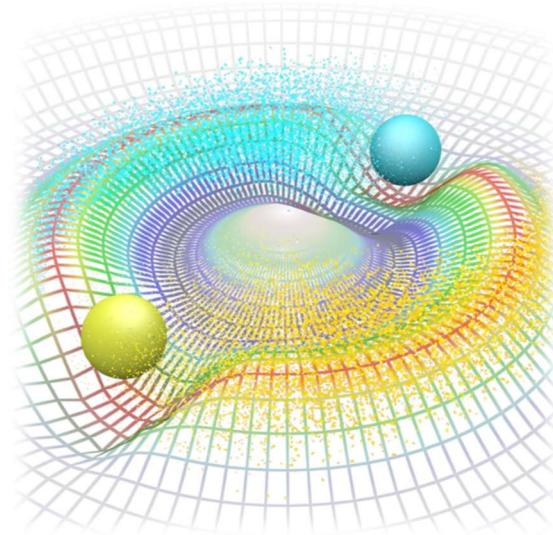


EAPOTs

User Manual



2021-1-1

By B. N. Yao and R. F. Zhang

Beihang University

Contents

1.	Introduction.....	1
1.1.	Overview of EAPOTs	1
1.2.	EAPOTs features.....	3
1.3.	Run EAPOTs.....	3
1.4.	License statement.....	4
2.	EAPOTs framework.....	6
3.	Potential fitting.....	9
3.1.	Choose potential functions.....	11
3.2.	Prepare fitting targets	14
3.3.	Set minimization algorithms	18
3.4.	Add other constraints	21
3.5.	Run optimizations	23
3.6.	Potential fitting task manager	25
4.	Potential presetting.....	27
4.1.	Create atomic structures for DFT calculations	28
4.2.	Parse atomic data from DFT calculations	30
5.	Structure modeling.....	32
6.	Potential plotting	34
6.1.	Snapshot of potential curves	35
6.2.	Potential energy surface	37
7.	Potential Model	39

7.1.	EAM model.....	39
7.2.	FS/TB model	42
7.3.	BOP Tersoff/Brenner model.....	44
7.4.	SW model.....	45
7.5.	MEAM model	46
	Reference.....	49

1. Introduction

1.1. Overview of EAPOTs

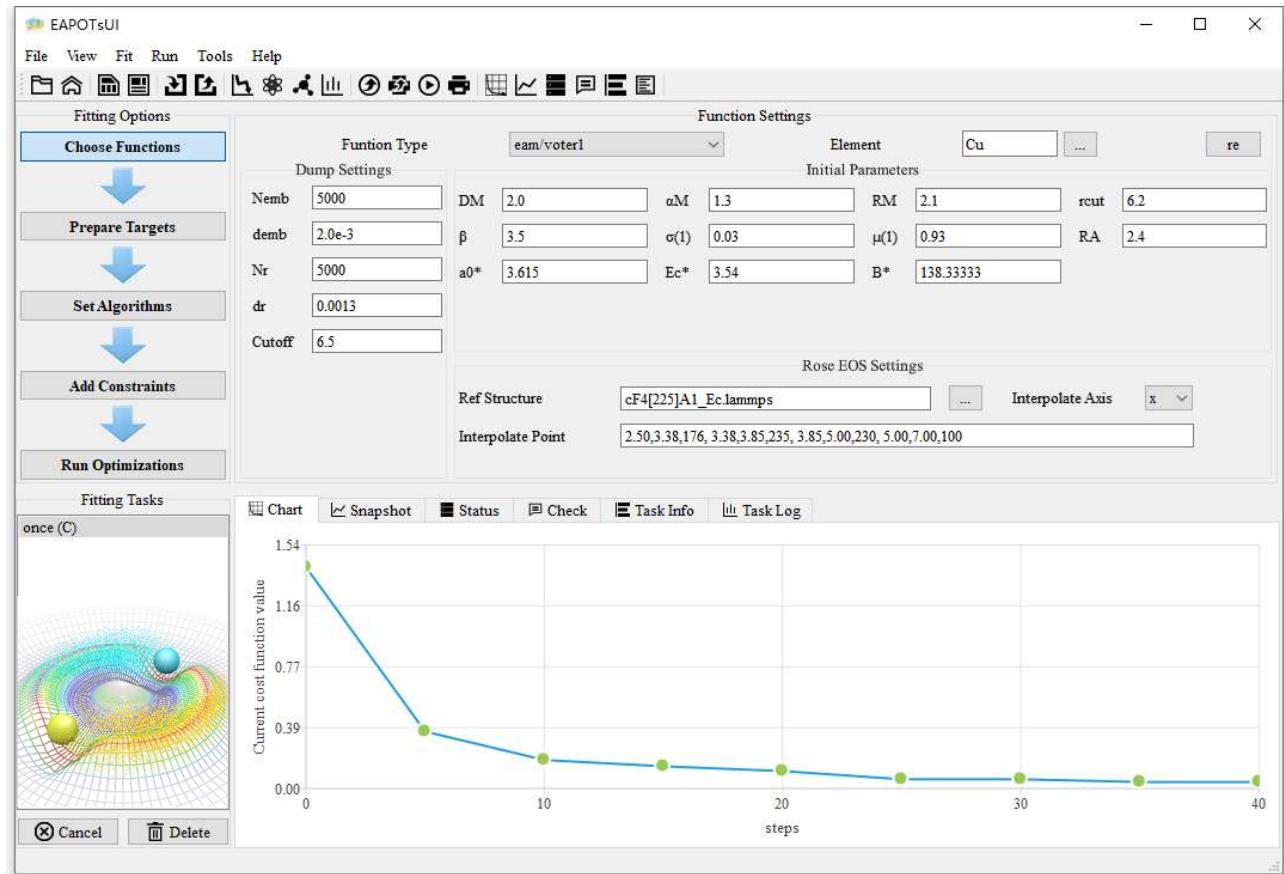


Fig. 1 Screenshot of the main windows of EAPOTs

EAPOT is an integrated software platform developed by the research group of Professor Ruifeng Zhang at Beihang University in recent five years to build empirical interatomic potential. It includes single component eapots [Copyright No.: 2019sr0190574], multi-component eapotc [Copyright No.: 2019sr1395181] and verification system eachk [Copyright No.: 2020sr0748849]. Through the convenient interactive interface, it can realize the choice of various potential function forms, the

setting of fitting target parameters, the integration of various local and global optimization algorithms, and the verification and analysis window, so as to provide users with a whole process integration platform. The software platform not only realizes the high-throughput automatic fitting process based on first principles calculation, but also realizes the multiple combination and multi-level objective optimization scheme of "energy-stress-force-elasticity", which ensures that the fitting strategy can meet the needs of different simulation scenarios. Compared with other similar softwares such as potfit, meamfit, atomicrex, etc., this software exhibits a series of advantages, e.g. more concise control interface, richer function form, more powerful optimization algorithms, and higher integration degree.

EAPOTs is the first module used to construct and check the element potential. The supported potential functions include: Zhou's EAM multi-body potential function, Voter's EAM multi-body potential function, Mishin's EAM multi-body potential function, Finnis FS multi-body potential function, Ackland's FS multi-body potential function, Dai's FS multi-body potential function, Rosato's TB multi-body potential function, Li's TB multi-body potential function, SW three-body potential function, Tersoff's and Brenner's bond order potential function, the MEAM multi-body potential function of Baskes and the second-order MEAM multi-body potential function of Lee. The software platform can run on a single processor or multi task distribution. It is written in highly portable C++ language. It has high running efficiency and easy to expand new functions. It can be built into a library to call EAPOTs or EAPOTc through the library interface.

1.2. EAPOTs features

- a user-friendly GUI, as shown in Fig. 1
- a high-throughput (HT) flowchart for model construction with various modifications and data exchange with first-principles calculations
- multiple combinations of "energy-stress-force-elasticity" and multi-level objective, optimization schemes for different simulation scenarios
- a variety of interatomic potential forms for elemental metallic and covalent solids,
- the custom definition of functional forms via Python script
- extensive training and validation sets for high-efficiency computation

1.3. Run EAPOTs

Please run the program [bin/EAPOTsUI.exe](#) to open EAPOTs. For more information, Table 1 shows the EAPOTs file directory with several sub-directories, and Fig. 2 shows the module hierarchy within the EAPOTs code.

Table 1 Overview of EAPOTs file directory

directory	description
/bin	Potential fitting module
/EACHK	Potential checking module
/config	Program settings
/example	Potential fitting examples
/res	Resource files
/lammmps	LAMMPS (for potential checking) [1]
/struct	Atomic structure files for potential checking
/tasks	Submitted potential fitting tasks
EAPOTs-Manual.pdf	EAPOTs User Manual

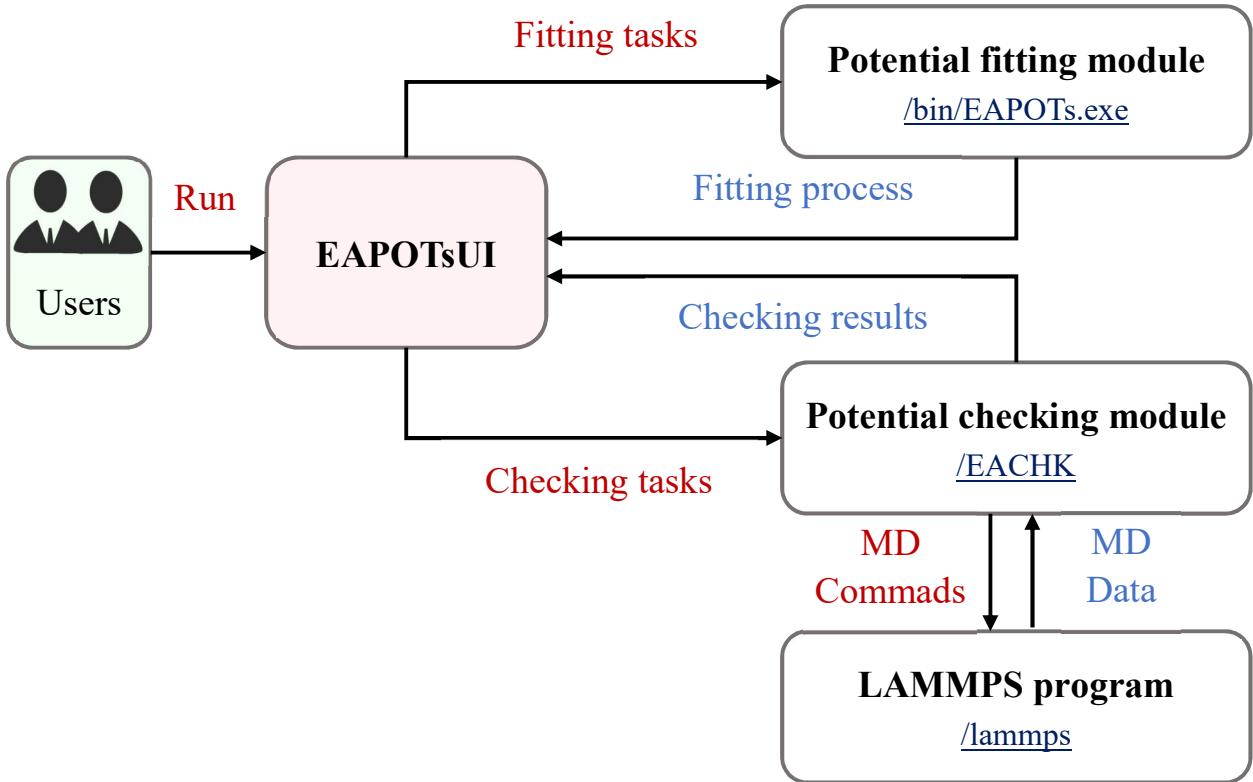


Fig. 2 Module hierarchy within EAPOTs code

1.4. License statement

The EAPOT is currently copyrighted and distributed free of charge for academic, scientific, and educational, and non-commercial users with our permission. Commercial users may also use this software with no cost until a license is established, or with a contract for payment for the technical support. This software is provided ‘as it is’ without any expressed or implied warranty.

The full EAPOTs program package with the companion components may also be obtained upon request (email to zrfcms@buaa.edu.cn) and more details can be found at:

<https://sites.google.com/site/zrbuaa/softwares/eapot-generator>

Registration Form for EAPOT Studio

Name	
Title (e.g. Dr., Prof., Mr., Mrs, Ms)	
Affiliation (e.g. Dept., Univ., Co.)	
Address	
Country / Region	
License type (academic/educational/commercial)	
E-mail	
Fields	

Signature:

Date:

2. EAPOTs framework

There are four top windows in the EAPOTs framework, and they can be activated by the menu or toolbar buttons, as shown in Fig. 3:

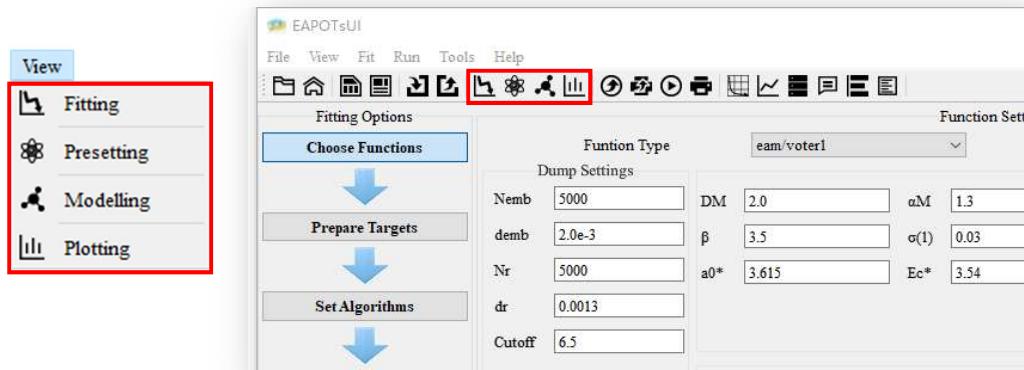


Fig. 3 Top window switch

1. Potential fitting window

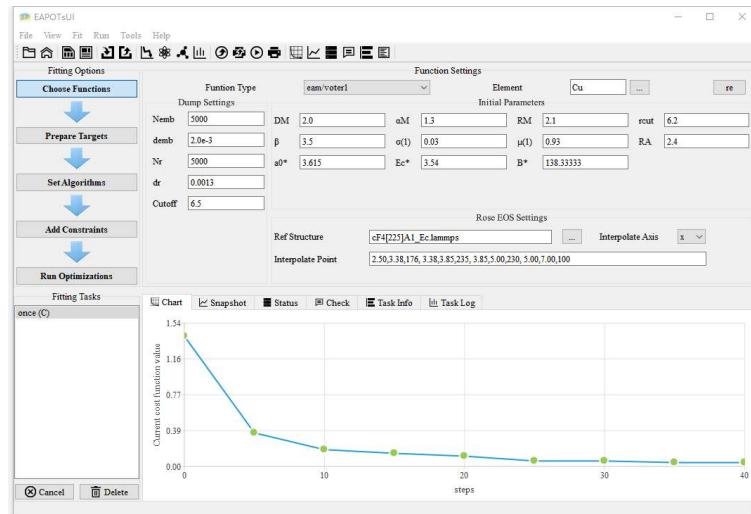


Fig. 4 Screenshot of the potential fitting window

The potential fitting page, as shown in Fig. 4, is mainly used for the potential fitting task creation and submission.

2.Potential presetting window

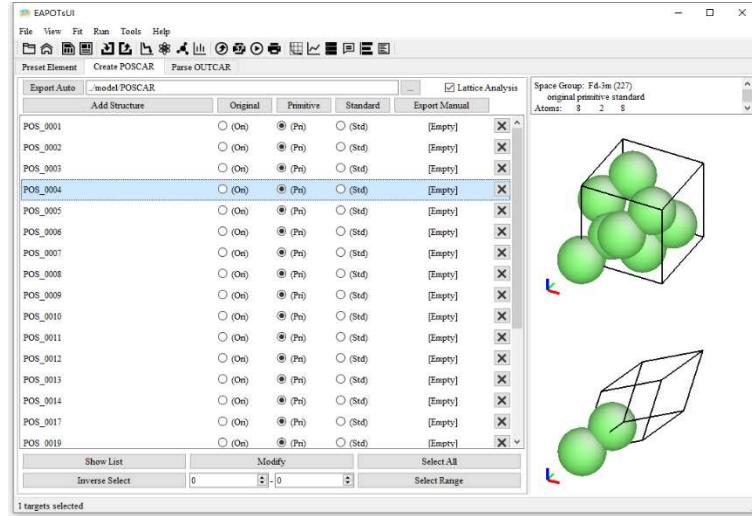


Fig. 5 Screenshot of the potential presetting window

The potential presetting window, as shown in Fig. 5, is used for the structure preparation and DFT calculations parsing.

