1. Training a moment tensor potential (MTP).

1.1 Access to the MLIP package.

MLIP is a software package implementing MTP. It is distributed upon sending a reasonable request to Alexander Shapeev at a.shapeev@skoltech.ru.

1.2 Creating training sets.

Training sets are created by running ab-initio molecular dynamics (AIMD) at different temperatures using the *Vienna Ab-initio Simulation Package* (VASP)[1–3]. In the Mendeley dataset, the folder "`AIMD-inputs`", two samples of VASP input files (namely, `POSCAR`, `POTCAR`, `INCAR` and `KPOINTS`) for silicene monolayer and bulk silicon are included. After the completion of AIMD simulations, the `OUTCAR` file can be used to create the training set (`train.cfg`) with the following command:

```
./mlp convert-cfg OUTCAR train.cfg --input-format=vasp-outcar
```

This converts the configurations to a recognizable file format that is later used for training routine. The training set now contains the correlated configurations and can be reduced (subsampled) using the following command:

```
./mlp subsample train.cfg subsample.cfg 5
```

Here each one out of every 5 snapshots in the original "`train.cfg`" will be written to *"*`subsample.cfg`*"*. The subsampled training sets at different temperatures or structures should then be merged together to create the final training set, which can be achieved using the Linux `cat` *command*.

1.3 Training of MTPs.

Training of MTPs is done by solving the following minimization problem:

$$\sum_{k=1}^{K}\left[w_e\left(E_k^{\mathrm{AIMD}}-E_k^{\mathrm{MTP}}\right)^2+w_f\sum_{i}^{N}\left|f_{k,i}^{\mathrm{AIMD}}-f_{k,i}^{\mathrm{MTP}}\right|^2+w_s\sum_{i,j=1}^{3}\left|\sigma_{k,ij}^{\mathrm{AIMD}}-\sigma_{k,ij}^{\mathrm{MTP}}\right|^2\right]\rightarrow \min,$$

where E_k^{AIMD}, $f_{k,i}^{\mathrm{AIMD}}$ and $\sigma_{k,ij}^{\mathrm{AIMD}}$ are the energy, atomic forces, and stresses in the training set, respectively, and E_k^{MTP}, $f_{k,i}^{\mathrm{MTP}}$, and $\sigma_{k,ij}^{\mathrm{MTP}}$ are the corresponding values calculated with the MTP, K is the number of the configurations in the training set, N is the number of atoms in a configuration and w_e, w_f and w_s are the non-negative weights that express the importance of energies, forces, and stresses in the optimization problem, respectively, which in our study were set to 1, 0.1 and 0.001, respectively. We note that the weights for energy and stress are the default values.

As an example, the training of a MTP can be achieved using the following command:

mpirun -n n_cores ./mlp train p.mtp train.cfg --energy-weight=1 --force-weight=0.1 --stress-weight=0.001 --max-iter=3000 --curr-pot-name=p.mtp --trained-pot-name=p.mtp

Here "n_cores" is the number of cores used for parallel training of MTP, *"*p.mtp*"* is the input/output (curr-pot-name/trained-pot-name) MTP file, "train.cfg" is the training set in the internal *.cfg MLIP format, the option "max-iter" determines the maximum number of iterations in the optimization algorithm. The options "energy-weight", "force-weight", and "stress-weight", respectively, define the w_e, w_f and w_s weights explained earlier.

In our work, we conducted the passive training, by parameterizing the MTPs using the subsampled AIMD trajectories. In this approach, from the complete sets of AIMD configurations, only subsamples are selected for the training of first MTPs. Nonetheless, some critical configurations that could result in the improved accuracy of trained MTPs may have been missed in the created subsamples. Therefore, the accuracy of the developed MTP "p.mtp" over current subsampled training set "train.cfg" should once again be checked over the full AIMD configurations "trainF.cfg*"*, and the configurations with high extrapolations grades [4] will be selected, and will written to the file "trainN.cfg*", via the following command:*

./mlp select-add p.mtp train.cfg trainF.cfg trainN.cfg

The selected configurations "trainN.cfg*"* should be added to the original training sets "train.cfg" and the final MTP will developed by retraining of new clean potentials over the updated training set. This way, the efficient use of conducted AIMD simulations will be guaranteed.

