ij}) \right) \tag{7}$$

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.

$$E_j^B = \frac{1}{2} \sum_{i \neq j}^{N_j^{BB}} \phi_{BB}(r_{ij}) + \frac{1}{2} \sum_{i \neq j}^{N_j^{BA}} \phi_{AB}(r_{ij})$$
$$+ F_B \left(\sum_{j \neq i}^{N_j^{BB}} \rho_B(r_{ij}) + \sum_{j \neq i}^{N_j^{BA}} \rho_A(r_{ij}) \right) \tag{8}$$

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 of structure i. Hence, we can pre-determine the maximum neighor list size N^{nl}:

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

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:

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

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

We can write Equation 7 in an equivalent expression:

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

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

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

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

$$\mathbf{g}_i = \begin{bmatrix} \vec{\mathbf{r}}_i^{AA} & \vec{\mathbf{r}}_i^{AB} & \vec{\mathbf{r}}_i^{BB} & \vec{\mathbf{r}}_i^{BA} \end{bmatrix}^T \tag{14}$$
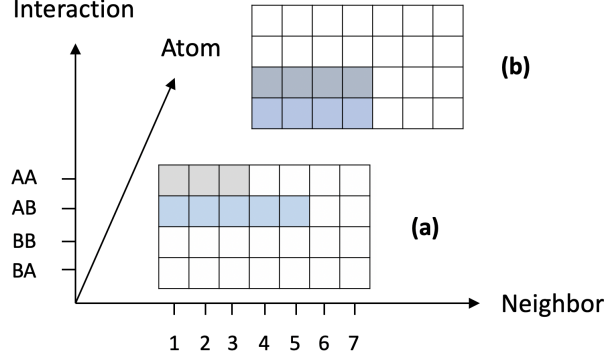
Figure 1: Visualization of the descriptor matrix $\mathbf{G}'$ of the binary AB system and N^{nl} is 7. Non-colored cells are padded zeros. (a) is a sample $\mathbf{g}_i$ of A-type atom and (b) represents a B-type atom.

$\mathbf{g}_i$ is a $4 \times N^{\mathrm{nl}}$ 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 $N \times 4 \times N^{\mathrm{nl}}$.

During the training phase, the maximum appearances of element A and B in any structure ($N_{\mathrm{A}}^{\mathrm{max}}$ and $N_{\mathrm{B}}^{\mathrm{max}}$) are also constants. Thus, any $\mathbf{G}$ can be expanded to a $(N_{\mathrm{A}}^{\mathrm{max}} + N_{\mathrm{B}}^{\mathrm{max}}) \times 4 \times N^{\mathrm{nl}}$ 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 α, β, γ components of $\vec{r}_i$.

Fig 1 is the visualization of the descriptor matrix $\mathbf{G}'$. $\mathbf{G}'$ has three axes: the neighbor axis, the interaction axis and the atom axis. N^{nl} is the maximum of the neighbor axis (in this case $N^{\mathrm{nl}} = 7$). For the binary AB system there are four types of interactions: AA, AB, BB and BA. (a) and (b) are samples of $\mathbf{g}_i$ for A-type atoms and B-type atoms.

2.3. Functions

In this work, we use the EAM potential published by Zhou, Johnson and Wadley (Zjw04) [45, 46, 47] 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:

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

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:

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

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, the interpolation form is used:

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

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

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

5

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:

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

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

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

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

c_1 and c_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:

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

2.4. 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) [48, 49, 50] 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

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

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

$$
\alpha = \sqrt{-\frac{9V_0 B}{E_0}} \tag{27}
$$

where V_0 is the equilibrium volume and B is the corresponding 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 $E^{\text{rose}}(x)$ and their corresponding predicted energy $E(x)$:

$$\mathbf{L}^{\text{Rose}} = \sum_{\Omega} \sqrt{\sum_{i} (E(x_i) - E^{\text{rose}}(x_i))^2} \tag{28}$$

Here Ω denotes all selected crystals. In this work, for each included crystal, we use the same choices of x, which determines the fitting region: $x_i = x_0 + \Delta x \cdot i, x_0 = -0.1, N_i^{\max} = 20, \Delta x = 0.01$. (-10% to +10% of the equilibrium volume). In this work, β^{Ni} is 0.005 and β^{Mo} is 0.008.

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 find a straightforward 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 [51]:

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

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

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}:

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

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

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

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.

The adoption of physical constraints in optimizing MLIPs can be very helpful. Without any physical constraint, adding high-quality data points is, perhaps, the only effective way to tune MLIPs for specific purposes. On the contrast, with physical constraints very few training samples are sufficient. As an example, the elementary SNAP potentials (Mo, Ni) were fitted with just hundreds of structures but they performed quite well on elastic constants and surface energies, as in the outer loop [36] of the fitting process such material properties were already used. One must also note that data consistency is important. Physical constraints are just supplementary roles in developing MLIPs, so they are preferred to be measured with first-principle calculations.

3. Implementation

Fig 2 shows the architecture of the TensorAlloy program. The program has two major steps: the first step is calculating the losses of energy, force and stress terms while the second step is measuring the contributions of the physical constraints.