3.Structure modeling window

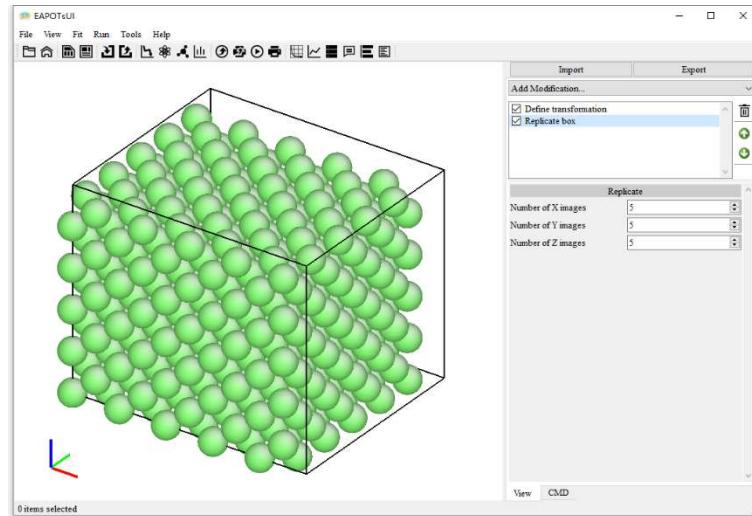


Fig. 6 Screenshot of the structure modeling window

The potential presetting window, as shown in Fig. 6, is used for fine editing of atomic structure.

4.Potential plotting window

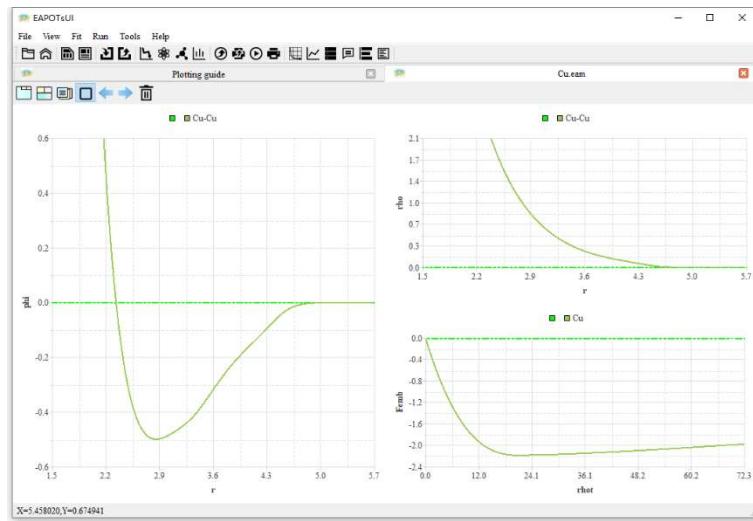


Fig. 7 Screenshot of the potential plotting window

The potential plotting window, as shown in Fig. 7, is used for potential curve visualization and energy surface visualization.

3. Potential fitting

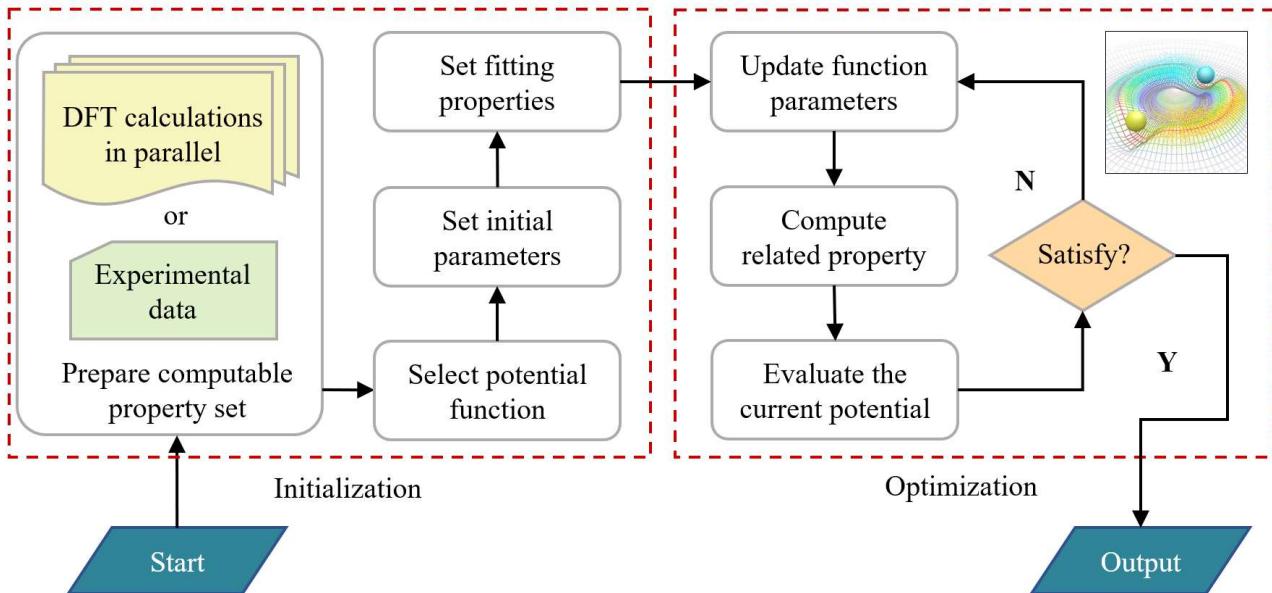


Fig. 8 Workflow of the core construction process in EAPOTs

Generally, the potential construction can be divided into three steps: initialization, optimization, validation, and output, where the related workflow is illustrated in Fig. 8. At the initialization step, the potential parameters and training targets should be provided or prepared, although some default parameters are already provided by EAPOTs. During each optimization step, all fitting parameters of the potential functions vary according to the minimization strategy and then: (i) update the potential parameters at each loop step; (ii) compute the related target properties, and (iii) evaluate the predefined cost function. In the validation and output step, the optimized potential parameters are validated by reproducing several predefined properties, and then the final potential is saved in a tabulated potential format adapted to atomistic simulation codes, such as LAMMPS [1]. In EAPOTs, the construction workflow is divided into five steps:

1. Choose potential functions,
2. Prepare fitting targets,
3. Set minimization algorithms,
4. Add other constraints,
5. Run optimizations.



The five construction steps can be switched by fit menu or left buttons split by arrows, as shown in Fig. 9 marked by orange color. In the following sections, we provide a full description of the five construction steps.

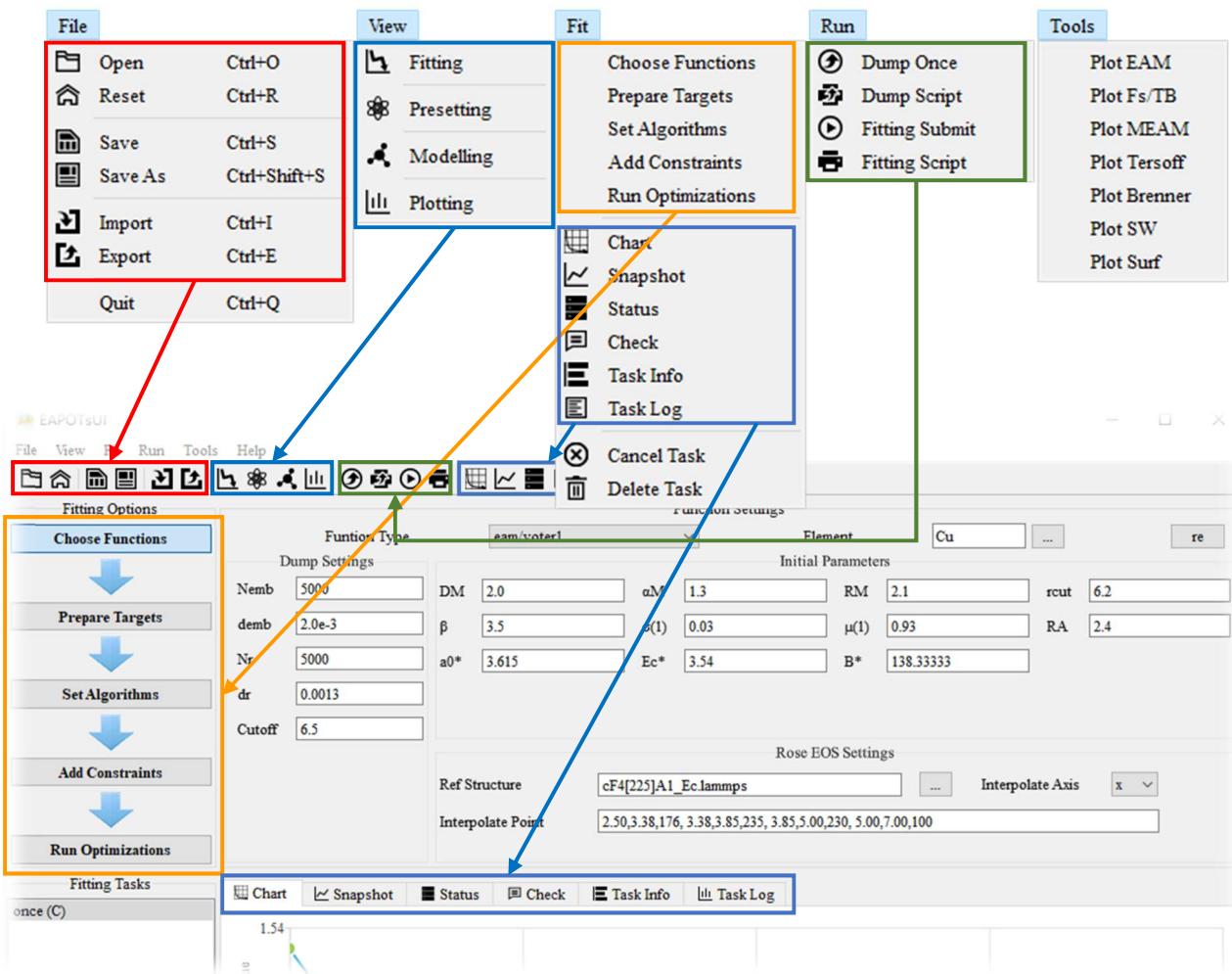


Fig. 9 Fitting workflow and main menu options

3.1. Choose potential functions

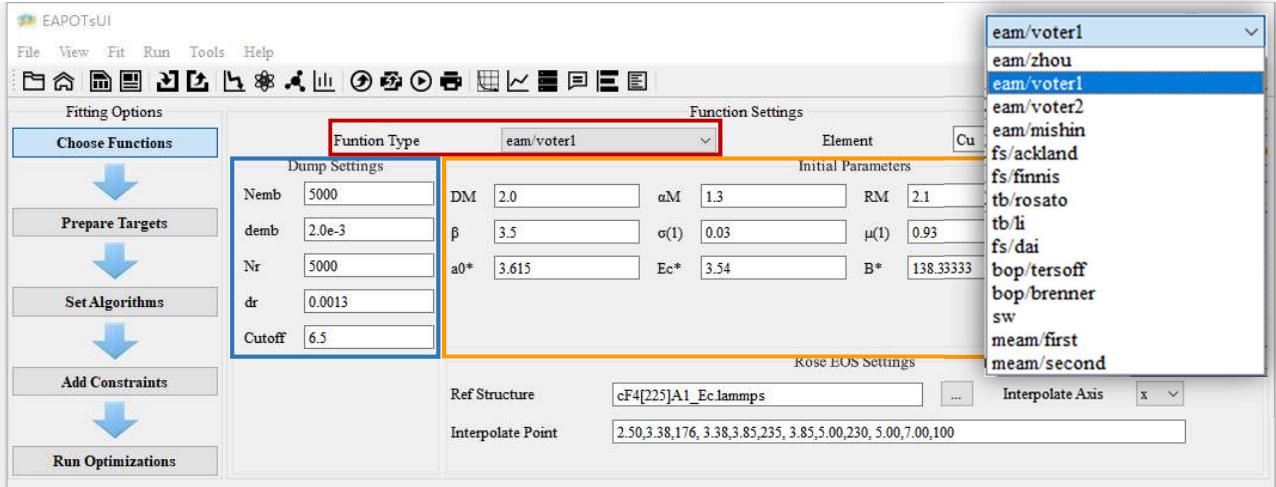


Fig. 10 Screenshot of potential function settings

In EAPOTs, the following potential models are implemented: i) the embedded-atom method (EAM) [2-4], tight-binding (TB) [5-7], and Finnis-Sinclair (F-S) [8] potentials; ii) the Stillinger-Weber potential [9] and the analytic bond-order potential (BOP) in the Tersoff–Abell [10, 11] and Brenner formats [12]; and iii) the modified EAM (MEAM) and second-nearest neighbor MEAM potentials [13-15].

As shown in Fig. 10, users can choose the potential function form in the “Function Type” Combobox (marked by red color), set the potential parameters in the “Initial Parameters” frame (marked by orange color), and other properties in this step. As for EAM, FS, and TB models, EAPOTs will output the potential in a specific tabulated format, as controlled by those parameters in the “Dump Settings” frame as shown in Fig. 10 (marked by cyan color). For EAM potentials, potential functions are tabulated in the LAMMPS *eam/alloy* format, which is the same as the DYNAMO multi-element setfl format. As an example, the [res/Resource/FeCuNi.eam](#) file has tabulated EAM values

for 3 elements and their alloy interactions: Fe, Cu, and Ni and formatted as follows:

- lines 1,2,3 = comments (ignored)
- line 4: Nelements Element1 Element2 ... ElementN
- line 5: Nrho, drho, Nr, dr, cutoff

In an *eam/alloy* file, the element name (Fe, Cu, Ni) of each element must be added to the line in the order, the elements appear in the file. On line 5, Nrho and Nr are the number of tabulated values in the subsequent arrays, drho and dr are the spacing in density and distance space for the values in those arrays, and the cutoff (in Angstroms) is a global value, valid for all pairwise interactions for all element pairings.

Following the 5 header lines are Nelements sections, one for each element, each with the following format:

- line 1 = atomic number, mass, lattice constant, lattice-type (e.g. FCC)
- embedding function F(rho) (Nrho values)
- density function rho(r) (Nr values)

The cubic lattice constant is in Angstroms and the F and rho arrays are unique to a single element. Following the Nelements sections, Nr values for each pair potential phi(r) array are listed for all i, j element pairs in the same format as other arrays. Since these interactions are symmetric ($i, j = j, i$) only phi arrays with $i \geq j$ are listed, in the following order: $i,j = (1,1), (2,1), (2,2), (3,1), (3,2), (3,3), (4,1), \dots, (Nelements, Nelements)$. The tabulated values for each phi function are listed as $r*\phi$ (in units of eV-Angstroms) since they are for atom pairs.

For FS and TB potentials, tabulated files include more information than the

eam/alloy format files, in that i,j density functionals for all pairs of elements are included as needed by the Finnis/Sinclair formulation, formatted as follows:

- lines 1,2,3 = comments (ignored)
- line 4: Nelements Element1 Element2 ... ElementN
- line 5: Nrho, drho, Nr, dr, cutoff
- The 5-line header section is identical to an EAM file.

Following the header are Nelements sections, one for each element , each with the following format:

- line 1 = atomic number, mass, lattice constant, lattice type (e.g. FCC)
- embedding function F(rho) (Nrho values)
- density function for element at element 1 (Nr values)
- density function for element at element 2
- ...
- density function for element at element N

The units of these quantities in line 1 are the same as for EAM files. Note that the rho(r) arrays in Finnis/Sinclair can be asymmetric $\rho_{ij}(r) \neq \rho_{ji}(r)$ so there are $Nelements^2$ of them listed in the file. Following the Nelements sections, Nr values for each pair potential phi(r) array are listed in the same manner ($r^*\phi$, units of eV-Angstroms) as in EAM files. Note that in Finnis/Sinclair, the phi(r) arrays are still symmetric, so only phi arrays for $i \geq j$ are listed.

For Stillinger-Weber, the analytic bond-order potential (BOP) in the Tersoff-Abell or Brenner formats, and the modified EAM (MEAM) models, EAPOTs will directly

output potential parameters formatted as a series of entries, as more details can be found in the LAMMPS [1] manual.

