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 traditionally both microscopic observables (energy, forces, virial, etc.) and macroscopic observables (melting point, surface energy, etc.) can be used to tune these parameters.

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 by Parinello and Behler is the most popular choice in modeling metallic interactions. The smooth overlap atomic positions descriptor based gaussian approximation potential (SOAP-GAP) can give extremely accurate results, although it's a bit computationally expensive. Spectral neighbor analysis potential (SNAP) is another quantum-accurate MLIP and it performs very well on a broad range of metals and alloys.

Compared with empirical potentials, MILPs typically have several orders of magnitudes more model parameters. The redundant parameter space greatly reduces the difficulty of fitting complicated potential energy surfaces with MLIPs. However, it also raises some new challenges. To effectively train MLIPs with large numbers of parameters, typically tens of thousands high-quality *ab initio* data points are necessary. The cost of building training datasets can not be ignored. The lack of physical background makes the performances of MILPs outside the training 'zone' not stable. For large-scale alloy systems, computation efficiency becomes a major concern.

Traditionally, to find optimal parameters of an EAM potential, global optimization approaches (such as Basin Hopping or Genetic Algorithm) will be used because the analytical gradients of the total loss with respect to model parameters are difficult to derive and implement *by hand*. In this paper, we chose a new route to address this issue: to 'ask' machine learning platform to handle derivations and implementations.

This paper is organized as follows. Section II describes the theoretical background of this work, including the formalism of the embedded atom method, the expressions of the empirical potential and the algorithms for integrating EAM with machine learning. Section III summarizes the training results and the optimal parameters. Applications of the new potentials are discussed in Section IV. Technical details can be found in the appendix.

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. Typically, 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:

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

The original EAM potential 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.

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 and it is applicable to a number of metals and alloys. In the Zjw04 model, the electron density function has the following form:

$$\rho_b(r) = \frac{f_e \exp\left[-\beta\left(r/r_e - 1\right)\right]}{1 + \left(r/r_e - \lambda\right)^{20}} \tag{8}$$

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:

$$\phi_{aa}(r) = \frac{A\exp\left[-\alpha\left(r/r_e - 1\right)\right]}{1 + \left(r/r_e - \kappa\right)^{20}} - \frac{B\exp\left[-\beta\left(r/r_e - 1\right)\right]}{1 + \left(r/r_e - \lambda\right)^{20}} \tag{9}$$

where A, α and κ are also trainable parameters, B, β and κ are used in Equation 8 already. For the pairwise interaction between two atoms of different species, Zhou et al proposed an interpolation form:

$$\phi_{ab}(r) = \frac{1}{2}\left(\frac{\rho_b(r)}{\rho_a(r)}\phi_{aa}(r) + \frac{\rho_a(r)}{\rho_b(r)}\phi_{bb}(r)\right) \tag{10}$$

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

$$F(\rho) = \begin{cases} \sum_{i=0}^{3} F_{ni}\left(\frac{\rho}{\rho_n} - 1\right)^i & \rho < \rho_n \\ \sum_{i=0}^{3} F_i\left(\frac{\rho}{\rho_e} - 1\right)^i & \rho_n \le \rho < \rho_0 \\ F_e\left[1 - \eta\ln\left(\frac{\rho}{\rho_s}\right)\right](\frac{\rho}{\rho_s})^\eta & \rho_0 \le \rho \end{cases} \tag{11}$$

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 11:

$$\begin{aligned} F(\rho) =& c_1 \cdot \sum_{i=0}^{3} F_{ni}\left(\frac{\rho}{\rho_n} - 1\right)^i \\ &+ c_2 \cdot \sum_{i=0}^{3} F_i\left(\frac{\rho}{\rho_e} - 1\right)^i \\ &+ c_3 \cdot F_e\left[1 - \eta\ln\left(\frac{\rho}{\rho_s}\right)\right](\frac{\rho}{\rho_s})^\eta \end{aligned} \tag{12}$$

$$c_1 = \frac{1}{1 + e^{-2(\rho_n - \rho)}} \tag{13}$$

$$c_3 = \frac{1}{1 + e^{-2(\rho - \rho_0)}} \tag{14}$$

$$c_2 = 1 - c_1 - c_3 \tag{15}$$

The new equation can be minimized without considering about boundary continuity. ω_1 and ω_3 are just damping factors calculated by the sigmoid functions (Equations 13 and 14).