For the first step, the implementation is almost the same with our previous work [42] except the formula of the potential energy. Briefly, this step starts from checking through the training dataset and determine some key constants, including the global symbol list and its size N^{vap}. For arbitrary structure, it has N_{ij} different atom-atom pairs within
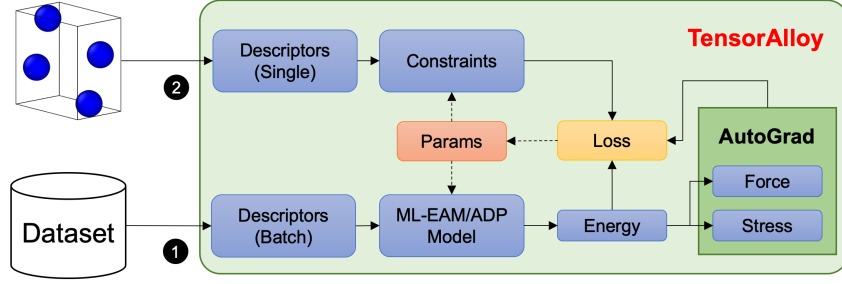
Figure 2: The architecture of TensorAlloy.

cutoff r_c. For a concrete dataset, the corresponding maximum $N_{ij}^{\max}$ can also be determined before training. Hence, Structures of any stoichiometries can be transformed to their corresponding universal GSL forms for batching. The descriptors (2.2) are built upon these GSL arrays. Then the direct computation graph from atomic positions (GSL) to total energy can be easily built. Hence, atomic forces and virial stress tensors can be calculated automatically thanks to the AutoGrad feature of TensorFlow.

For the second step, the situation is quite different. Only a few structures, which may vary significantly, will be used to form the physical constraints. Hence, batching is no longer a concern for this step. Instead, we need a more flexible way to model complicated physical equations. Recall that the virtual-atom approach has two phases: training and prediction [42]. The training phase is designed for mini-batch training. N^{vap}, $N_{ij}^{\max}$ and the GSL are pre-determined constants. On the other hand, the prediction phase targets on a single input structure. N^{vap}, $N_{ij}^{\max}$ and the GSL are determined dynamically. Therefore, the prediction phase is much suitable for modeling physical constraints. In our current implementation, for simplicity, we just generate an independent compuation graph for each type of physical constraint of each selected crystal structure. The model parameters are shared across all the computation graphs (including the graph of step 1). For example, for the Ni-Mo system, if we select the elastic constants of c_{11}, c_{22} and c_{44} of fcc Ni and bcc Mo, and the Rose EOS of fcc Ni and bcc Mo as the physical constraints, there should be 2×2 separate computation graphs (Elastic/fcc/Ni, Elastic/bcc/Mo, Rose/fcc/Ni, Rose/bcc/Mo) under the scope 'Constraint'. The complicated derivatives of the losses with respect to the model parameters can be solved by the AutoGrad module of TensorFlow automatically.

The total loss is just the sum of the two steps introduced above: Equation 34 demonstrates the loss function used in this work:

$$\mathbf{Loss} = \sqrt{\frac{1}{N_b}\sum_{i=1}^{N_b}\left(E_i - E_i^{\mathrm{dft}}\right)^2}$$
$$+ \chi_{\mathrm{f}}\sqrt{\frac{1}{3\sum_i^{N_b} N_i}\sum_i^{N_b}\sum_j^{N_i}\sum_\alpha\left(f_{ij\alpha} - f_{ij\alpha}^{\mathrm{dft}}\right)^2}$$
$$+ \chi_{\mathrm{rose}}\mathbf{L}^{\mathrm{Rose}} + \chi_{\mathrm{elastic}}\mathbf{L}^{\mathrm{elastic}} \tag{34}$$

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. Typicall χ_{f} is 2, χ_{rose} is 10 and χ_{elastic} is set to 0.01.

To find optimal parameters of the funcitons in section 2.3, 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. The Adam optimizer [52] is used to minimize Equation 34. 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 compared with neural network potentials, the optimization typically needs very few training steps to converge. When the optimization is finished, the corresponding Lammps setfl potential file will be exported.
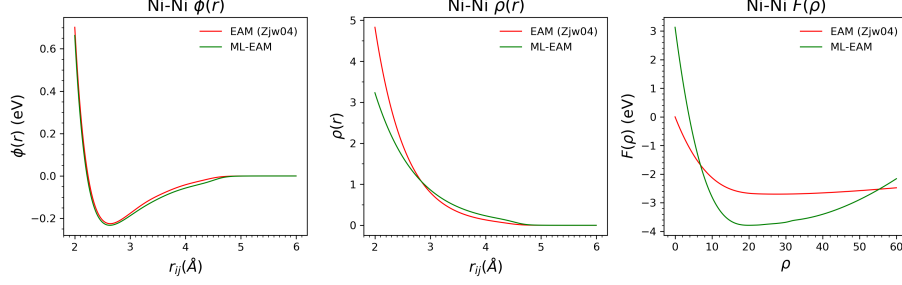
Figure 3: $\rho(r)$, $F(\rho)$ and $\phi(r)$ of the original Zjw04 EAM and the machine learned EAM (ML-EAM).

		EAM[47]	SNAP[37]	ML-EAM	ML-ADP
Energy	Ni	10.6	1.2	3.9	
(meV/atom)	Mo	58.9	13.2	23.8	18.7
Force	Ni	0.06	0.05	0.05	
(eV/Å)	Mo	0.31	0.25	0.30	0.29

Table 1: Comparisons of the energy and force MAEs of elementary Ni and Mo.