3.2. Prepare fitting targets

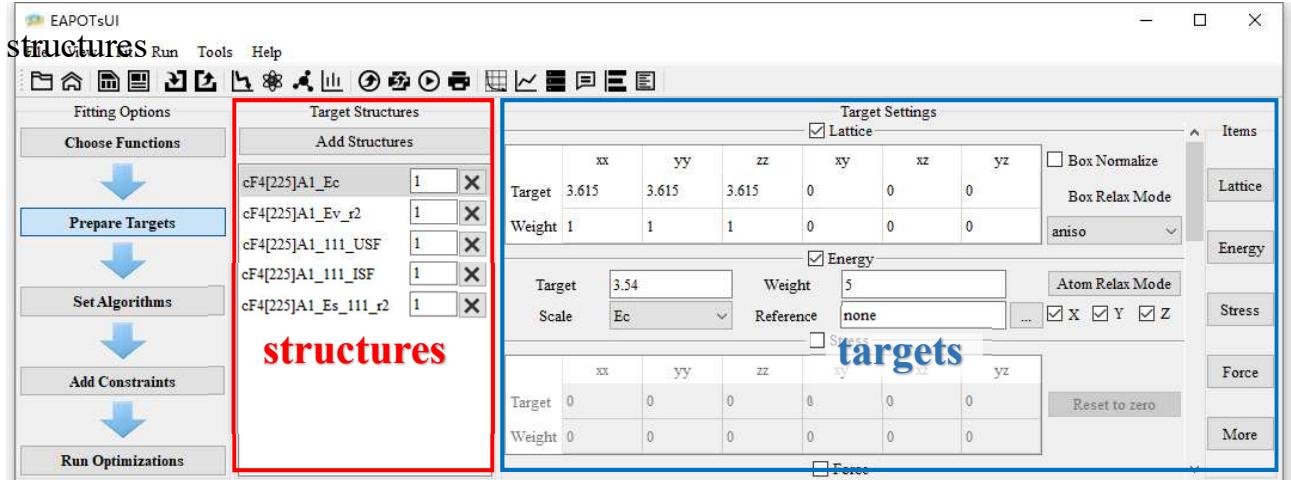


Fig. 11 Screenshot of fitting targets settings

The optimization quality depends on the minimization process for a cost function that quantifies the difference between the predicted and trained sets. The cost function used in EAPOTs is the well-known least-squares method for minimizing the sum of squares of the deviations for evaluation, which is expressed in the general form:

$$Z_{\text{tot}} = \sum_i^N u_i Z_i, \quad (3.1)$$

where N is the number of structures, u_i is the weight factor for the i th structure, and Z_i represents the corresponding difference between the predicted and trained targets, given by:

$$Z_i = \sum_{k=0}^N w_k \delta(A_k^{\text{predict}}, A_k^{\text{target}})^2 + \frac{1}{3n} \sum_{j=0}^n \sum_{\alpha=0}^3 w_{i,\alpha} \delta(F_{j,\alpha}^{\text{predict}}, F_{j,\alpha}^{\text{target}})^2, \quad (3.2)$$

with the deviation function:

$$\delta(x^{\text{predict}}, x^{\text{target}}) = \begin{cases} (x^{\text{predict}} - x^{\text{target}})/x^{\text{target}} & , x^{\text{target}} \neq 0 \\ x^{\text{predict}} - x^{\text{target}} & , x^{\text{target}} = 0 \end{cases}. \quad (3.3)$$

Here, A_k^{predict} and A_k^{target} are the k th global prediction and target, respectively, and $F_{j,\alpha}^{\text{predict}}$ and $F_{j,\alpha}^{\text{target}}$ values are the α component of the force prediction and target for the atom j , respectively. Moreover, w_k and $w_{i,\alpha}$ are the corresponding weight factors, which are necessary since several specific targets are usually more critical than others during the fitting procedure. It is helpful to measure the relative deviations rather than the absolute deviations from the reference data, except for the zero values and atomic forces, where the absolute deviations are used by default due to the large magnitude difference in atomic force values. (The deviation function of forces can be set manually, as shown in Fig. 12 marked by red color).

To meet the demands of different simulation scenarios, EAPOTs realizes multiple combinations of "energy-stress-force-elasticity" and multi-level objective optimization schemes, where both experimental and ab initio data can be introduced as training targets in the optimization. The inner relaxation is supported in EAPOTs for a given structure, enabling the training to the experimental equilibrium values, e.g., the lattice constant, cohesive energy, elastic constants, stacking fault energies (SFEs), surface energies, or other equilibrium properties. The inner relaxation can be divided into two types: the box relaxation and atomic relaxation, as shown in Fig. 12.

<input checked="" type="checkbox"/> Lattice																	
Target	xx	yy	zz	xy	xz	yz											
Weight	5.14393	2.96985	34.5492	0	0	0											
						<input checked="" type="checkbox"/> Box Normalize Box Relax Mode none											
<input checked="" type="checkbox"/> Energy																	
Target	1889	Weight	5	Atom Relax Mode													
Scale	Es-xy	Reference	cF4[225]A1_Ec	<input type="checkbox"/> X <input type="checkbox"/> Y <input type="checkbox"/> Z													
<input checked="" type="checkbox"/> Energy scale						<input checked="" type="checkbox"/> Reference											
Target	xx	yy	zz	xy	xz	yz											
Weight	0	0	0	0	0	0											
<input checked="" type="checkbox"/> Force																	
1	T(x)	W(x)	T(y)	W(y)	T(z)	W(z)	Weight 0 <input type="checkbox"/> Use Max Force Reset 0.0 All Select Target Weight Eval Mode Fx abs Fy abs Fz abs										
2	0	0	0	0	0	0											
3	0	0	0	0	0	0											
4	0	0	0	0	0	0											
5	0	0	0	0	0	0											
6	0	0	0	0	0	0											
7	0	0	0	0	0	0											
8	0	0	0	0	0	0											
<input checked="" type="checkbox"/> More																	
Elastic Constant Target						Elastic Constant Weight											
xx	yy	zz	xy	xz	yz	xx	yy	zz	xy	xz	yz						
xx	0	0	0	0	0	0	0	0	0	0	0						
yy	0	0	0	0	0	0	0	0	0	0	0						
zz	0	0	0	0	0	0	0	0	0	0	0						
xy	0	0	0	0	0	0	0	0	0	0	0						
xz	0	0	0	0	0	0	0	0	0	0	0						
yz	0	0	0	0	0	0	0	0	0	0	0						
Bulk			Shear			Poisson Ratio			Young-x			Young-y			Young-z		
Target	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Weight	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Lattice

Normalize

Energy

Relax/box

Stress

Relax/atom

Force

Force batch operations

Elastic

Elastic constants

Fig. 12 Complete training target setting window

Box relax mode: The keyword *none* (as shown in Fig. 13) means no box relaxation in the inner loop. The keyword *iso* means couple all 3 diagonal components together when pressure is computed (hydrostatic pressure). The keyword *aniso* means x, y, and z dimensions are controlled independently using the Pxx, Pyy, and Pzz components of the stress tensor as the driving forces. The keyword *full* means x, y, z, xy, xz, and yz dimensions are controlled independently using their individual stress components as the driving forces.

Atom relax mode: Check to activate atomic relaxation in the corresponding axis.

Reference: The choice of this parameter has two effects: 1. as a reference structure to get relax ratio (as shown in Fig. 13). 2. as the reference energy if “energy scale” calculates the different energy.

Normalize: Normalize the lattice according to the relax ratio of the reference structure.

Energy scale: A series of predefined expressions, as shown in Table 2.

For more details on energy settings, please refer to Fig. 13.

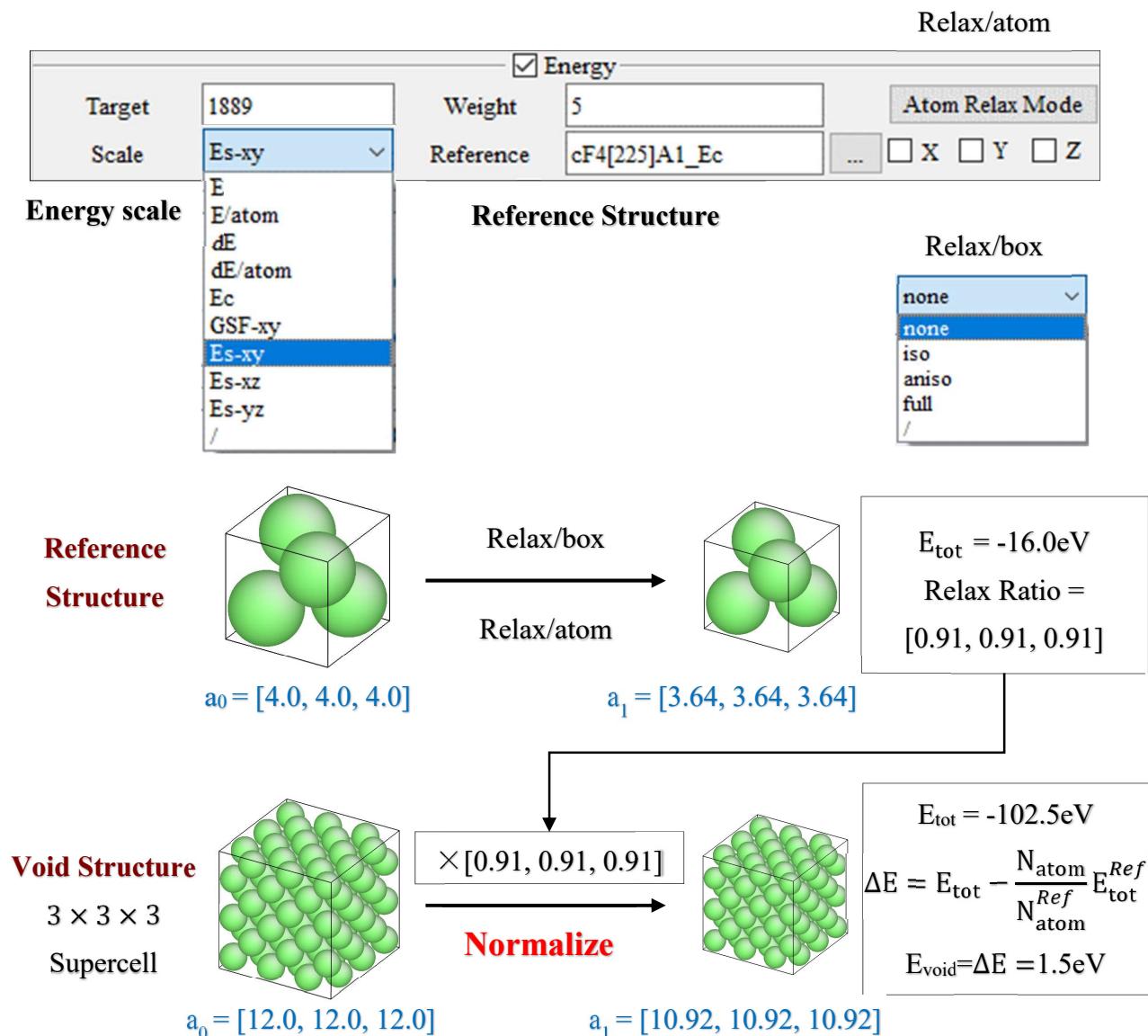


Fig. 13 Energy calculation workflow

Table 2 Energy scale expressions

	Unit	expression
E	eV	E_{tot}
E/atom	eV	$E_{\text{tot}}/N_{\text{atom}}$
dE	eV	$E_{\text{tot}} - \left(N_{\text{atom}}/N_{\text{atom}}^{\text{Ref}} \right) \cdot E_{\text{tot}}^{\text{Ref}}$
dE/atom	eV	$E_{\text{tot}}/N_{\text{atom}} - E_{\text{tot}}^{\text{Ref}}/N_{\text{atom}}^{\text{Ref}}$
Ec	eV	$-E_{\text{tot}}/N_{\text{atom}}$
GSF-xy	mJ/m ²	$\frac{16020}{S_{\text{xy}}} \left[E_{\text{tot}} - \left(\frac{N_{\text{atom}}}{N_{\text{atom}}^{\text{Ref}}} \right) \cdot E_{\text{tot}}^{\text{Ref}} \right]$
Es-xy	mJ/m ²	$\frac{8010}{S_{\text{xy}}} \left[E_{\text{tot}} - \left(\frac{N_{\text{atom}}}{N_{\text{atom}}^{\text{Ref}}} \right) \cdot E_{\text{tot}}^{\text{Ref}} \right]$
Es-xz	mJ/m ²	$\frac{8010}{S_{\text{xz}}} \left[E_{\text{tot}} - \left(\frac{N_{\text{atom}}}{N_{\text{atom}}^{\text{Ref}}} \right) \cdot E_{\text{tot}}^{\text{Ref}} \right]$
Es-yz	mJ/m ²	$\frac{8010}{S_{\text{yz}}} \left[E_{\text{tot}} - \left(\frac{N_{\text{atom}}}{N_{\text{atom}}^{\text{Ref}}} \right) \cdot E_{\text{tot}}^{\text{Ref}} \right]$

Note: S represents the area size and 16020 is the unit conversion factor.

3.3. Set minimization algorithms

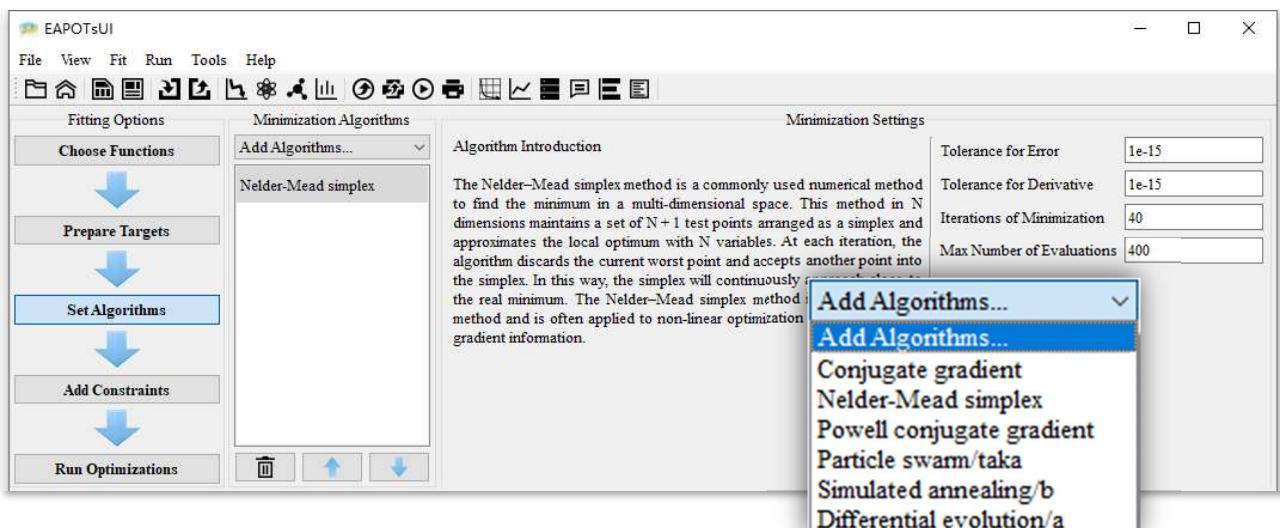


Fig. 14 Screenshot of minimization algorithms settings

In general, minimization algorithms can be classified into local and global algorithms. While local algorithms always find the closest local minimum, global algorithms are non-deterministic methods that sample a larger part of the parameter space and find the best minimum but have a computational burden that is several orders of magnitude larger.

As evaluating the nonlinear cost function is relatively expensive, several efficient local minimization algorithms were implemented in EAPOTs. The Nelder–Mead simplex method [16] is a commonly used numerical method to find the minimum in a multi-dimensional space. This method in N dimensions maintains a set of $N + 1$ test points arranged as a simplex and approximates the local optimum with N variables. At each iteration, the algorithm discards the current worst point and accepts another point into the simplex. In this way, the simplex will continuously approach close to the real minimum. The Nelder–Mead simplex method is a direct search method and is often applied to non-linear optimization problems without gradient information.

The conjugate gradient (CG) algorithm was implemented in the Polak-Ribiere version [17] where the next search direction must be conjugate to the previous search directions. This algorithm begins the optimization in the negative gradient direction, and at each iteration, the gradient is combined with the previous iteration information to create a new search direction. This method has the feature that it only requires the storage of two gradient vectors, which for large problems with many parameters, is a significant saving in storage versus Newton-type algorithms.

Powell algorithm [18] is a conjugate-gradient-like optimization method that is

efficient in the number of calculations and does not require any gradient information. The Powell method first determines all search vectors at the starting point and then transforms the multi-dimensional optimization into a line search problem. At each iteration, a new parameter guess is determined by a sequence of line searches. Afterward, the search direction which gives the largest impact is replaced by a new search direction, and the iterations are performed until convergence.

The particle swarm optimization (PSO) [19] is an efficient global searching method, where the optimization is carried out with typically dozens of particles, each treated as a parameter set that is initialized randomly. The basic idea of PSO is to evolve each particle towards the optimum using two contributions: (1) their best-fitted parameters in the past and (2) the best-fitted particle in the swarm. Each particle keeps track of its coordinates in the problem space associated with the best solution it has achieved so far. When a particle takes all the population as its neighbors, the global solution is achieved.

Differential evolution [20] algorithms perform optimization by mimicking natural selection with a population of candidate solutions to the problem. The population is then evolved from one generation to the next, using techniques inspired by biological evolution such as mutation or natural selection. Typically weaker members of a population are replaced by fitter offspring created by the combination of traits from other population members, thus increasing the population's overall quality.

The simulated annealing algorithm was also implemented [21, 22]. Here, jumps are made in potential-parameter space, with downhill moves (those that result in a decrease in Z_{cost}) always accepted and uphill moves accepted only with a probability $\exp^{-\Delta Z/T}$.