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

$$\mu_{ab}(r) = \left[d_1^{ab}\exp\left(-d_2^{ab} r\right) + d_3^{ab}\right]\psi\left(\frac{r - r_0}{r_h}\right) \tag{16}$$

$$\omega_{ab}(r) = \left[q_1^{ab}\exp\left(-q_2^{ab} r\right) + q_3^{ab}\right]\psi\left(\frac{r - r_0}{r_h}\right) \tag{17}$$

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

$$\psi(x) = \begin{cases} 0 & x \ge 0 \\ \frac{x^4}{1+x^4} & x < 0 \end{cases} \tag{18}$$

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 a binary alloy, AB, to demonstrate how to derive this equation.

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

$$\begin{aligned} E_i^A =& \frac{1}{2}\sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) + \frac{1}{2}\sum_{j\neq i}^{N_i^{\mathrm{AB}}} \phi_{\mathrm{AB}}(r_{ij}) \\ &+ F_{\mathrm{A}}\left(\sum_{j\neq i}^{N_i^{\mathrm{AA}}} \rho_{\mathrm{A}}(r_{ij}) + \sum_{j\neq i}^{N_i^{\mathrm{AB}}} \rho_{\mathrm{B}}(r_{ij})\right) \end{aligned} \tag{19}$$

where N_i^{AA} represents the number of neighors of specie A of atom i and N_i^{AB} represents the number of neighors of specie B. E_i^B has a similar expression. For each atom, we can calculate its N_i^{AA} and N_i^{AB}. Thus, we can find the maximum neighor list size N_{nl}:

$$N_{\mathrm{nl}} = \max\left(\max\left(N_i^{\mathrm{AA}}\right), \max\left(N_i^{\mathrm{AB}}\right), \max\left(N_i^{\mathrm{BB}}\right)\right) \tag{20}$$

N_{nl} is a constant during training because both the dataset and r_c are fixed.

Assume $H(x)$ represents the heaviside step function:

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{21}$$

Then,

$$\begin{aligned} \sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) =& \sum_{j\neq i}^{N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(r_{ij}) \cdot 1 + \sum_{j\neq i}^{N_{\mathrm{nl}}-N_i^{\mathrm{AA}}} \phi_{\mathrm{AA}}(0) \cdot 0 \\ =& \phi_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) \end{aligned} \tag{22}$$

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 padding zeros.

We can write Equation 19 in an equivalent expression:

$$\begin{aligned} E_i^{\mathrm{A}} =& \frac{1}{2}\left(\phi_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) + \phi_{\mathrm{AB}}(\vec{\mathbf{r}}_i^{\mathrm{AB}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AB}})\right) \\ &+ F_{\mathrm{A}}\left(\rho_{\mathrm{AA}}(\vec{\mathbf{r}}_i^{\mathrm{AA}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AA}}) + \rho_{\mathrm{AB}}(\vec{\mathbf{r}}_i^{\mathrm{AB}})^T H(\vec{\mathbf{r}}_i^{\mathrm{AB}})\right) \end{aligned} \tag{23}$$

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}:

$$\begin{aligned} E_j^{\mathrm{B}} =& \frac{1}{2}\left(\phi_{\mathrm{BB}}(\vec{\mathbf{r}}_j^{\mathrm{BB}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BB}}) + \phi_{\mathrm{BA}}(\vec{\mathbf{r}}_j^{\mathrm{BA}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BA}})\right) \\ &+ F_{\mathrm{A}}\left(\rho_{\mathrm{BB}}(\vec{\mathbf{r}}_j^{\mathrm{BB}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BB}}) + \rho_{\mathrm{BA}}(\vec{\mathbf{r}}_j^{\mathrm{BA}})^T H(\vec{\mathbf{r}}_j^{\mathrm{BA}})\right) \end{aligned} \tag{24}$$

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:

$$\mathbf{g}_i = \left[\vec{\mathbf{r}}_i^{\mathrm{AA}} \;\; \vec{\mathbf{r}}_i^{\mathrm{AB}} \;\; \vec{\mathbf{r}}_i^{\mathrm{BB}} \;\; \vec{\mathbf{r}}_i^{\mathrm{BA}}\right] \tag{25}$$

$\mathbf{g}_i$ 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, $\mathbf{G}'$, with zero-padding. As $\mathbf{G}'$ is a fixed-shape matrix and $\rho(r)$, $\phi(r)$ and $F(\rho)$ are all elementwise functions, we can use any modern machine learning platform (TensorFlow, PyTorch, MXNet, etc) to implement the EAM model. In this paper, we chose TensorFlow as our coding framework. Based on the virtual atom approach, we successfully designed a route to build the machine learning computation graph directly from positions to total energy. Thus, analytically-derived EAM force and virial stress can be calculated without pain.

$$f_i = -\frac{\partial E^{total}}{\partial r_i} \tag{26}$$

$$V \cdot \epsilon = -\mathbf{F}^T\mathbf{R} + \left(\frac{\partial E^{total}}{\partial \mathbf{h}}\right)^T \mathbf{h} \tag{27}$$