	DFT	SNAP [37]	EAM	ML-EAM	Exp.
T_m(K)		1785	1520 [47]	1520	1728
c_{11} (GPa)	276	276 (0.0%)	248 (-10.1%)	274 (-0.7%)	261 [53]
c_{12} (GPa)	159	159 (0.0%)	147 (-7.5%)	163 (2.5%)	151 [53]
c_{44} (GPa)	132	132 (0.0%)	125 (-5.3%)	131 (-0.8%)	132 [53]
B_{VRH} (GPa)	198	198 (0.0%)	181 (-8.6%)	195 (-1.5%)	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 [54]
E_m (eV)	1.12	1.07 (-4.5%)	0.90 (-19.6%)	0.87 (-22.3%)	1.01–1.48 [54]
$E_a = E_v + E_m$ (eV)	2.58	2.75 (6.6%)	2.58 (0.0%)	2.58 (0.0%)	2.77–2.95 [54]

Table 2: 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.
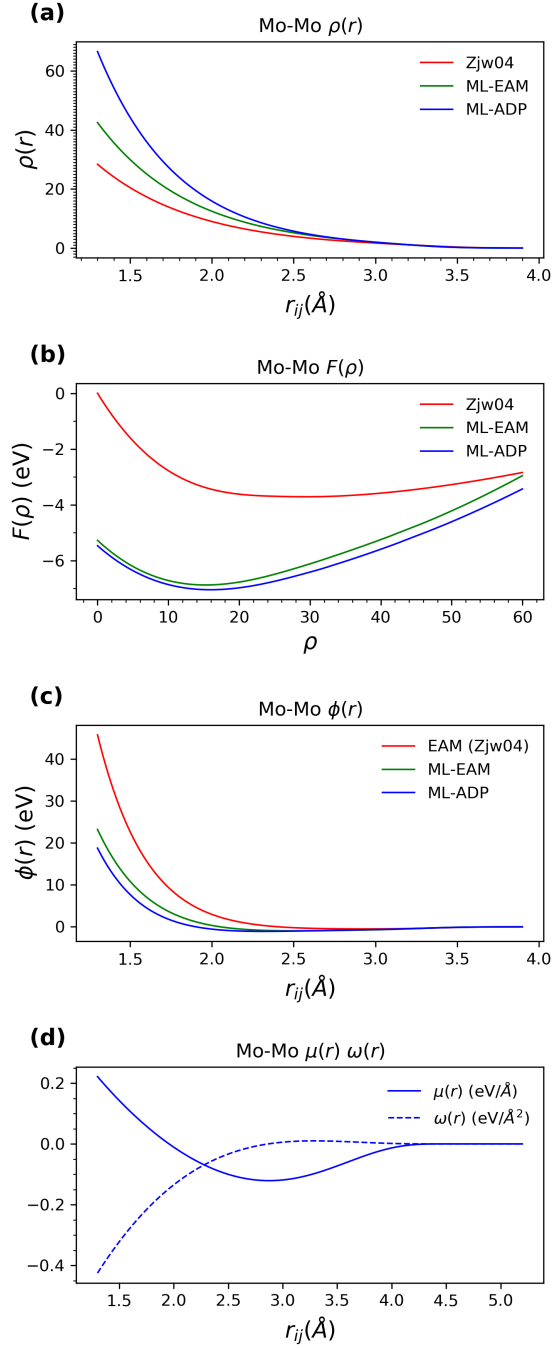
Figure 4: $\rho(r)$, $F(\rho)$, $\phi(r)$ of the original Zjw04 EAM, ML-EAM and ML-ADP for elementary Mo. **(d)** shows the $\mu(r)$ and $\omega(r)$ of ML-ADP.
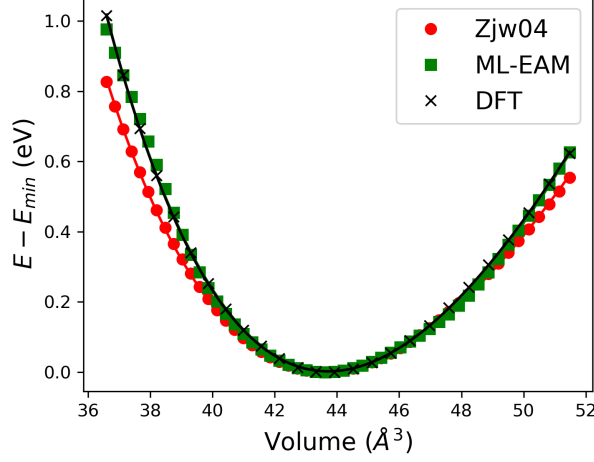
Figure 5: The energy-volume curves of the original EAM, ML-EAM and DFT for the fcc Ni.

4. Results

The public Ni-Mo dataset [37] is used to evaluate our program. This dataset is provided by Shyue Ping Ong and co-workers along with their SNAP models. It contains 3973 different Ni-Mo solids. All calculations were done by VASP [55] with the PBE [56] functional and the projector augmented-wave approach [57].

The optimized parameters are listed in the appendix.