Here, ΔR is the increase in Z due to the proposed move, and T is a fictitious temperature providing an overall control on the probability of an uphill. By allowing uphill moves in this way, a more exhaustive search of the potential-parameter space is enabled, which leads to a real global minimum.

3.4. Add other constraints

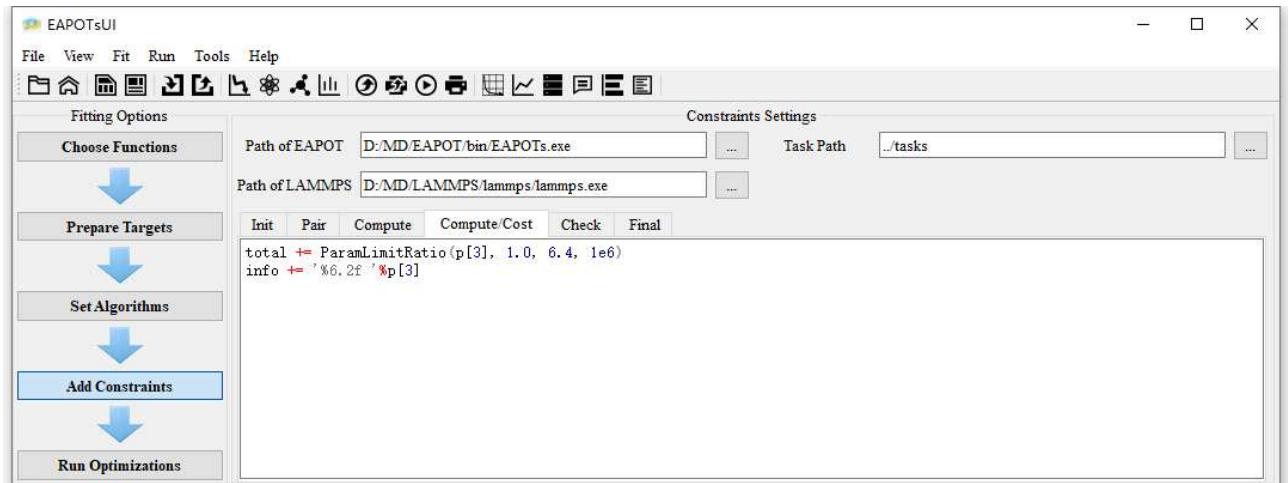


Fig. 15 Screenshot of other constraints settings

This window can be used to specify other program settings or enter additional custom codes. The optimization quality depends on the cost function that quantifies the difference between the predicted and trained sets, thus EAPOTs will always generate a python script according to the settings in the fitting target window to get the cost function value. For example, settings in Fig. 11 will be converted to the following Python script. In addition, this window allows additional custom codes to modify certain EAPOTs properties during the fitting procedure, for example, custom codes shown in Fig. 15 will be inserted into the cost function script (see those codes separated by comments):

```
sq2 = lambda w, t, v: w*(v/t-1)**2 if t else w*t**2
fmt = lambda u: '%9.3g '%u if abs(u) > 1e5 else '%5.4g '%u
def ParamLimitRatio(x, a, b, c): return c*(x<a)*(x/a-1)**2 + c*(b>x)*(x/b-1)**2
```

```

def ParamlimitAbslo(x, a, b, c): return c*(x<a)*(x - a)**2 + c*(b>x)*(x - b)**2
weight = [ 1, 1, 1, 5, 3, 3, 3, 5, 10, 10, 5 ]
target = [ 3.615, 3.615, 3.615, 3.54, 170, 122.5, 75.8, 1.27, 161, 45, 1889 ]
targetMsg = ''.join(map(fmt, target))

def computeCostScript(caller, np, p, nparam, style, param, res):
    v = EAPOT.parseCompute(nparam, style, param)

    Ec          = v[0]
    EcEla       = v[1]
    Ev_r2       = v[2]
    USF         = v[3]
    ISF         = v[4]
    Es_r2       = v[5]

    val = [0.0]*11
    total, info = 0.0, ''
    val[0] = Ec.lattice[0]
    val[1] = Ec.lattice[1]
    val[2] = Ec.lattice[2]
    val[3] = Ec.Ec
    val[4] = EcEla.cubic[0]
    val[5] = EcEla.cubic[1]
    val[6] = EcEla.cubic[2]
    val[7] = Ev_r2.energy-Ec.energy*7.75
    val[8] = 16020/(USF.xx*USF.yy)*(USF.energy-Ec.energy*6)
    val[9] = 16020/(ISF.xx*ISF.yy)*(ISF.energy-Ec.energy*6)
    val[10] = 8010/(Es_r2.xx*Es_r2.yy)*(Es_r2.energy-Ec.energy*3)

    # custom extra code begin
    total += ParamlimitAbslo(p[3], 1.0, 6.4, 1e6)
    info += '%6.2f %p[3]
    # custom extra code end

    res[0] = sum(map(sq2, weight, target, val)) + total
    return EAPOT.packMsg(''.join(map(fmt, val)) + info)

```

Cost function script will evaluate the difference between the predicted and trained sets, and save it into the *res* variable. The return value of the cost function script is a short string message to give a summary of current fitting information. In this example, the custom script will limit the third fitting parameter, where “ParamlimitAbslo(p[3],

$1.0, 6.4, 1e6)$ " will limit $p[3]$ to below 6.4 to prevent it from exceeding the cut-off radius.

3.5. Run optimizations

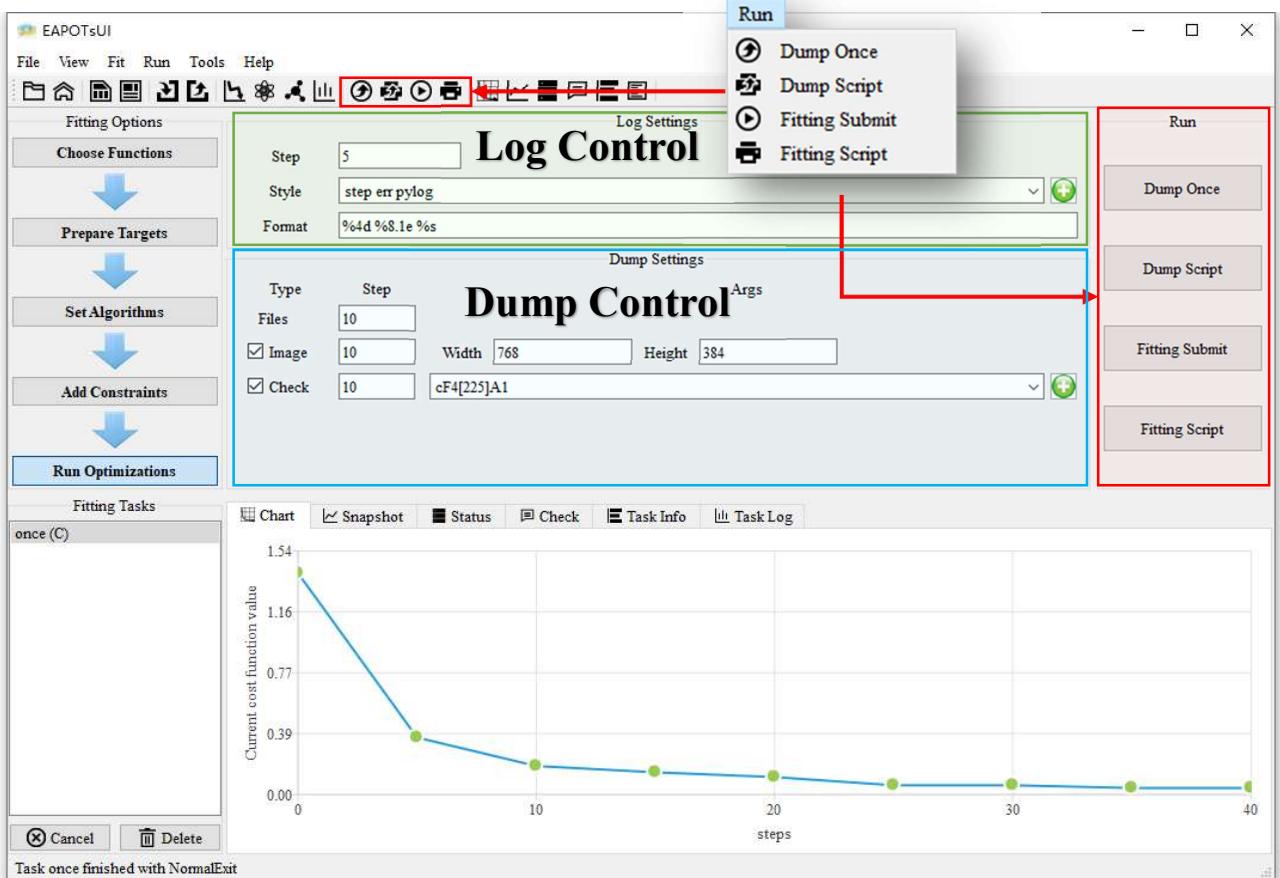


Fig. 16 Screenshot of other constraints settings

The log system print info (e.g. cost function value, timestep) on timesteps that are a multiple of N and at the beginning and end of a simulation. A step value of 0 will only print the log at the beginning and end. The content and format of what is printed are controlled by the “style Combobox” and “format Combobox”, as shown in Table 3.

Table 3 Log keywords

keyword	description
step	timestep
eval	cost function evaluation times
err	the current cost function value
pylog	summary of current fitting information from the cost function

The dumping system outputs a potential file, snapshot, or checking result to one or more files every N timesteps in one of several styles. After all the settings are finished, users can submit a task by the right buttons in Fig. 16 and more details are shown in Table 4. Normally, the “Dump Once” button can be used to test and will not allocate a new folder, while the “Fitting Submit” button will submit a new task formally.

Table 4 Submit button content

button	content
Dump Once	Submit the task in a fixed folder: /tasks/once
Dump Script	Export the project file in a fixed folder: /tasks/once
Fitting Submit	Submit the task in a new folder
Fitting Script	Export the project file in a new folder

3.6. Potential fitting task manager

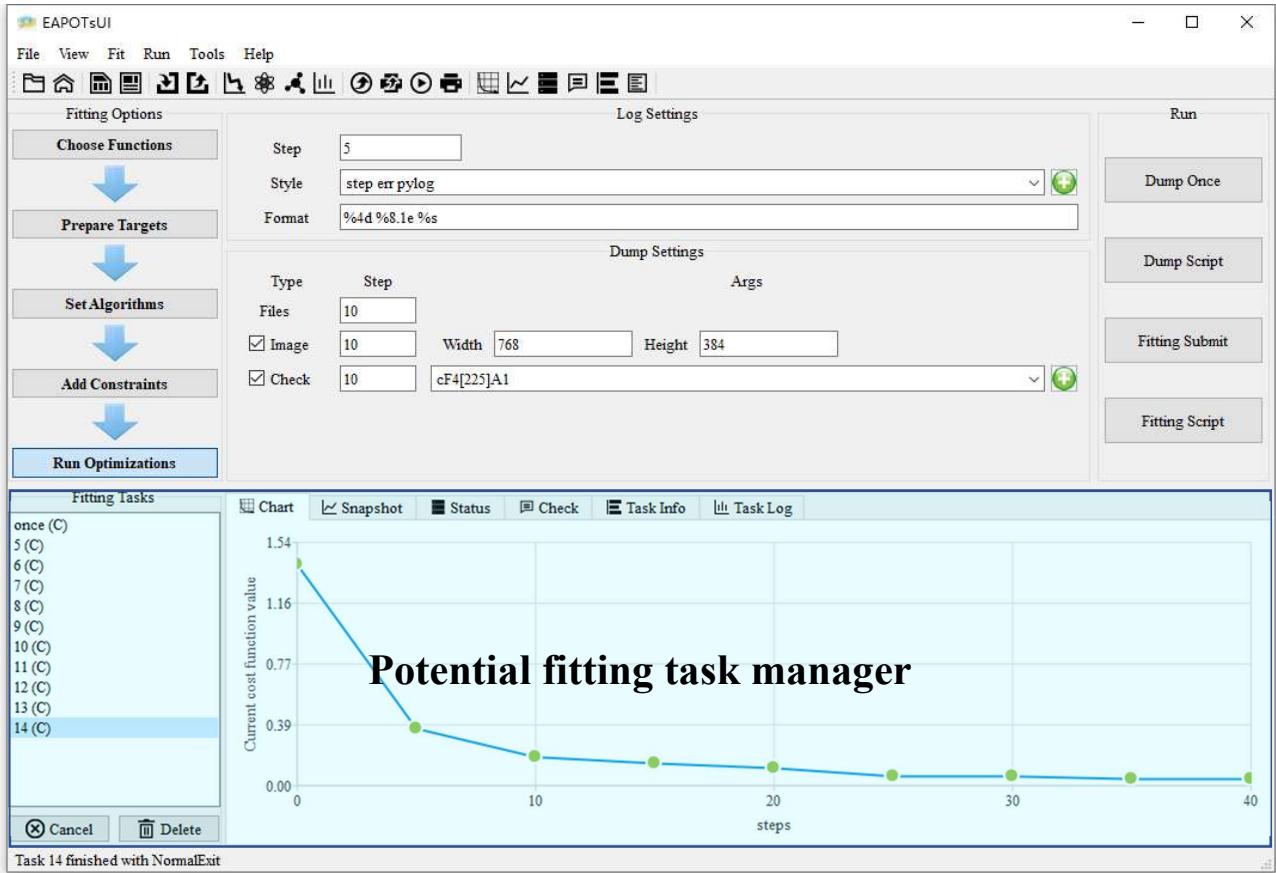


Fig. 17 Screenshot of Potential fitting task manager (Cost function)

Once a task is submitted, the potential fitting task manager will provide a dynamic real-time view of running tasks. It can display summary information of the selected task as well as a list of tasks currently being managed by the EAPOTs, where users can double-click the list item to open the task folder in the file manager. The task summary information includes the potential snapshot, status, checking results, running information, and exception log which records all exceptions during the running time. The fitting task manager provides an interactive interface for process manipulation as more details are shown in Fig. 18.

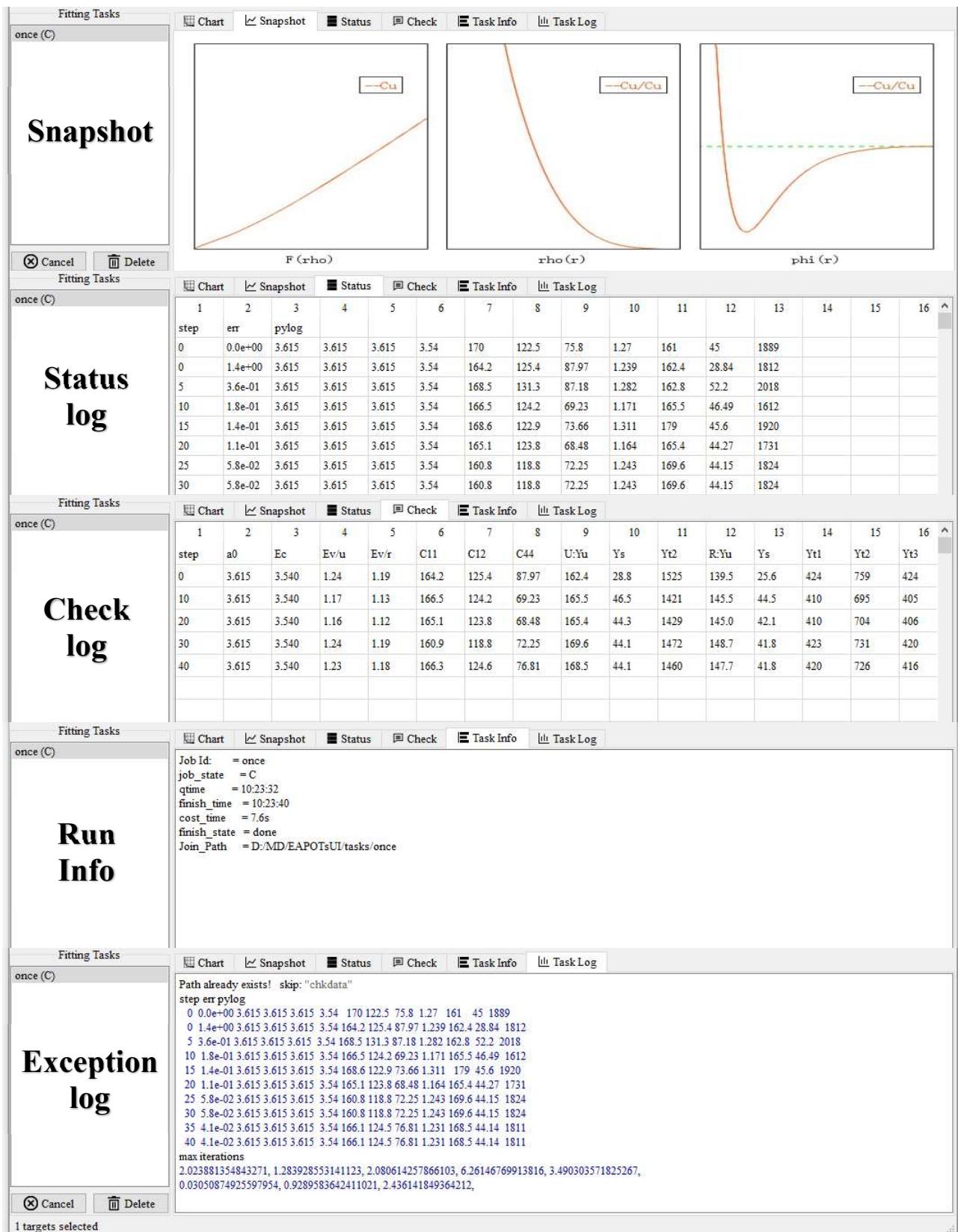


Fig. 18 Screenshot of Potential fitting task manager (All views)