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. The derivation of Equation 27 has been published elsewhere. The details of the virtual atom approach can be found in the appendix. The last step is to find optimal parameters. This can be done by minimizing the mini-batch loss function:

$$\begin{aligned} \mathbf{Loss} =& \sqrt{\frac{1}{N_b}\sum_{i=1}^{N_b}\left(E_i - E_i^{\mathrm{dft}}\right)^2} \\ &+ \chi_{\mathrm{f}}\sqrt{\frac{1}{3\sum_i^{N_b} N_i}\sum_i^{N_b}\sum_j^{N_i}\sum_\alpha\left(f_{ij\alpha} - f_{ij\alpha}^{\mathrm{dft}}\right)^2} \\ &+ \chi_{\mathrm{s}}\sqrt{\frac{1}{6N_b}\sum_i^{N_b}\sum_j^{6}\left(\epsilon_j^{\mathrm{voigt}} - \epsilon_j^{\mathrm{voigt,dft}}\right)^2} \end{aligned} \tag{28}$$

where N_b is the batch size and N_i is the number of atoms in structure i. The superscript 'voigt' means that stress tensors should be converted to Voigt form. The Adam optimizer is used to minimize Equation 28. In most cases, we use 0.01 as the initial learning rate and the batch size ranges from 20 to 50.

C. Implementation

D. Constraints

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

$$E(y) = E_0\left[1 + \alpha x + \beta\alpha^3 x^3 \frac{2x+3}{(x-1)^2}\right]e^{-\alpha x} \tag{29}$$

is used because the original form tends to underestimate energies under high pressures. In Equation 29, 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

$$\alpha = \sqrt{-\frac{9V_0B}{E_0}} \tag{30}$$

where V_0 is the equilibrium volume and B is the bulk modulus. The adoption of the Rose constraint guarantees the exact prediction of the 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 $E(x)^{\mathrm{Rose}}$ and their corresponding predicted $E(x)$:

$$\mathbf{L}^{\mathrm{Rose}} = \sum_i \mathbf{RMSE}(E_i(x), E_i(x)^{\mathrm{Rose}}) \tag{31}$$

In this work, for each included crystal, we use the same choices of x: $x_t = x_0 + \Delta x \cdot t, x_0 = -0.1, N_t^{\max} = 20, \Delta x = 0.01$.

Elastic constraints are common metrics for tuning empirical potentials. In the ML framework, fitting elastic constants of specific crystals is also possible. The elastic constant c_{ijkl} can be derived from E^{total} directly:

$$c_{ijkl}|_{\epsilon\to 0,\mathbf{F}\to 0} = \frac{1}{V}\left[\left(\frac{\partial \epsilon_{ij}}{\partial \mathbf{h}}\right)^{\mathrm{T}} \mathbf{h}\right]_{kl} \tag{32}$$

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

$$\mathbf{L}^{\mathrm{elastic}} = \sum_i c_i \cdot \mathbf{RMSE_i} + |\epsilon| + |\mathbf{F}| \tag{33}$$

$$c_i = \mathbf{ReLU}(\mathbf{MAE_i} - \tau) \tag{34}$$

$$\mathbf{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{35}$$

where $\sum_i$ loops through all included crystals, **MAE** is the mean absolute error and τ is a pre-selected gate parameter. When **MAE** is below τ, $\mathbf{L}^{\mathrm{elastic}}$ will not contribute to the total loss. In this work, τ is set to 1 GPa.

III. RESULTS AND DISCUSSIONS

The publicly available Ni-Mo dataset is use to demonstrate our ML-EAM approach. This dataset is built by Shyue Ping Ong and co-workers. It contains 3973 unique Ni-Mo solids, including 373 elemental Ni and 284 elemental Mo. DFT calculations were done by VASP at the PBE level with projector augmented-wave approach.

Table I shows the optimization results. All these minimization tasks include energy, force and stress terms in their total losses. The χ_{f} in Equation 28 is set to 10 and χ_{s} is fixe to 80. In this paper, the superscript tag 'old' denotes the original Zjw04, 'res' means restricted optimization and 'unres' means unrestricted optimization. Here 'unrestricted' indicates the parameter r_e will be treated as a common adjustable variable.

Table II demonstrates the prediction errors of the machine learned EAM models and reference models.