4.1. Elementary Ni and Mo

We firstly use the two elementary datasets (Ni and Mo) to show our approach. These two datasets has 461 and 284 structures, respectively. 61 and 34 structures were randomly selected to form the test subsets. In these two experiments, χ_f, χ_{rose} and $\chi_{elastic}$ 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 MAE:

$$\mathbf{rMAE} = \frac{1}{N} \sum_{i=1}^{N} |\frac{(E_i - E_i^{dft})}{N_i} - (E_{eq} - E_{eq}^{dft})| \tag{35}$$

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 3 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.

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 eV/Åon the entire dataset. ML-EAM is nearly as accurate as SNAP (1.2 meV/atom, 0.05 eV/Å), but later has almost three orders of magnitude more computational expense. Figure 3 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 ∼ 59 meV/atom and the force MAE is more than 0.31 eV/Å. Our method can significantly reduce the error: ML-EAM can achieve test MAEs of 23.8 meV/atom and 0.26 eV/Åand overall MAEs of 26.7 meV/atom and 0.30 eV/Å. We also optimized an ADP potential for this dataset. Interestingly, ML-ADP gets better results: 20.6
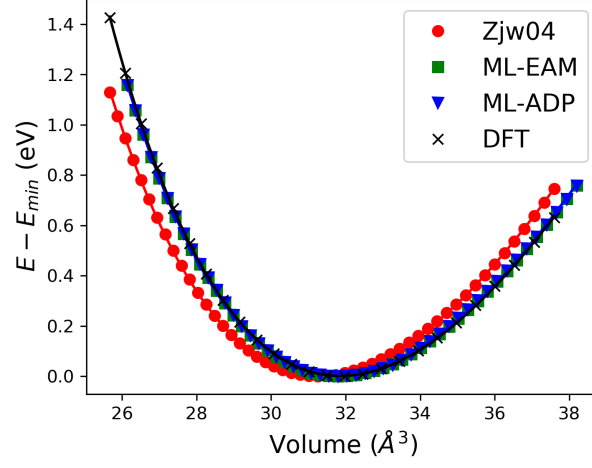
Figure 6: The energy-volume curves of the original EAM, ML-EAM, ML-ADP and DFT for the bcc Mo.
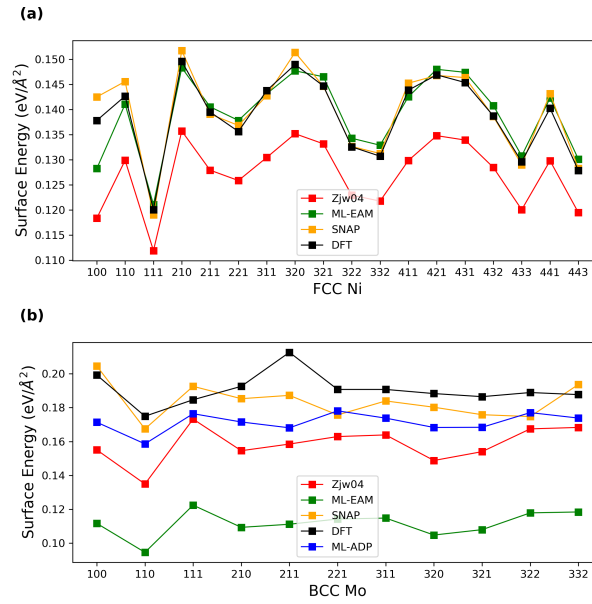

Figure 7: Comparison of calculated surface energies of **(a)** fcc Ni and **(b)** bcc Mo.

	DFT	SNAP [36]	EAM [47]	ML-EAM	ML-ADP	Exp.
T_m(K)		3000	3750	2840	2910	2890
c_{11} (GPa)	472	473 (0.2%)	457 (-3.2%)	463 (-1.9%)	469 (-0.6%)	479 [58]
c_{12} (GPa)	158	152 (-3.8%)	168 (6.3%)	153 (-3.2%)	159 (0.6%)	165 [58]
c_{44} (GPa)	106	107 (0.9%)	116 (9.4%)	98 (-7.5%)	102 (-3.8%)	108 [58]
B_{VRH} (GPa)	263	259 (-1.5%)	264 (0.4%)	256 (-2.7%)	262 (-0.4%)	270 [58]
G_{VRH} (GPa)	124	126 (1.6%)	127 (2.4%)	118 (-4.8%)	121 (-2.4%)	125 [58]
μ	0.30	0.29 (-3.3%)	0.29 (-3.3%)	0.30 (0.0%)	0.30 (0.0%)	0.30
E_v (eV)	2.87	2.61 (-9.1%)	3.02 (5.2%)	1.96 (-31.7%)	2.51 (-12.5%)	
E_m (eV)	1.12	1.39 (24.1%)	1.54 (37.5%)	1.33 (18.7%)	1.06 (-5.3%)	
$E_a = E_v + E_m$ (eV)	3.99	4.00 (-0.1%)	4.56 (14.3%)	3.29 (-17.5%)	3.57 (-10.5%)	4.00 [59]