4. Potential presetting

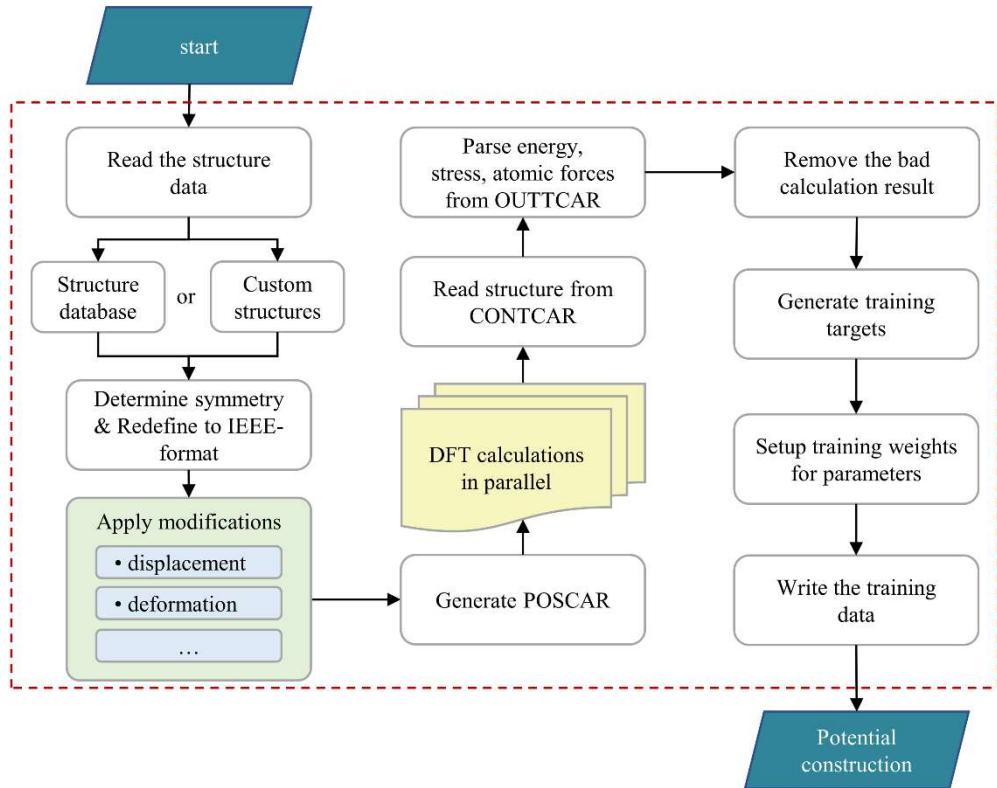


Fig. 19 The high-throughput model construction workflow used in the EAPOTs code.

To address the deficiency of experimental data [23], Ercolessi and Adams [24] proposed a force-matching method via first-principles calculations, which provided a feasible solution to expand the search for a high-dimensional, potential energy space. Motivated by this advancement, EAPOTs realizes the HT strategy to generate effective interaction potentials by matching microscopic quantities from first-principles methods, such as energies, stresses, elasticity, and forces. Fig. 19 shows the HT flowchart implemented in EAPOTs through the automatic processing of model construction and data delivery, and a brief introduction of the two steps will be shown in the following.

4.1. Create atomic structures for DFT calculations

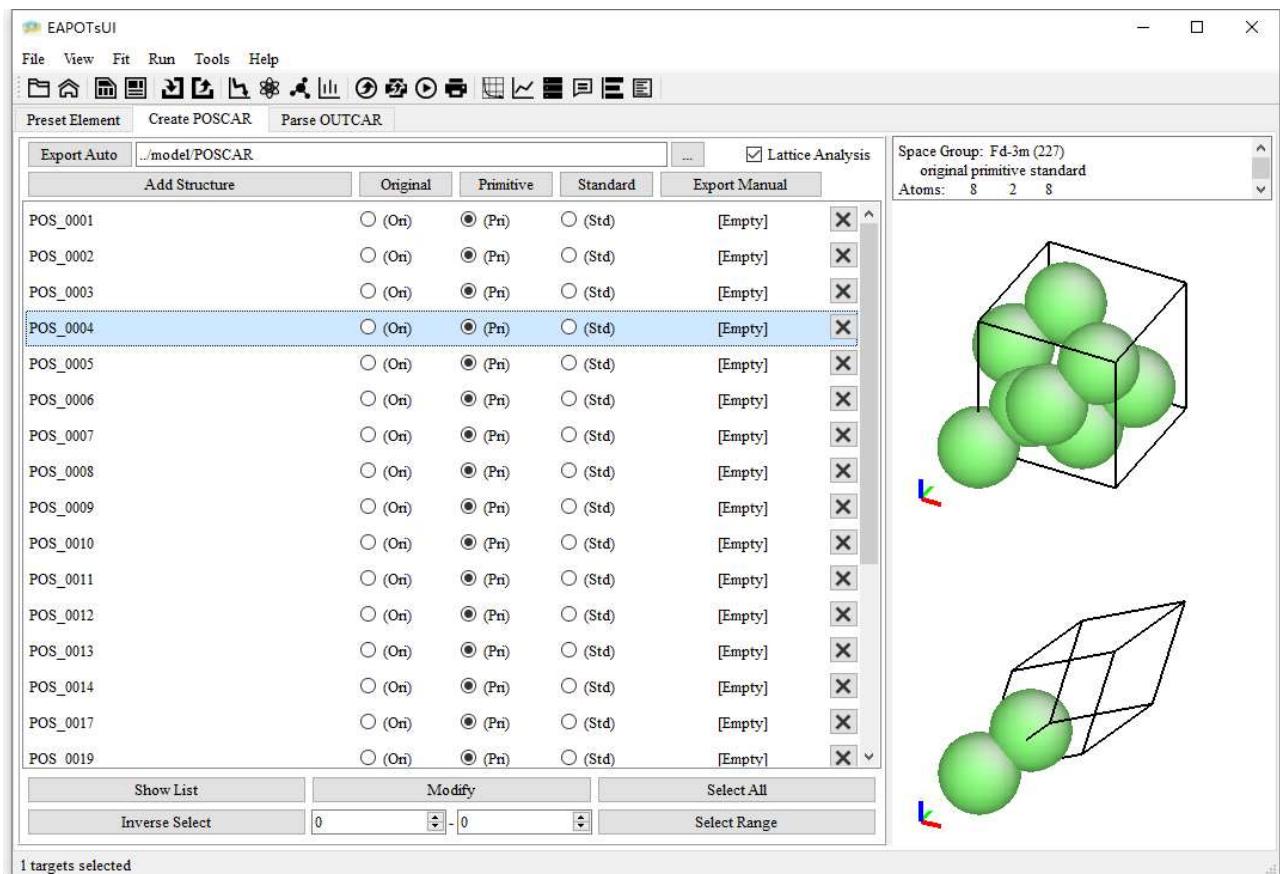


Fig. 20 Model import and lattice symmetry analyze

First, EAPOTs read structure data containing lattice vectors and atomic coordinates, and then the lattice symmetry is analyzed using the SPGLIB code [25], as shown in Fig. 20. Based on the symmetry, various lattice and atomic modifications are performed on the reduced primary or unit cells to cover various possible configurations that may be relevant to phase transformation, crystal plasticity, or diffusions, as shown in Fig. 21.

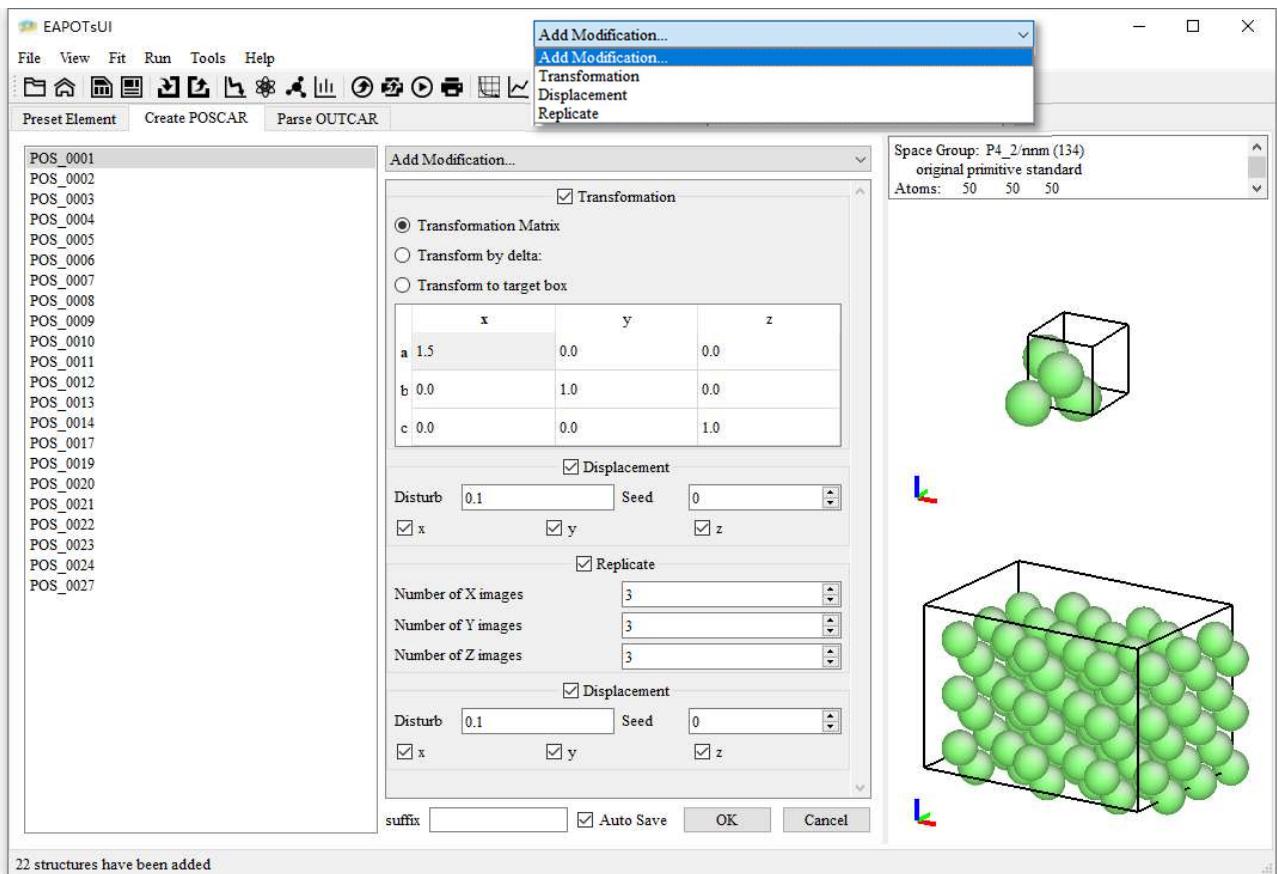


Fig. 21 Batch lattice and atomic modifications

Transformation modifier: This modifier applies an affine transformation to the system, which may be used to translate, scale, rotate, or shear the simulation cell. The transformation can be explicitly specified in terms of a 3×3 matrix, or directly by prescribing a target geometry for the simulation cell.

1. Transformation Matrix: $\mathbf{C} = \mathbf{C} \cdot \mathbf{M}$,
2. Transform by delta: $\mathbf{C} = \mathbf{C} + \mathbf{C} \cdot \mathbf{M}$,
3. Transform to target box: $\mathbf{C} = \mathbf{M}$,

where \mathbf{M} is the input matrix parameter and \mathbf{C} represents the domain vectors:

$$\mathbf{C} = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{pmatrix}$$

Displacement modifier: Apply random displacement /perturbation to atoms.

Replicate modifier: This modifier copies all particles multiple times to a system and the numbers of images on each axis (X/Y/Z) specify how many times the system is copied in each direction.

4.2. Parse atomic data from DFT calculations

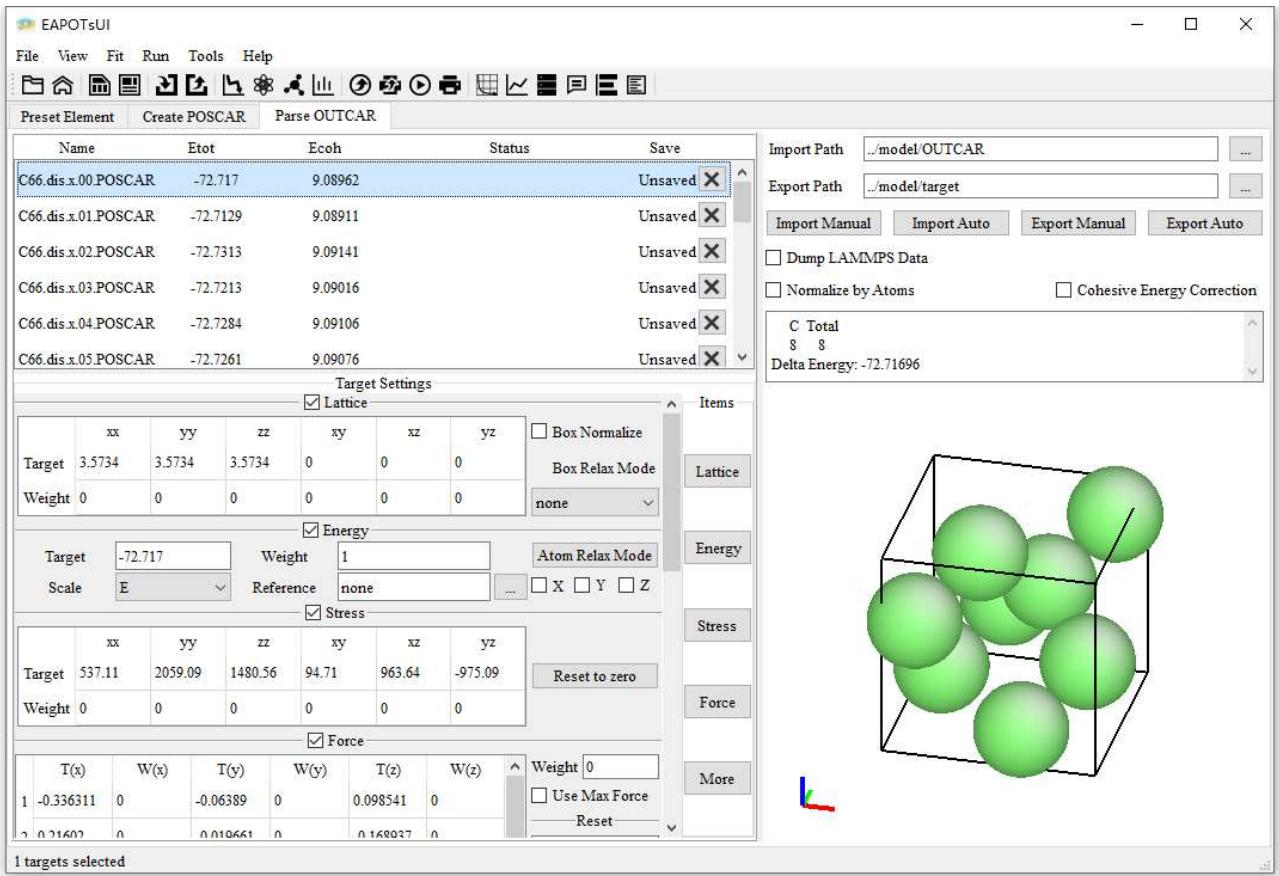


Fig. 22 DFT calculations parsing

Then, all structures are sent for parallel first-principle calculations to obtain their energies, stresses, and forces, and these calculated data are parsed from the generated files for further processing in EAPOTs. Finally, the targets are generated according to their different combinations or derivations to other physical properties, such as the SFE,

lattice stability, or elasticity.