IV. DISCUSSIONS

V. CONCLUSIONS

ACKNOWLEDGMENTS

	Ni^{old}	Ni	Ni^{alloy}	Mo^{old}	Mo	Mo^{alloy}
r_e	2.488746	2.124480		2.728100		
f_e	2.007018	2.633256		2.723710		
ρ_e	27.562015	27.233315				
ρ_s	27.930410	26.392414				
α	8.383453	8.452753				
β	4.471175	3.285651				
A	0.429046	0.9802988				
B	0.633531	0.8919016				
κ	0.443599	0.5685785				
λ	0.820658	1.1653832				
F_{n0}	-2.693513	-3.4354472				
F_{n1}	-0.076445	0.3544341				
F_{n2}	0.241442	-2.5563858				
F_{n3}	-2.375626	-7.1984844				
F_0	-2.70	-3.236908				
F_1	0	1.4576268				
F_2	0.265390	2.1785288				
F_3	-0.152856	-1.642411				
η	0.469000	4.305329				
F_e	-2.699486	-3.6163342				

TABLE I. The original (labeled as 'old'), elemental and alloy parameters.

	Model	Mo	Ni_4Mo	Ni_3Mo	Ni_{Mo}	Mo_{Ni}	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	45.0	10.6	7.1	39.0	29.8	10.4	27.4
	NN (esf)	30.0	6.1	9.0	16.6	26.2	7.8	19.1
Force (eV/Å)	SNAP	0.29	0.14	0.16	0.13	0.55	0.11	0.23
	EAM	0.31	0.20	0.19	0.21	0.57	0.06	0.26
	ML-EAM	0.31	0.17	0.15	0.23	0.18	0.09	0.17
	NN (esf)	0.36	0.10	0.11	0.08	0.16	0.06	0.12

TABLE II. Comparion of the MAEs in predicted energies (mev/atom) and forces (eV/Å) relative to DFT.

	DFT	Ni-Mo SNAP	ML-EAM	EAM	Experiment
Mo					
c_{11}	472	475 (0.6%)	477 (1.1%)	457 (-3.2%)	479
c_{12}	158	163 (3.2%)	166 (5.1%)	168 (6.3%)	165
c_{44}	106	111 (4.7%)	101 (-4.7%)	116 (9.4%)	108
B_{VRH}	263	267 (1.5%)	270 (2.7%)	264 (0.4%)	270
G_{VRH}	124	127 (2.4%)	127 (2.4%)	127 (2.4%)	125
μ	0.30	0.29 (-3.3%)	0.31 (3.3%)	0.29 (-3.3%)	0.30
Ni					
c_{11}	276	269 (-2.5%)	268 (-2.9%)	248 (-10.1%)	261
c_{12}	159	150 (-5.7%)	165 (3.8%)	147 (-7.5%)	151
c_{44}	132	135 (2.3%)	116 (-12.1%)	125 (-5.3%)	132
B_{VRH}	198	190 (-4.0%)	200 (1.0%)	181 (-8.6%)	188
G_{VRH}	95	97 (2.1%)	84 (-11.6%)	87 (-8.4%)	479
μ	0.29	0.28 (-3.4%)	0.31 (6.8%)	0.29 (0.0%)	0.29
Ni_3Mo					
c_{11}	385	420 (9.1%)	430 (11.7%)	195 (-49.4%)	
c_{12}	166	197 (18.7%)	187 (12.7%)	98 (-41.0%)	
c_{13}	145	162 (11.7%)	180 (24.1%)	98 (-32.4%)	
c_{23}	131	145 (10.7%)	210 (60.3%)	107 (-18.3%)	
c_{22}	402	360 (-10.4%)	457 (13.7%)	98 (-75.6%)	
c_{33}	402	408 (-1.5%)	412 (2.5%)	295 (-26.6%)	
c_{66}	94	84 (-10.5%)	27 (-71.3%)	36 (-61.7%)	
B_{VRH}	230	243 (5.7%)	280 (21.7%)	156 (-32.2%)	
G_{VRH}	89	100 (12.4%)	66 (-25.8%)	61 (-31.5%)	
μ	0.33	0.32 (-3.0%)	0.39 (18.2%)	0.33 (0.0%)	
Ni_4Mo					
c_{11}	300	283 (-5.7%)	278 (-7.3%)	172 (-42.7%)	
c_{12}	186	179 (-3.8%)	184 (-1.1%)	158 (-15.1%)	
c_{22}	313	326 (4.2%)	278 (-11.2%)	158 (-49.5%)	
c_{23}	166	164 (-1.2%)	230 (38.6%)	80 (-51.8%)	
c_{66}	130	126 (-3.1%)	87 (-33.1%)	125 (-3.8%)	
B_{VRH}	223	220 (-1.3%)	233 (10.5%)	161 (-27.8%)	
G_{VRH}	91	95 (4.4%)	57 (37.4%)	-156 (-162%)	
μ	0.33	0.31 (-6.1%)	0.39 (18.2%)	0.70 (112%)	

TABLE III. 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.