Table 3: Comparison of the calculated and experimental material properties of elementary bcc Mo. SNAP refers to the elementary SNAP Mo model.
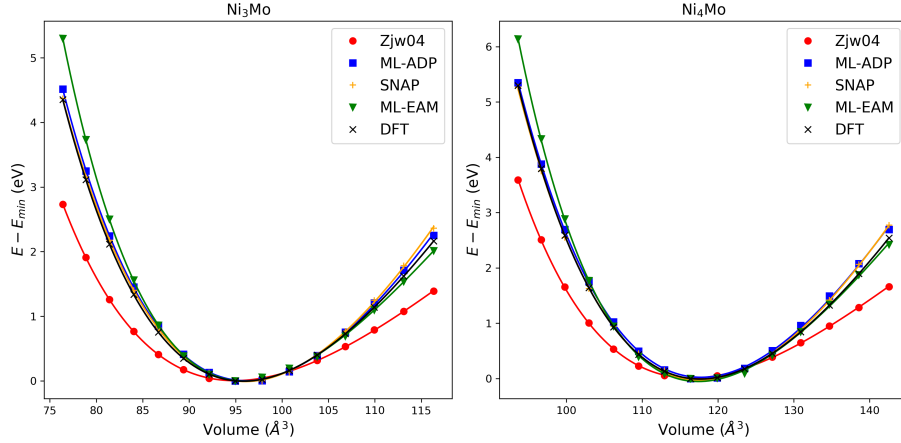

Figure 8: Energy-volume curves of Ni$_3$Mo and Ni$_4$Mo obtained with the original Zjw04 EAM, ML-EAM, ML-ADP, SNAP and DFT.

meV/atom and 0.24 eVÅon the test set and 18.7 meV/atom and 0.29 eV/Åon the entire dataset. On the contrast, the SNAP method (13.2 meV/atom, 0.25 eV/Å) behaves slightly better than ML-ADP. But considering the computational cost, the performace of ML-ADP is totally acceptable. Figure 4 demonstrates the potential functions. The $\rho(r)$, $F(\rho)$ and $\phi(r)$ functions of ML-EAM and ML-ADP are quite similar. The additional accuracy of ML-ADP mainly comes from the dipole function $\mu(r)$.

Figure 5 and 6 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 2 and 3 summerize 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. For the fcc Ni, the machine-learned EAM has gained better performances on predicting elastic constants. For the bcc Mo, ML-ADP not only gives quality predictions on elastic properties but also significantly reduces the error of predicting E_m. We also tested these models with surface energy, as shown in Fig 7. One must note that all the slab structures are relaxed independently with corresponding potentials. For the fcc Ni, ML-EAM can give almost exactly the same results with DFT. For the more complicated bcc Mo, the central-force ML-EAM model becomes not that good due to the lacking of directional d bonding [14]. The angular correction, introduced by Mishin, shows its importance. ML-ADP can give comparable results to the SNAP method.

13

	Model	Mo	Ni_4Mo	Ni_3Mo	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 4: Comparion of the MAEs in predicted energies (mev/atom) and forces (eV/Å) relative to the DFT on the subsets (Mo, Ni_4Mo, Ni_3Mo, Ni-doped Mo, Mo-doped Ni, Ni) and the entire Mo-Ni dataset.

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 supercells $30 \times 15 \times 15$ were used. 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) while both ML-EAM and ML-ADP can give quite accurate melting temperature.

4.2. Mo-Ni

Secondly, we tested our method on the entire binary Mo-Ni dataset. 300 structures were randomly chosen to be the test dataset and the rest 3673 were used to train the potentials. χ_f, χ_{rose} and $\chi_{elastic}$ 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 combined 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 4 summarizes the energy and force MAEs of SNAP, ML-EAM and ML-ADP on the different subsets. 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 but 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_3Mo, Ni-doped Mo and especially the Mo-doped Ni). On the contrast, ML-EAM still performs significantly worse compared with ML-ADP due to the restriction of the central-force model, as discussed before. Interestingly, our results suggest that Mo-Ni dipole and quadrupole contributions 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 8 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 in wide range. The ML-ADP curves agrees perfectly with the DFT curves. ML-EAM agrees well under tensile strains but overestimates the energy under compress strains.

Table 5 summarizes the elastic constants predictions of Ni_3Mo and Ni_4Mo. One should note that these two crystals were not included in the 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.

4.3. 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 using exactly the same dataset. 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 [60].

	DFT	SNAP [37]	EAM [47]	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 5: 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, bcc Mo and binary alloys Ni$_3$Mo and Ni$_4$Mo.

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 $k \leq N$:

$$E = \sum_i^{C_1^N} E_i^{(1)} + \sum_{i,j}^{C_2^N} E_{ij}^{(2)} + \sum_{i,j,k}^{C_3^N} E_{ijk}^{(3)} + \cdots \tag{36}$$

where C_k^N (k=1,2,3) 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 36 can be further transformed to:

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

$$= \sum_i^N E_i \tag{37}$$

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. For the EAM method, the embedding contribution $F(\rho)$ serves as a "special" one-body term because the many-body effect is considered. 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 37. For the SNAP method, the atomic energy is calculated with the following equation:

$$E_i = \beta_0^{\alpha_i} + \sum_k \beta_k^{\alpha_i} B_k^i \tag{38}$$

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 [35] 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. The central-force EAM works well on fcc metals. For bcc metals with partially filled d bands, the directional d bonding cannot be ignored. Hence, the pairwise dipole and quadrupole functions introduced by ADP are necessary.