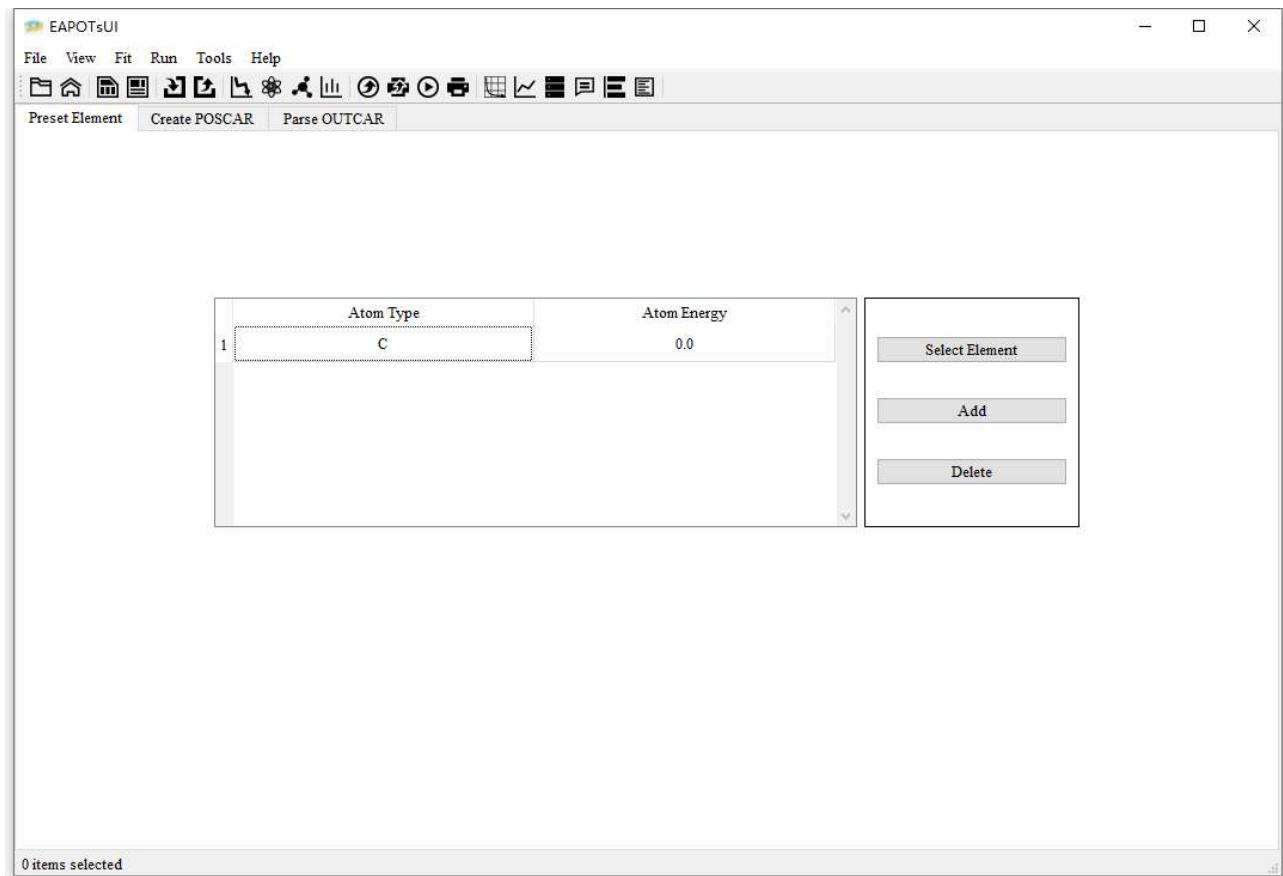


Fig. 23 Element information Settings

Besides, element type and energy for a single type atom (to calculate the cohesive energy) can be assigned in the first window, as shown in Fig. 23.

5. Structure modeling

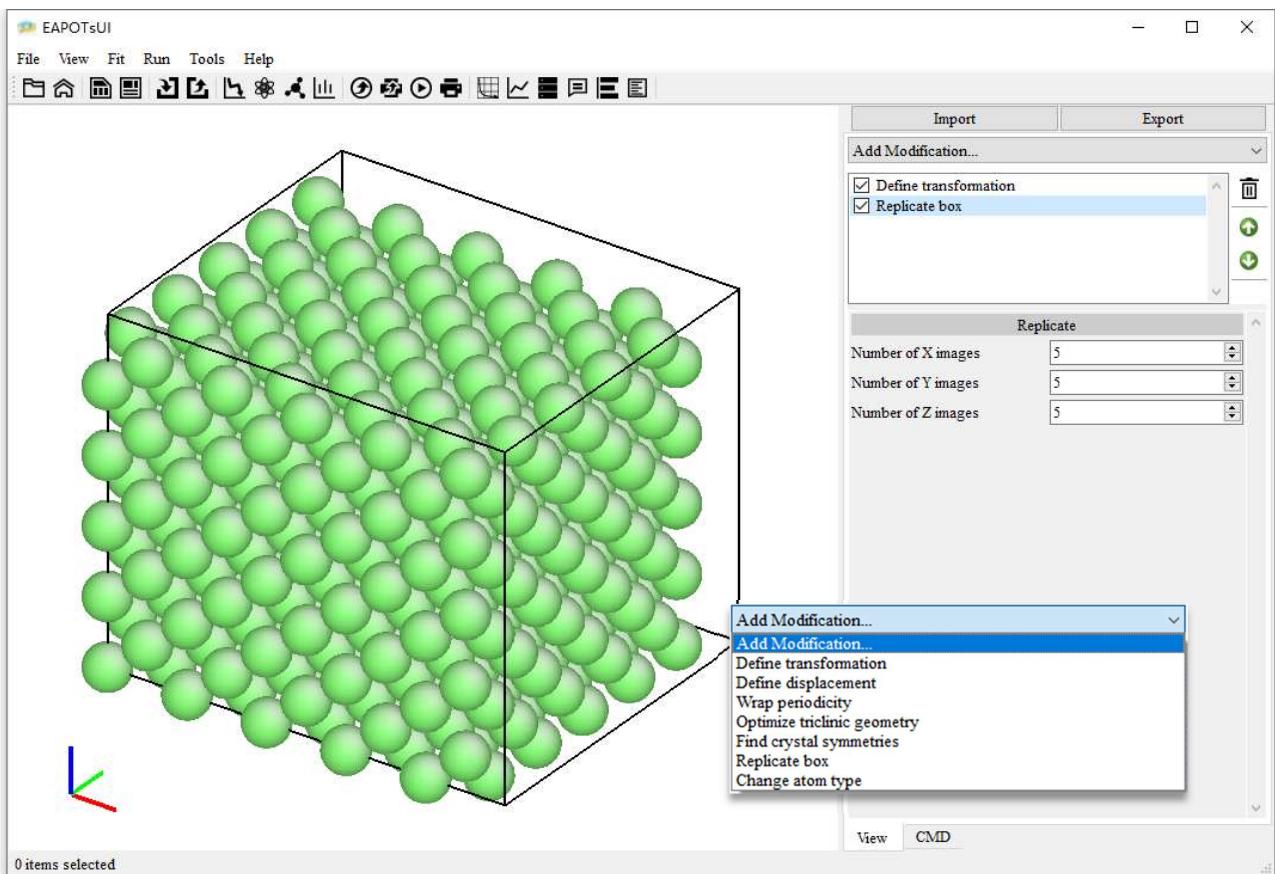


Fig. 24 Screenshot of the structure modeling window

Modifiers are the basic building blocks for creating a data pipeline in the structure modeling module, and each modifier implements a very specific, well-defined type of operation or computation, and typically you will need to combine several modifiers to accomplish more complex tasks, as more details shown in Table 5.

Modifier name	Description
Define transformaiton	Applies an affine transformation to the system.
Define displacement	Apply random displacement /perturbation to atoms.
Wrap periodicity	Folds particles located outside of the periodic

	simulation box back into the box.
Optimize triclinic geometry	Limit tilt factors that skew the box less than half the distance of the parallel box length
Find crystal symmetries	Get the space group symbol, group number, or create the standardized primitive cell
Replicate box	Duplicates particles to periodic images of the system.
Change atom type	Assign the element type to atoms

Table 5 Structure modeling modifiers

6. Potential plotting

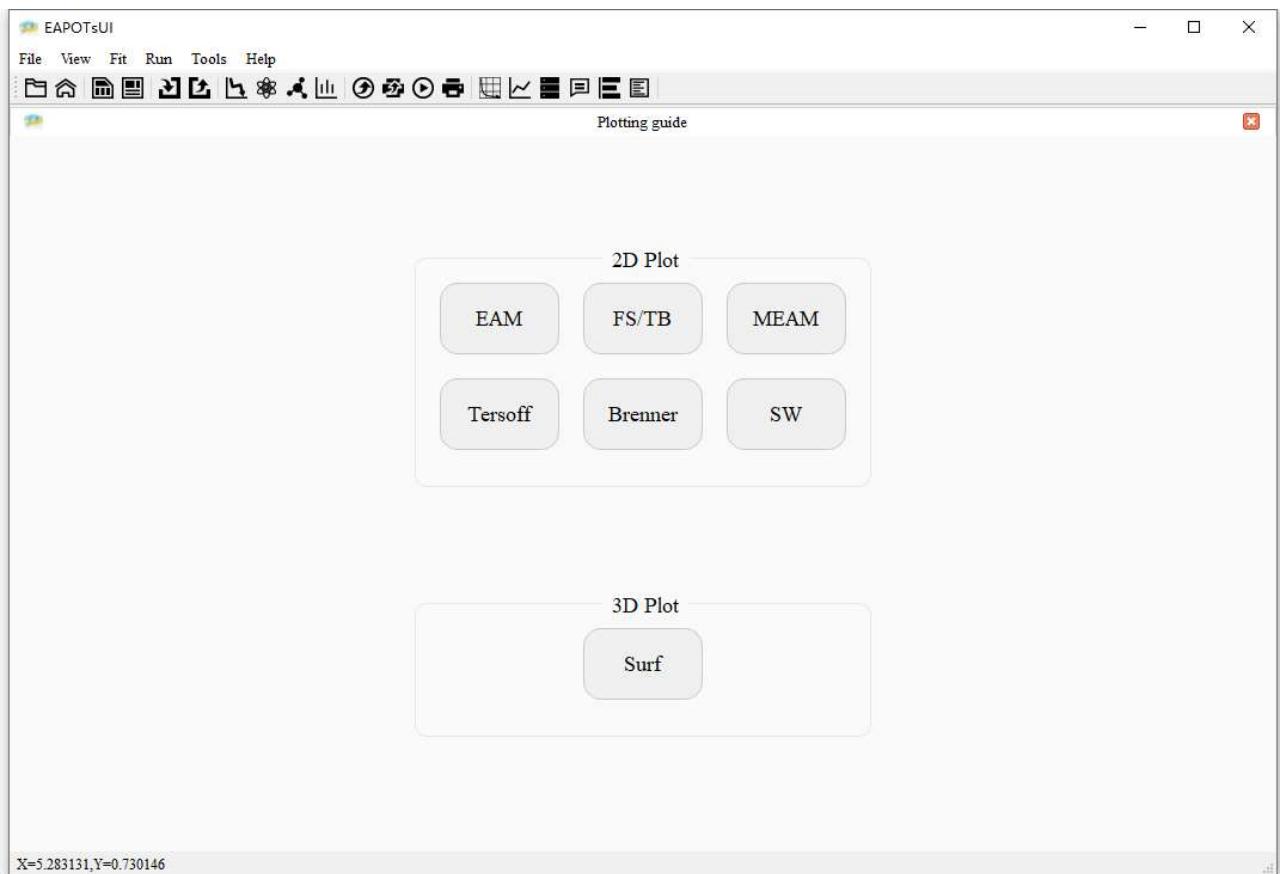


Fig. 25 Screenshot of the potential plotting window

The potential plotting window, as shown in Fig. 25, is used for potential curve visualization and energy surface visualization.

6.1. Snapshot of potential curves

Here are two visualization examples. The EAM potential, as shown in Fig. 26, contains: (a) pair interaction function, (b) embedding energy function, and (c) electron charge density function.

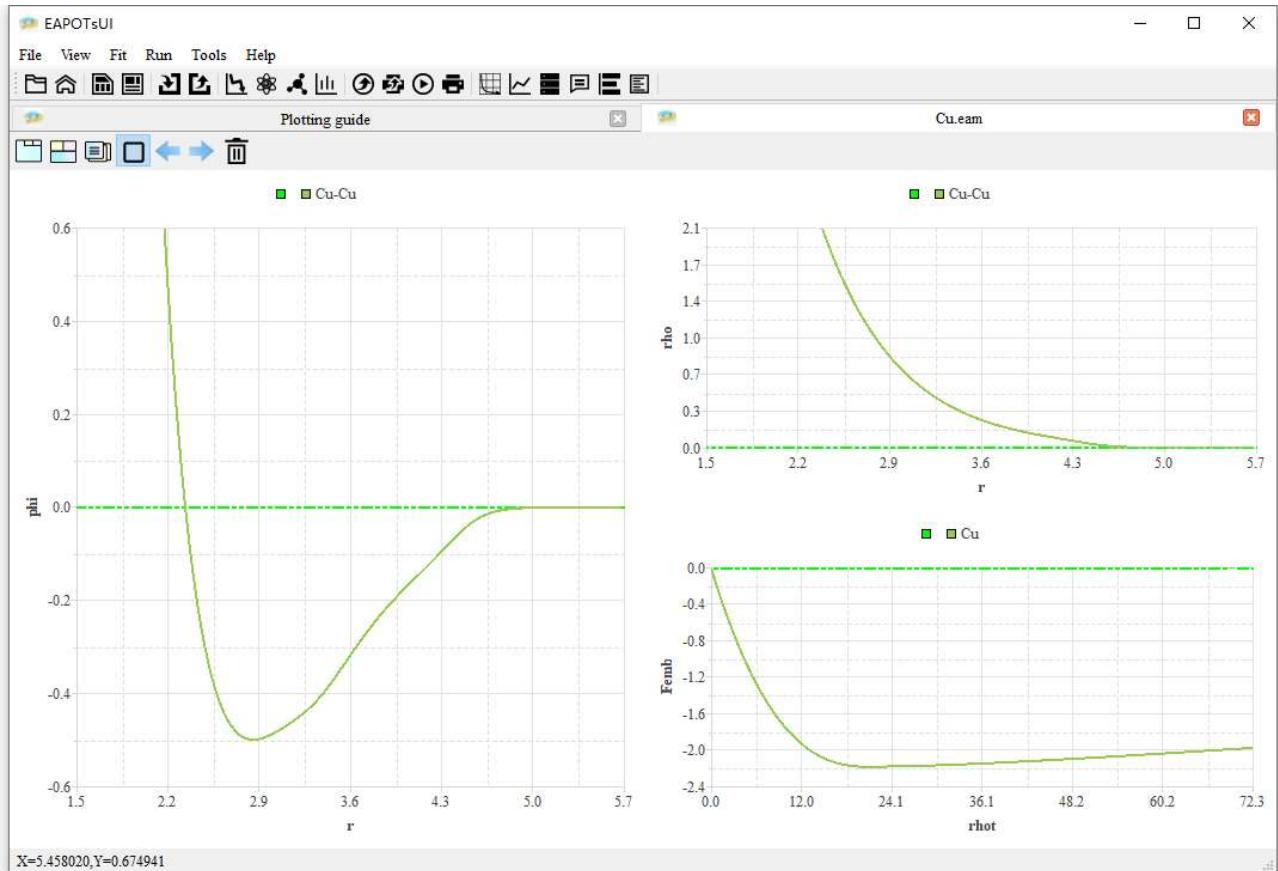


Fig. 26 Snapshot of EAM potential

Fig. 27 shows the SW potential:

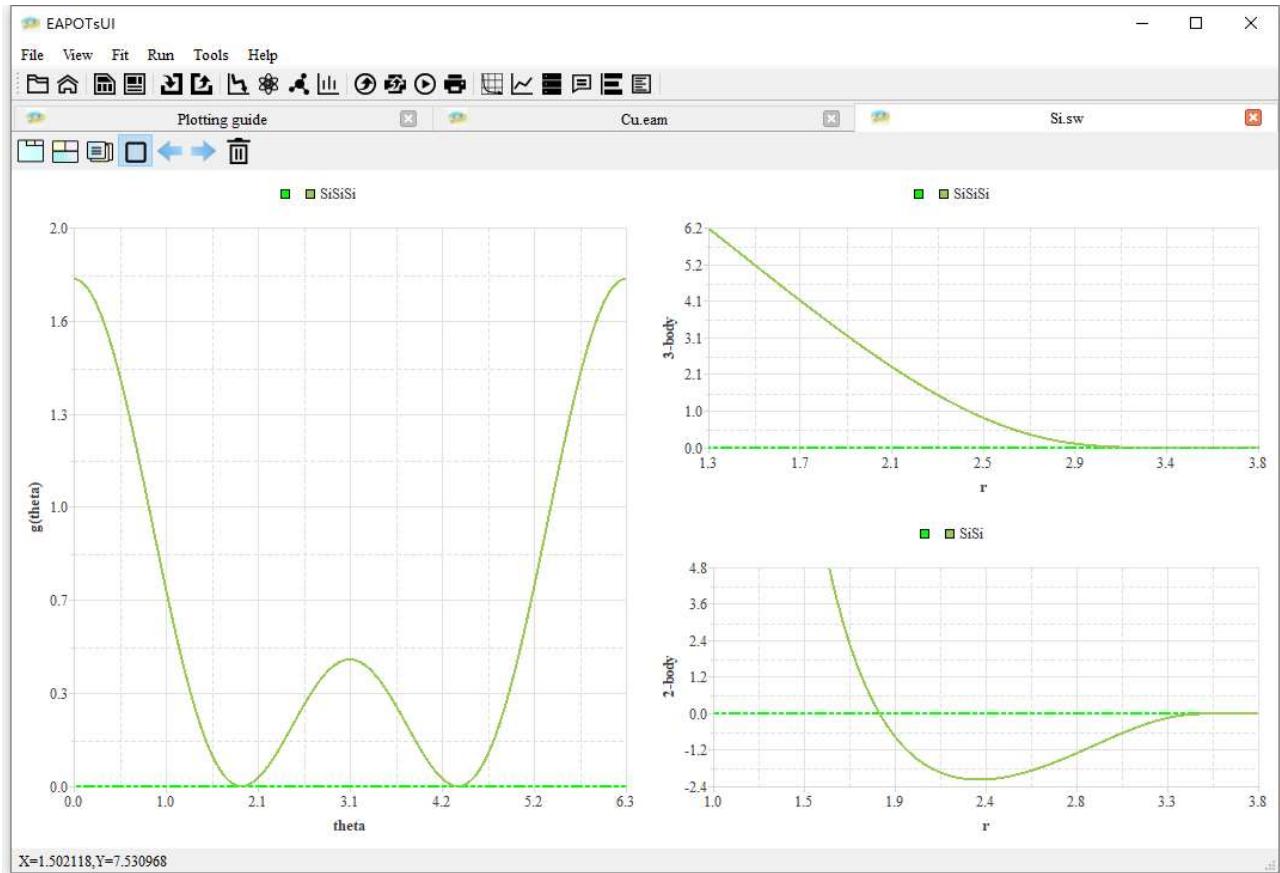


Fig. 27 Snapshot of SW potential

6.2. Potential energy surface

For energy surface visualization, Fig. 28 and Fig. 29 exhibit the typical γ surface of the compact {111} plane for Cu metal and the generalized SFE curve along with the [112] directions with marked unstable stacking fault energy (USFE). Here, the USFE represents the direct energy barrier required for the sliding between two neighboring atomic planes, and the intrinsic stacking fault energy (ISFE) represents indirect resistance by restricting the width of the stacking fault.

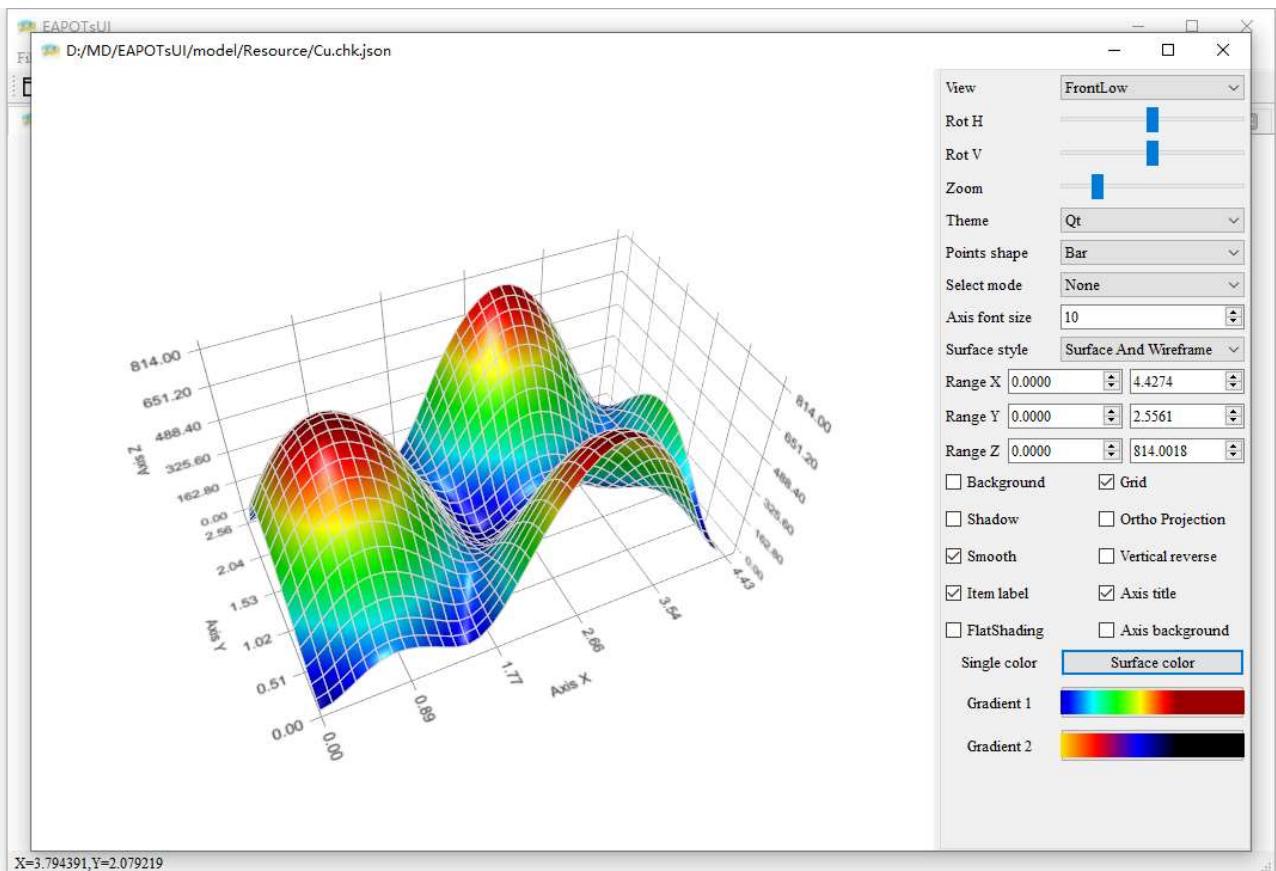


Fig. 28 The typical γ surface of the compact {111} plane for Cu metal

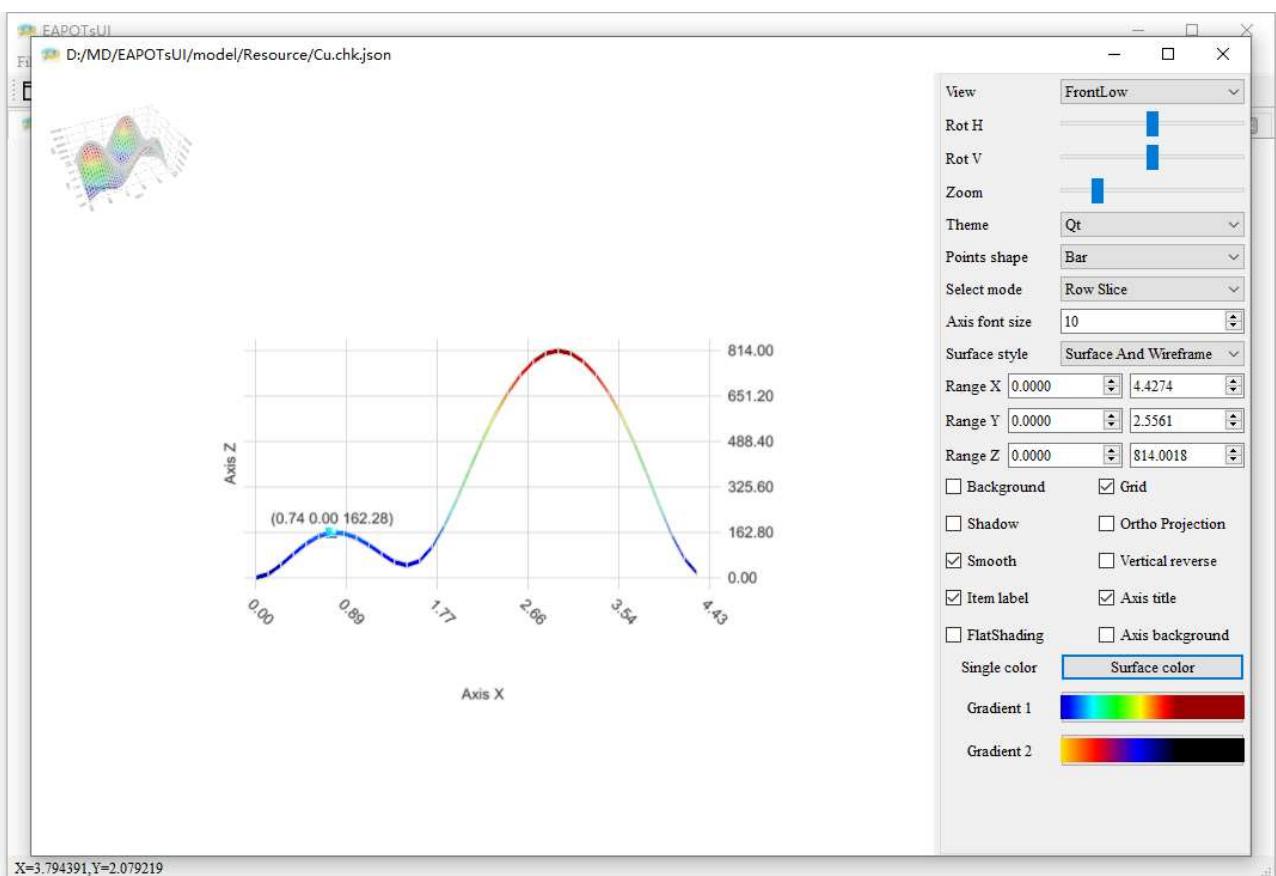


Fig. 29 The generalized SFE curve along with the [112] directions

7. Potential Model

7.1. EAM model

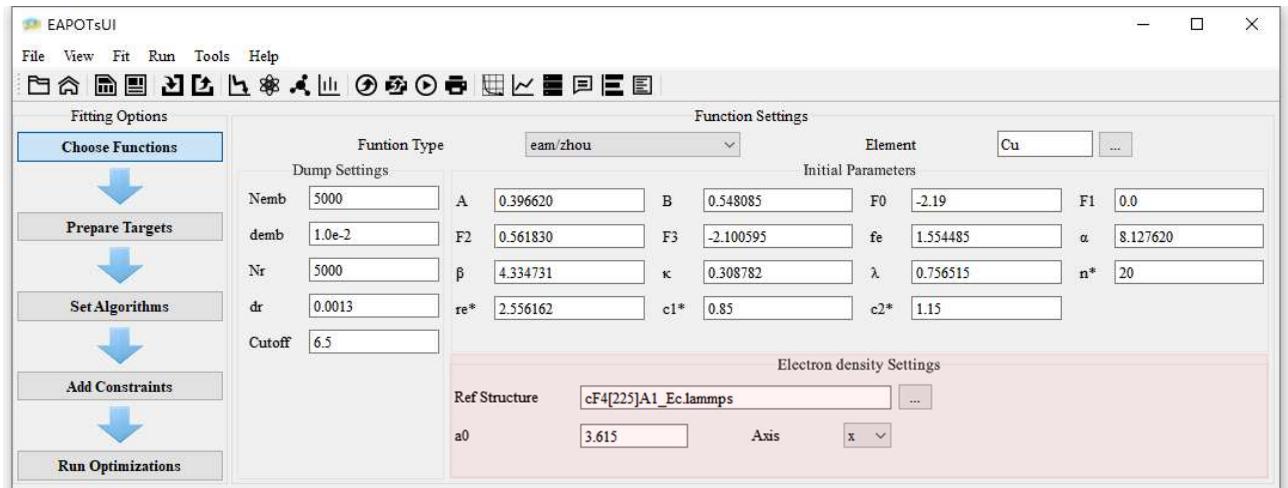


Fig. 30 Screenshot of potential model EAM-Zhou

For metallic systems, the most widely used pair functional model for describing interatomic interactions in materials science is the embedded atom method (EAM) [2-4]. Several EAM potentials forms have been developed with a similar expression, and the energy of an ion is defined by:

$$E_i = F \left(\sum_{i \neq j}^N \rho_{ij}(r_{ij}) \right) + \frac{1}{2} \sum_{i \neq j}^N \phi_{ij}(r_{ij}) \quad (7.1)$$

The equation has two parts: the first defines the energy for an ion i embed into the background electron charge density, while the second part is the pair interaction between the ions i and j separated by distance r_{ij} , and N is the number of neighbor atoms. The generalized potential model proposed by Zhou et al. [26], which can provide a reasonable approximation of the interactions between different metal elements, can be calculated with electronic density and pair potential in

$$\rho_{ij}(r_{ij}) = \frac{f_e \exp\left[-\beta\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \lambda\right)^n} \quad (7.2)$$

$$\phi_{ij}(r_{ij}) = \frac{A \exp\left[-\alpha\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \kappa\right)^n} - \frac{B \exp\left[-\beta\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \lambda\right)^n} \quad (7.3)$$

where $A, B, f_e, \alpha, \beta, \kappa$, and λ are the fitting parameters. The embedding energy function is expressed by three equations, which works well over a wide range of electron density ranges,

$$F(\rho) = \sum_{i=0}^3 F_{ni} \left(\frac{\rho}{\rho_n}\right)^i, \quad \rho < \rho_n, \quad \rho_n = 0.85\rho_e \quad (7.4)$$

$$F(\rho) = \sum_{i=0}^3 F_i \left(\frac{\rho}{\rho_e}\right)^i, \quad \rho_n \leq \rho < \rho_0, \quad \rho_0 = 1.15\rho_e \quad (7.5)$$

$$F(\rho) = F_e \left[1 - \ln\left(\frac{\rho}{\rho_s}\right)^\eta\right] \left(\frac{\rho}{\rho_s}\right)^\eta, \quad \rho_0 \leq \rho \quad (7.6)$$

where F_0, F_1, F_2 , and F_3 are the fitting parameters, and these equations must match values and slopes at their junctions for a smooth variation of the embedding energy. Besides, the equilibrium electronic dentistry ρ_e is based on a reference structure, as controlled by those parameters in the red region of Fig. 30.

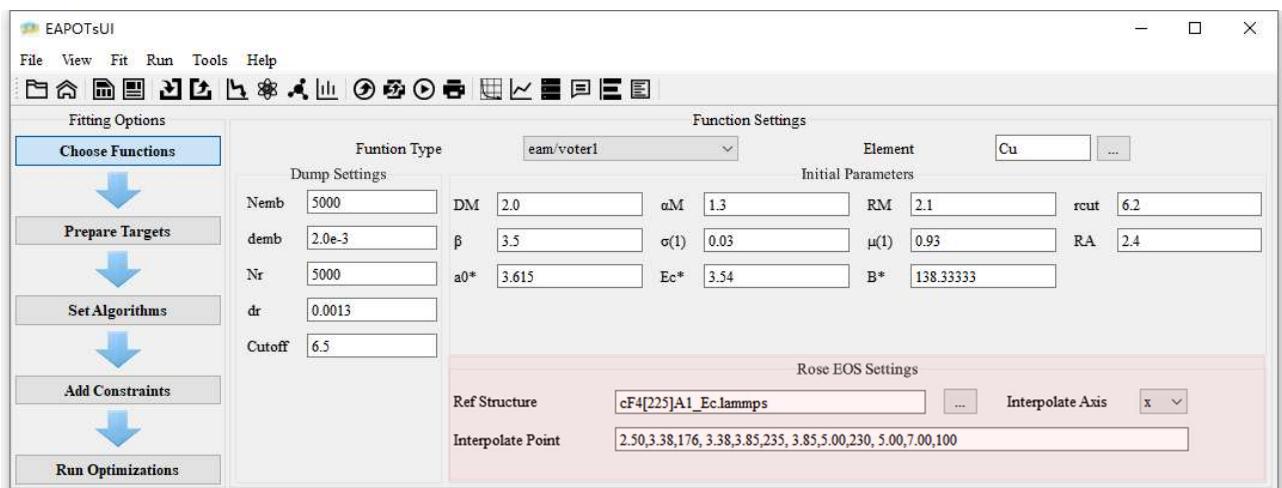


Fig. 31 Screenshot of potential model EAM-Voter

The potential form proposed by Onat et al. [27], which follows the scheme presented by Voter et al. [28], can also produce reliable predictions for the structural properties and energetics as well. Following their scheme, the charge distribution is present by:

$$\rho_{ij}(r) = \tanh(20r^2) \left\{ r^6 (e^{-\beta r} + 2^9 e^{-2\beta r}) + \frac{\delta}{\mu R_A} e^{-\frac{1}{2}[\mu(r-R_A)]^2} + 0.1\delta e^{-\frac{1}{2}[\mu(r-R_A-0.5)]^2} \right\} \quad (7.7)$$

where β , σ , μ , R_A in Eq. A2 are the fitting parameters. The pair interaction, on the other hand, was taken to be a Morse type potential

$$\phi_{ij}(r) = D_M [1 - e^{\alpha_M(r-R_M)}]^2 - D_M \quad (7.8)$$

here D_M , α_M , and R_M are fitting parameters. $\phi(r)$ and $\rho(r)$ had a smooth cutoff at $r = r_{cut}$ using the function

$$f_{smooth}(r) = h(r) - h(r_{cut}) + \left(\frac{r_{cut}}{m}\right) \left[1 - \left(\frac{r_{cut}}{m}\right)^m\right] \left(\frac{dh}{dr}\right) \Big|_{r=r_{cut}} \quad (7.9)$$

where $m = 20$. The embedding function was determined using the scheme introduced by Foiles et al [4], where the total energy of the fcc crystal is defined through the Rose universal equation of state (EOS) [29]:

$$E(x) = -E_{coh}(1+x)e^{-x} \quad (7.10)$$

where x is a function of the lattice spacing and given by

$$x(a) = \left(\frac{a}{a_0} - 1\right) \left(\frac{E_{coh}}{9\Omega_0 B}\right)^{-1/2} \quad (7.11)$$

Here, the parameters a_0 , Ω_0 , B and E_{coh} are the lattice constant, the atomic volume, the bulk modulus, and the cohesive energy, respectively, for the equilibrium fcc crystal at 0 K. To tabulate the embed energy (by interpolation), EAPOTs need to sample the

EOS equation based on a reference structure, as controlled by those parameters in the red region of Fig. 31, and the input format of the interpolation point is [start₁, end₁, point number₁, start₂, end₂, point number₂,...start_n, end_n, point number_n].