As we analysed in Section 2.2, the fundamental atomic descriptor of EAM, the $N \times N^{\mathrm{nl}} \times 4$ 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} for the binary SNAP is 4.6 Å. But such descriptor is still too rough to describe complicated potential energy surfaces (e.g. bcc Mo). To further enhance EAM/ADP, complicated three–atoms interactions may be necessary.

5. Conclusions

To conclude, we have successfully combined the training of empirical potentials with the machine learning approach into our TensorAlloy framework. The machine learning approaches (big data, optimization, etc), together with physical constraints (Rose EOS, elastic constants) can significantly improve 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 the SNAP method, while later method is about three 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.

Data Availability

The optimized parameters of the associated EAM and ADP models are all listed in the appendix. The learning datasets are originally created by Shyue Ping Ong and can be accessed from GitHub freely.

Author Contributions

D.Y.L and H.F.S contributed the central conceptions of this study, designed the research project, conducted the analyses and revised the manuscript. X.C. designed the research project, wrote the related codes, trained the machine learning interatomic potentials and conducted the analyses. L.F.W performed molecular dynamics simulations and validated our models. X.Y.G., Y.F.Z. and W.D.C: provided valuable comments and suggestions to the work. X.C. and L.F.W. wrote the initial draft of the paper. All authors contributed to discussions of the results.

Additional Information

Competing interests: The authors declare no Competing Financial or Non-Financial Interests.

References

[1] M. S. Daw, M. I. Baskes, Semiempirical, quantum mechanical calculation of hydrogen embrittlement in metals, Phys. Rev. Lett 50 (17) (1983) 1285.

[2] M. S. Daw, M. I. Baskes, Embedded-atom method: Derivation and application to impurities, surfaces and other defects in metals, Phys. Rev. B 29 (12) (1984) 6443.

[3] M. I. Baskes, J. S. Nelson, A. F. Wright, Semiempirical modified embedded-atom potentials for silicon and germanium, Phys. Rev. B 40 (9) (1989) 6085–6100. doi:10.1103/PhysRevB.40.6085.

[4] R. A. Johnson, Alloy models with the embedded-atom method, Phys. Rev. B 39 (17) (1989) 12554–12559. `doi:10.1103/PhysRevB.39.12554`.

[5] M. I. Baskes, Application of the embedded-atom method to covalent materials: A semiempirical potential for silicon, Phys. Rev. Lett 59 (23) (1987) 2666–2669.

[6] M. I. Baskes, Modified embedded-atom potentials for cubic materials and impurities, Phys. Rev. B 46 (5) (1992) 2727–2742.

[7] M. I. Baskes, R. A. Johnson, Modified embedded atom potentials for hcp metals, Modelling Simul. Mater. Sci. Eng. 2 (1994) 147–163.

[8] B.-J. Lee, J.-H. Shim, M. I. Baskes, Semiempirical atomic potentials for the fcc metals cu, ag, au, ni, pd, pt, al, and pb based on first and second nearest-neighbor modified embedded atom method, Phys. Rev. B 68 (14) (2003). `doi:10.1103/PhysRevB.68.144112`.

[9] B. Jelinek, S. Groh, M. F. Horstemeyer, J. Houze, S. G. Kim, G. J. Wagner, A. Moitra, M. I. Baskes, Modified embedded atom method potential for al, si, mg, cu, and fe alloys, Phys. Rev. B 85 (24) (2012). `doi:10.1103/PhysRevB.85.245102`.

[10] D. G. Pettifor, M. W. Finnis, D. Nguyen-Manh, D. A. Murdick, X. W. Zhou, H. N. G. Wadley, Analytic bond-order potentials for multicomponent systems, Mater. Sci. Eng. A 365 (1-2) (2004) 2–13. `doi:10.1016/j.msea.2003.09.001`.

[11] D. A. Murdick, X. W. Zhou, H. N. G. Wadley, D. Nguyen-Manh, R. Drautz, D. G. Pettifor, Analytic bond-order potential for the gallium arsenide system, Phys. Rev. B 73 (4) (2006). `doi:10.1103/PhysRevB.73.045206`.

[12] B. A. Gillespie, X. W. Zhou, D. A. Murdick, H. N. G. Wadley, R. Drautz, D. G. Pettifor, Bond-order potential for silicon, Phys. Rev. B 75 (15) (2007). `doi:10.1103/PhysRevB.75.155207`.

[13] D. K. Ward, X. W. Zhou, B. M. Wong, F. P. Doty, J. A. Zimmerman, Analytical bond-order potential for the cadmium telluride binary system, Phys. Rev. B 85 (11) (2012). `doi:10.1103/PhysRevB.85.115206`.

[14] Y. Mishin, A. Y. Lozovoi, Angular-dependent interatomic potential for tantalum, Acta Mater. 54 (19) (2006) 5013–5026. `doi:10.1016/j.actamat.2006.06.034`.

[15] A. Hashibon, A. Y. Lozovoi, Y. Mishin, C. Elsässer, P. Gumbsch, Interatomic potential for the cu-ta system and its application to surface wetting and dewetting, Phys. Rev. B 77 (9) (2008). `doi:10.1103/PhysRevB.77.094131`.

[16] F. Apostol, Y. Mishin, Angular-dependent interatomic potential for the aluminum-hydrogen system, Phys. Rev. B 82 (14) (2010). `doi:10.1103/PhysRevB.82.144115`.