7.2. FS/TB model

Besides the EAM model, the pair functional models also include the tight-binding (TB) [5-7] and Finnis-Sinclair (F-S) [8] model. In the tight-binding model (TB) [6], the bonding energy in a metal system is approximatively proportional to the average bandwidth of the local electron density of the state. In particular, the second moment of the electron density of state can be written as a sum of squares of hopping integrals between the atoms and their neighbors [5, 7]:

$$E_{total} = - \sqrt{\sum_{i \neq j}^N \rho_{ij}(r_{ij}) + \frac{1}{2} \sum_{i \neq j}^N \phi_{ij}(r_{ij})} \quad (7.12)$$

where

$$\phi_{ij}(r_{ij}) = A \exp \left[-p \left(\frac{r}{r_0} - 1 \right) \right] \quad (7.13)$$

$$\rho_{ij}(r_{ij}) = \xi^2 \exp \left[-2q \left(\frac{r}{r_0} - 1 \right) \right] \quad (7.14)$$

here, the four parameters, p, q, A, and B, are assumed to depend only on the interacting atomic species i and j, and can be determined by fitting them to the physical properties. Although the TB potential initially seemed to apply to half-filled band metals, Ackland and Finnis have generalized the validity of the approach for band-filling metals [30]. The smoothly truncated potential form proposed by Li et al. [31, 32] was also

implemented in EAPOTs, which cut the long tail of the interatomic interaction by incorporating a truncation function into the pair-wise and local electronic density function.

In the Finnis-Sinclair (F-S) model [8], the total energy of an assembly of N atoms can be empirically written as

$$E_{total} = - \sqrt{\sum_{i \neq j}^N \rho_{ij}(r_{ij}) + \frac{1}{2} \sum_{i \neq j}^N \phi_{ij}(r_{ij})} \quad (7.15)$$

where

$$\phi_{ij}(r_{ij}) = (r - c)^2(c_0 + c_1 r_{ij} + c_2 r_{ij}^2) \quad (7.16)$$

$$\rho_{ij}(r_{ij}) = (r - d)^2 \quad (7.17)$$

where c_0 , c_1 , and c_2 are the fitting parameters, c and d represent the cutoff radii for each function. The FS potential has been used successfully to calculate the properties in several bcc metals. However, as clarified by Ackland et al. [33], the FS potential appears to give unphysical results for properties involving small interatomic separation, e.g., at high pressure, where the calculated energy is much lower than the experimental observation or theoretical prediction when the atoms are forced to be close together. To overcome this problem, Ackland et al. [33] added a product of a cubic and an exponential term to the pair-wise function as an additional repulsive term:

$$\phi_{ij}(r_{ij}) = (r - c)^2(c_0 + c_1 r_{ij} + c_2 r_{ij}^2) + B(a_0 - r_{ij})^3 e^{-\alpha r_{ij}} \quad (7.18)$$

The adjusted potentials are shown to predict a more realistic pressure-volume relationship for the V, Nb, Ta, Mo, and W metals [33].

7.3. BOP Tersoff/Brenner model

For covalent materials, such as semiconductors, we need to introduce those cluster potentials that contain the angular item for directional forces. One of the most commonly used empirical interatomic potentials in covalent materials is the Tersoff model [10, 11], where the total configuration energy of the center atoms is:

$$E_i = \frac{1}{2} \sum_{i \neq j}^N f_C(r_{ij}) [f_R(r_{ij}) + b_{ij} f_A(r_{ij})] \quad (7.19)$$

and

$$f_R(r_{ij}) = A \exp(-\lambda_1 r_{ij}) \quad (7.20)$$

$$f_A(r_{ij}) = -B \exp(-\lambda_2 r_{ij}) \quad (7.21)$$

$$f_C(r) = \begin{cases} 1 & , r < R - D \\ \frac{1}{2} - \frac{1}{2} \sin\left(\frac{\pi}{2} \frac{r - D}{D}\right) & , R - D < r < R + D \\ 0 & , r > R + D \end{cases} \quad (7.22)$$

$$b_{ij} = (1 + \beta^n \zeta_{ij}^n)^{-1/2n} \quad (7.23)$$

$$\zeta_{ij} = \sum_{k \neq i, j} f_C(r_{ik}) g(\theta_{ijk}) \exp[\lambda_3^m (r_{ij} - r_{ik})^m] \quad (7.24)$$

$$g(\theta) = \gamma_{ijk} \left(1 + \frac{c^2}{d^2} - \frac{c^2}{[d^2 + (\cos\theta - \cos\theta_0)^2]} \right) \quad (7.25)$$

where $f_R(r_{ij})$ a two-body term, $f_A(r_{ij})$ represents the three-body interactions, $g(\theta)$ is the modified angular-dependent term, and $f_C(r)$ is the cutoff function. Here i , j , and k label the atoms of the system, r_{ij} is the length of the ij bond, and θ_{ijk} is the bond angle between bonds ij and ik . The summations in the formula are overall neighbors j and k of atom i within a cutoff distance equal to $R + D$, and it has been shown that the simulation results with Tersoff potential accord well with the

experimental data [34], including defect structures, activation energies for defect motion, and coupling to strain.

The BOP-Brenner potential [12] is a reformulation of the Tersoff potential [10, 11], and the difference lies in the two-body parameters A , B , λ_1 , and λ_2 . These parameters have been rewritten in terms of the more physical formats, r_0 , D_0 , α , and S , which relate to dimer properties and the Pauling relation:

$$f_R(r_{ij}) = \frac{D_0}{S-1} \exp[-\alpha\sqrt{2S}(r_{ij} - r_0)] \quad (7.26)$$

$$f_A(r_{ij}) = \frac{SD_0}{S-1} \exp[-\alpha\sqrt{2/S}(r_{ij} - r_0)] \quad (7.27)$$

7.4. SW model

The SW potentials [9] have been extensively used to investigate semiconductor materials, which can also reproduce the buckling structure and lattice thermal conductivity. For particles with no net charge, the potentials are based on the idea that the force is repulsive if the atoms are very close and attractive, vanishing smoothly to zero when the distance between them increases. Compared with the Tersoff [10, 11] potential, the SW potential has a simpler form and fewer parameters to fit, so it is faster and more appropriate for parameter fitting:

$$E_i = \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j \neq ik} \sum_{k>j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) \quad (7.28)$$

$$\phi_2(r_{ij}) = A_{ij}\epsilon_{ij} \left[B_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{q_{ij}} \right] \exp \left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right) \quad (7.29)$$

$$\begin{aligned} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) &= \lambda_{ijk}\epsilon_{ijk} [\cos\theta_{ijk} - \cos\theta_{0ijk}]^2 \\ &\exp \left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \right) \exp \left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}} \right) \end{aligned} \quad (7.30)$$

Notice that the potential favors pairs of bonds with the desired tetrahedral angle when $\cos\theta_0 = 1/3$ and the other parameters were chosen to ensure that the diamond structure is the most stable at low pressure.

7.5. MEAM model

The pair functional models mentioned above are all radially symmetric, which do not reflect the directional nature of the bonding. To improve the performance of these potentials in this aspect, Baskes [13], Lee [14, 15] et al. have proposed the modified EAM (MEAM) potentials, where the local electron density is computed by considering the directionality of the bonding. The total MEAM configuration energy is a sum of direct contributions from all atoms:

$$E_i = F(\bar{\rho}_i) + \frac{1}{2} \sum_{i \neq j}^N \phi_{ij}(r_{ij}) \quad (7.31)$$

The first term corresponds to the embedding energy of the center atom, and the second term is the pair potential between atoms. The embedding function has the form

$$F(\bar{\rho}_i) = A_i E_i^0 \bar{\rho}_i \ln \bar{\rho}_i \quad (7.32)$$

and

$$\bar{\rho}_i = \frac{\bar{\rho}_i^{(0)}}{\rho_i^0} G_i \left(\sum_{k=1}^3 t_i^{(k)} \left(\frac{\bar{\rho}_i^{(k)}}{\bar{\rho}_i^{(0)}} \right)^2 \right) \quad (7.33)$$

$G_i(x)$ can be chosen as $\sqrt{1+x}$ or $\exp(x/2)$, and ρ_i^0 is the electronic scale factor. Each partial electron density term has the following form:

$$\left(\bar{\rho}_i^{(0)} \right)^2 = \left[\sum_{j \neq i} \rho_j^{a(0)}(r_{ij}) \right]^2 \quad (7.34)$$

$$\left(\bar{\rho}_i^{(1)}\right)^2 = \sum_{\alpha}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha}}{r_{ij}} \rho_j^{a(1)}(r_{ij}) \right]^2 \quad (7.35)$$

$$\left(\bar{\rho}_i^{(2)}\right)^2 = \sum_{\alpha}^3 \sum_{\beta}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha} r_{ij\beta}}{r_{ij}^2} \rho_j^{a(2)}(r_{ij}) \right]^2 - \frac{1}{3} \left[\sum_{j \neq i} \rho_j^{a(2)}(r_{ij}) \right]^2 \quad (7.36)$$

$$\begin{aligned} \left(\bar{\rho}_i^{(3)}\right)^2 &= \sum_{\alpha}^3 \sum_{\beta}^3 \sum_{\gamma}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha} r_{ij\beta} r_{ij\gamma}}{r_{ij}^3} \rho_j^{a(3)}(r_{ij}) \right]^2 \\ &\quad - \frac{3}{5} \sum_{\alpha}^3 \left[\sum_{j \neq i} \frac{r_{ij\alpha}}{r_{ij}} \rho_j^{a(3)}(r_{ij}) \right]^2 \end{aligned} \quad (7.37)$$

Here, $\bar{\rho}_i^{(k)}$ represent atomic electron densities from j atom at a distance r_{ij} from the center atom and $r_{ij\alpha}$ is the α component of the distance vector. The specific forms above are chosen so that the partial background electron densities are invariant to lattice translation and rotation, scale simply with atomic electron density for homogeneous deformation and they are computed as

$$\rho_i^{a(k)}(r_{ij}) = \rho_{i0} \exp \left[-\beta \left(\frac{r_{ij}}{r_i^0} - 1 \right) \right] f_c \left(\frac{r_c - r_{ij}}{\Delta r} \right) \prod_{j \neq i} S_{ijk} \quad (7.38)$$

where

$$S_{ijk} = f_c \left(\frac{C_{ijk} - C_{min,ijk}}{C_{max,ijk} - C_{min,ijk}} \right) \quad (7.39)$$

$$C_{ijk} = 1 + 2 \frac{r_{ij}^2 r_{ik}^2 + r_{ij}^2 r_{jk}^2 + r_{ik}^2 r_{jk}^2}{r_{ij}^4 - (r_{ik}^2 - r_{jk}^2)^2} \quad (7.40)$$

As clarified by Baskes et al. [13], the specific forms above can provide an improved description of systems where directional bonding is essential, such as semiconductors and elements from the middle of the transition metal series, where the origin EAM potentials have several shortcomings. Here we only introduce a short description of the inclusion of the angular items, and the complete MEAM expression can be accessed in Ref. [13-15]. Considering that the original MEAM model is not quite successful for bcc

metals, the second nearest-neighbor MEAM proposed by Lee et al. [14, 15] was also implemented in EAPOTs, which can reasonably reproduce the surface properties of the bcc transition metals.

Reference

- [1] S. Plimpton, Journal of Computational Physics 117 (1995) 1-19.
- [2] M.S. Daw, M.I. Baskes, Physical Review B 29 (1984) 6443-6453.
- [3] S.M. Foiles, Physical Review B 32 (1985) 7685-7693.
- [4] S.M. Foiles, M.I. Baskes, M.S. Daw, Physical Review B 33 (1986) 7983-7991.
- [5] F. Cleri, V. Rosato, Physical Review B 48 (1993) 22-33.
- [6] F. Ducastelle, F. Cyrot-Lackmann, Journal of Physics and Chemistry of Solids 31 (1970) 1295-1306.
- [7] V. Rosato, M. Guillope, B. Legrand, Philosophical Magazine A 59 (1989) 321-336.
- [8] M.W. Finnis, J.E. Sinclair, Philosophical Magazine A 50 (1984) 45-55.
- [9] F.H. Stillinger, T.A. Weber, Physical Review B 31 (1985) 5262-5271.
- [10] J. Tersoff, Physical Review B 37 (1988) 6991-7000.
- [11] J. Tersoff, Physical Review B 38 (1988) 9902-9905.
- [12] D.W. Brenner, Physical Review B 42 (1990) 9458-9471.
- [13] M.I. Baskes, Physical Review B 46 (1992) 2727-2742.
- [14] B.-J. Lee, M.I. Baskes, Physical Review B 62 (2000) 8564-8567.
- [15] B.-J. Lee, M.I. Baskes, H. Kim, Y. Koo Cho, Physical Review B 64 (2001) 184102.
- [16] Jeffrey, C, Lagarias, James, A, Reeds, Margaret, H, Wright, Paul, SIAM Journal on Optimization (1998).
- [17] E. Polak, G. Ribiere, Revue Francaise Information Recherche Operationnelle 3 (1969) 35-43.

- [18] M.J.D. Powell, The Computer Journal 7 (1965) 303-307.
- [19] Z. Zhan, J. Zhang, Y. Li, H.S. Chung, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39 (2009) 1362-1381.
- [20] R. Storn, K. Price, Journal of Global Optimization 11 (1997) 341-359.
- [21] A. Corana, M. Marchesi, C. Martini, S. Ridella, 13 (1987) 262–280.
- [22] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, in: M.A. Fischler, O. Firschein (Eds.) Readings in Computer Vision, Morgan Kaufmann, San Francisco (CA), 1987, pp. 606-615.
- [23] M. Wen, J. Li, P. Brommer, R.S. Elliott, J.P. Sethna, E.B. Tadmor, Modelling and Simulation in Materials Science and Engineering 25 (2016) 014001.
- [24] F. Ercolessi, J.B. Adams, Europhysics Letters (EPL) 26 (1994) 583-588.
- [25] <http://atztogo.github.io/spglib> (accessed 1 Feb 2021).
- [26] X.W. Zhou, R.A. Johnson, H.N.G. Wadley, Physical Review B 69 (2004) 144113.
- [27] B. Onat, S. Durukanoğlu, Journal of Physics: Condensed Matter 26 (2013) 035404.
- [28] A.F. Voter, S.P. Chen, MRS Proceedings 82 (1986) 175.
- [29] J.H. Rose, J.R. Smith, F. Guinea, J. Ferrante, Physical Review B 29 (1984) 2963-2969.
- [30] G.J. Ackland, M.W. Finnis, V. Vitek, Journal of Physics F: Metal Physics 18 (1988) L153-L157.
- [31] J.H. Li, X.D. Dai, T.L. Wang, B.X. Liu, Journal of Physics: Condensed Matter 19 (2007) 086228.
- [32] J.H. Li, Y. Dai, X.D. Dai, T.L. Wang, B.X. Liu, Computational Materials Science

43 (2008) 1207-1215.

[33] G.J. Ackland, R. Thetford, Philosophical Magazine A 56 (1987) 15-30.

[34] J. Tersoff, Physical Review Letters 64 (1990) 1757-1760.