[17] F. Apostol, Y. Mishin, Interatomic potential for the al-cu system, Phys. Rev. B 83 (5) (2011). `doi:10.1103/PhysRevB.83.054116`.

[18] Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, Phase stability in the fe–ni system: Investigation by first-principles calculations and atomistic simulations, Acta Mater. 53 (15) (2005) 4029–4041. `doi:10.1016/j.actamat.2005.05.001`.

[19] D. J. Wales, J. P. K. Doye, Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms, J. Phys. Chem. A 101 (28) (1997) 5111–5116. `doi:10.1021/jp970984n`.
URL `http://dx.doi.org/10.1021/jp970984nhttp://pubs.acs.org/doi/pdfplus/10.1021/jp970984n`

[20] D. J. Wales, H. A. Scheraga, Global optimization of clusters, crystals, and biomolecules, Science 285 (5432) (1999) 1368–1372. `doi:10.1126/science.285.5432.1368`.
URL `http://science.sciencemag.org/content/sci/285/5432/1368.full.pdf`

[21] E. Jones, T. Oliphant, P. Peterson, et al., SciPy: Open source scientific tools for Python, [Online; accessed ¡today¿] (2001–).
URL `http://www.scipy.org/`

[22] H. Mühlenbein, M. Schomisch, J. Born, The parallel genetic algorithm as function optimizer, Parallel Computing 17 (6) (1991) 619 – 632. `doi:https://doi.org/10.1016/S0167-8191(05)80052-3`.
URL `http://www.sciencedirect.com/science/article/pii/S0167819105800523`

[23] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett 98 (14) (2007) 146401. `doi:10.1103/PhysRevLett.98.146401`.
URL `https://www.ncbi.nlm.nih.gov/pubmed/17501293`

[24] J. Behler, Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations, Phys. Chem. Chem. Phys 13 (2011) 17930–17955. `doi:0.1039/c1cp21668f`.

[25] J. Behler, Atom-centered symmetry functions for constructing high-dimensional neural network potentials, J. Chem. Phys 134 (7) (2011) 074106. `doi:10.1063/1.3553717`.
URL `https://www.ncbi.nlm.nih.gov/pubmed/21341827`

[26] J. Behler, Representing potential energy surfaces by high-dimensional neural network potentials, J. Phys. Condens. Matter 26 (18) (2014) 183001. `doi:10.1088/0953-8984/26/18/183001`.
URL `https://www.ncbi.nlm.nih.gov/pubmed/24758952`

[27] J. Behler, Constructing high-dimensional neural network potentials: A tutorial review, Int. J. Quantum Chem. 115 (16) (2015) 1032–1050. `doi:10.1002/qua.24890`.

[28] S. Hajinazar, J. Shao, A. N. Kolmogorov, Stratified construction of neural network based interatomic models for multicomponent materials, Phys. Rev. B 95 (1) (2017). `doi:10.1103/PhysRevB.95.014114`.

[29] R. Kobayashi, D. Giofré, T. Junge, M. Ceriotti, W. A. Curtin, Neural network potential for al-mg-si alloys, Phys. Rev. M 1 (5) (2017). `doi:10.1103/PhysRevMaterials.1.053604`.

[30] B. Onat, E. D. Cubuk, B. D. Malone, E. Kaxiras, Implanted neural network potentials: Application to li-si alloys, Phys. Rev. B 97 (9) (2018). `doi:10.1103/PhysRevB.97.094106`.

[31] A. P. Bartok, M. C. Payne, R. Kondor, G. Csanyi, Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons, Phys. Rev. Lett 104 (13) (2010) 136403. `doi:10.1103/PhysRevLett.104.136403`.
URL `https://www.ncbi.nlm.nih.gov/pubmed/20481899`

[32] A. P. Bartók, G. Csányi, Gaussian approximation potentials: A brief tutorial introduction, Int. J. Quantum Chem. 115 (16) (2015) 1051–1057. `doi:10.1002/qua.24927`.

[33] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, M. Ceriotti, Machine learning unifies the modeling of materials and molecules, Sci. Adv 3 (12) (2017). `doi:10.1126/sciadv.1701816`.
URL `http://advances.sciencemag.org/content/advances/3/12/e1701816.full.pdf`

[34] A. P. Bartók, J. Kermode, N. Bernstein, G. Csányi, Machine learning a general-purpose interatomic potential for silicon, Phys. Rev. X 8 (4) (2018). doi:10.1103/PhysRevX.8.041048.

[35] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, G. J. Tucker, Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials, J. Comput. Phys 285 (2015) 316–330. doi:10.1016/j.jcp.2014.12.018.

[36] C. Chen, Z. Deng, R. Tran, H. Tang, I.-H. Chu, S. P. Ong, Accurate force field for molybdenum by machine learning large materials data, Phys. Rev. M 1 (4) (2017). doi:10.1103/PhysRevMaterials.1.043603.

[37] X. G. Li, C. Hu, C. Chen, Z. Deng, J. Luo, S. P. Ong, Quantum-accurate spectral neighbor analysis potential models for ni-mo binary alloys and fcc metals, Phys. Rev. B. 98 (2018) 094104. doi:10.1103/PhysRevB.98.094104.

[38] M. A. Wood, M. A. Cusentino, B. D. Wirth, A. P. Thompson, Data-driven material models for atomistic simulation, Phys. Rev. B 99 (2019) 184305. doi:10.1103/PhysRevB.99.184305.
URL https://link.aps.org/doi/10.1103/PhysRevB.99.184305

[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, X. Zheng, Tensorflow: A system for large-scale machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association, Savannah, GA, 2016, pp. 265–283.
URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.

[41] C. Trott, A. P. Thompson, A brief description of the kokkos implementation of the snap potential in examinimd, Report (2017). doi:10.2172/1409290.

[42] X. Chen, X.-Y. Gao, Y.-F. Zhao, D.-Y. Lin, W.-D. Chu, H.-F. Song, Tensoralloy: An automatic atomistic neural network program for alloys, Computer Physics Communications 250 (2020) 107057. doi:https://doi.org/10.1016/j.cpc.2019.107057.
URL http://www.sciencedirect.com/science/article/pii/S0010465519303820

[43] T. J. Lenosky, B. Sadigh, E. Alonso, V. V. Bulatov, T. D. d. l. Rubia, J. Kim, A. F. Voter, J. D. Kress, Highly optimized empirical potential model of silicon, Modelling Simul. Mater. Sci. Eng. 8 (6) (2000) 825–841. doi:10.1088/0965-0393/8/6/305.

[44] G. P. Purja Pun, K. A. Darling, L. J. Kecskes, Y. Mishin, Angular-dependent interatomic potential for the cu–ta system and its application to structural stability of nano-crystalline alloys, Acta Mater. 100 (2015) 377–391. doi:10.1016/j.actamat.2015.08.052.

[45] H. N. G. Wadley, X. W. Zhou, R. A. Johnson, M. Neurock, Mechanisms models and methods of vapor deposition, Prog. Mater. Sci 46 (2001) 329–377.

[46] 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, T. F. Kelly, Atomic scale structure of sputtered metal multilayers, Acta Mater. 49 (2001) 4005–4015.

[47] X. W. Zhou, R. A. Johnson, H. N. G. Wadley, Misfit-energy-increasing dislocations in vapor-deposited cofe/nife multilayers, Phys. Rev. B 69 (14) (2004). doi:10.1103/PhysRevB.69.144113.

[48] J. H. Rose, J. R. Smith, F. Guinea, J. Ferrante, Universal features of the equation of state of metals, Phys. Rev. B 29 (6) (1984) 2963–2969. doi:10.1103/PhysRevB.29.2963.

[49] R. R. Zope, Y. Mishin, Interatomic potentials for atomistic simulations of the ti-al system, Phys. Rev. B 68 (2) (2003). doi:10.1103/PhysRevB.68.024102.

[50] Y. Mishin, Atomistic modeling of the and -phases of the ni–al system, Acta Mater. 52 (6) (2004) 1451–1467. doi:10.1016/j.actamat.2003.11.026.

[51] N. C. Admal, J. Marian, G. Po, The atomistic representation of first strain-gradient elastic tensors, J. Mech. Phys. Solids 99 (2017) 93–115. doi:10.1016/j.jmps.2016.11.005.

[52] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980 (2014). arXiv:1412.6980.
URL http://arxiv.org/abs/1412.6980

[53] G. Alers, J. Neighbours, H. Sato, Temperature dependent magnetic contributions to the high field elastic constants of nickel and an fe-ni alloy, Journal of Physics and Chemistry of Solids 13 (1) (1960) 40 – 55. doi:https://doi.org/10.1016/0022-3697(60)90125-6.
URL http://www.sciencedirect.com/science/article/pii/0022369760901256

[54] E. H. Megchiche, S. Pérusin, J.-C. Barthelat, C. Mijoule, Density functional calculations of the formation and migration enthalpies of monovacancies in ni: Comparison of local and nonlocal approaches, Phys. Rev. B 74 (2006) 064111. doi:10.1103/PhysRevB.74.064111.
URL https://link.aps.org/doi/10.1103/PhysRevB.74.064111

[55] G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, Phys. Rev. B 54 (1996) 11169–11186. doi:10.1103/PhysRevB.54.11169.
URL https://link.aps.org/doi/10.1103/PhysRevB.54.11169

[56] J. P. Perdew, K. Burke, M. Ernzerhof, Generalized gradient approximation made simple, Phys. Rev. Lett 77 (18) (1996) 3865–3868.
URL https://link.aps.org/doi/10.1103/PhysRevLett.77.3865

[57] P. E. Blöchl, Projector augmented-wave method, Phys. Rev. B 50 (1994) 17953–17979. doi:10.1103/PhysRevB.50.17953.
URL https://link.aps.org/doi/10.1103/PhysRevB.50.17953

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

[59] J. Askill, D. H. Tomlin, Self-diffusion in molybdenum, The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics 8 (90) (1963) 997–1001. arXiv:https://doi.org/10.1080/14786436308214459, doi:10.1080/14786436308214459.
URL https://doi.org/10.1080/14786436308214459

[60] X. Chen, M. S. Jorgensen, J. Li, B. Hammer, Atomic energies from a convolutional neural network, J. Chem. Theory Comput. 14 (7) (2018) 3933–3942. doi:10.1021/acs.jctc.8b00149.
URL https://www.ncbi.nlm.nih.gov/pubmed/29812930