1.4 Structure of MTPs.

MTP belongs to the family of machine-learning interatomic potentials by which potentials show flexible functional form that allows for systematically increasing the accuracy with an increase in the number of parameters and the size of the training. In the folder "Untrained-MTPs", we included three samples of clean MTPs. Depending on the number of parameters, the appropriate MTP should be chosen. Prior to training, there are some parameters to be adjusted, such as the "species_count", "min_dist" and "max_dist" which, respectively, define the number of elements in the system, minimum atomic distance and cutoff distance of the potential. Like the classical potentials, by increasing the cutoff distance more neighbors will be included in the calculations which accordingly increase the computational costs. The number of parameters in a MTP can be calculated via:

$$\mathbf{species\_count}^2 \cdot \mathbf{radial\_basis\_size}^2 \cdot \mathbf{radial\_funcs\_count} + \mathbf{alpha\_scalar\_moments} + \mathbf{1}$$

Note that "radial_funcs_count" and "alpha_scalar_moments" are the fixed features of a particular MTP and only "radial_basis_size" can be manually changed to adjust the number of constants.

2. Evaluation of phononic properties using the MTPs.

In our previous work [5], we included the full details and numerous examples for the evaluation of phononic properties using the MTP and PHONOPY [6] package in a public Mendeley dataset, please refer to: http://dx.doi.org/10.17632/7ppcf7cs27.1

3. MTP/ShengBTE interface.

ShengBTE [7] is a package for computing the lattice thermal conductivity on the basis of a full iterative solution to the Boltzmann transport equation. Its main inputs are sets of second- and third-order interatomic force constants and a CONTROL file for the adjustment of computational details. In this work, the calculation of anharmonic interatomic force constants is substantially accelerated by substituting DFT simulations with the MTP-based solution. For the calculation of anharmonic interatomic force constants, ShengBTE [7] provides a script, *"thirdorder.py"*, implementing a real-space supercell approach to anharmonic IFC calculations. In this approach, according to the defined supercell size and cutoff distance, the input geometries for the force constant calculations will be generated. For compatibility with "cfg"-file format, the "thirdorder_mtp.py" is developed using the original *"thirdorder_vasp.py"*. Moreover, we developed an additional script "fake_vasp_calcs.py", which uses the MTP-based calculated forces and artificially create the VASP output files of "vasprun.xml". This approach provides the possibility of direct comparison of forces by MTP and VASP. These developed two python scripts are included the folder "MTP_ShengBTE_py".

In the folder "Examples", complete input files are included for every structure. In this case the subfolder "ShengBTE-inputs" includes the complete input files for the ShengBTE solution (namely: CONTROL, FORCE_CONSTANTS_2ND and FORCE_CONSTANTS_3RD). Using the data provided in the subfolder called "Anharmonic-MTP", the anharmonic interatomic force constants can be obtained using the trained MTPs "p.mtp". MTP/ShengBTE interface follows the same routine as that of the VASP/ShengBTE, explained in the ShengBTE documentation. To facilitate the practical usage, for every example we included a shell script, named "getFC.sh". In the aforementioned script, "supcell" and "Cutoff" are respectively, the supercell size and cutoff neighbour for the evaluation of anharmonic force constants on the basis of primitive unitcell "POSCAR". Please note that "POSCAR", "p.mtp", "mlip.ini" and related python scripts should be located in this folder for complete calculations.

References

1. Kresse, G. & Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **6**, 15–50 (1996).
2. Kresse, G. & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169–11186 (1996).
3. Kresse, G. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B* **59**, 1758–1775 (1999).
4. Podryabinkin, E. V & Shapeev, A. V. Active learning of linearly parametrized interatomic potentials. *Comput. Mater. Sci.* **140**, 171–180 (2017).
5. Mortazavi, B. *et al.* Exploring phononic properties of two-dimensional materials using machine learning interatomic potentials. *Appl. Mater. Today* **20**, 100685 (2020).
6. Togo, A. & Tanaka, I. First principles phonon calculations in materials science. *Scr. Mater.* **108**, 1–5 (2015).
7. Li, W., Carrete, J., Katcho, N. A. & Mingo, N. ShengBTE: A solver of the Boltzmann transport equation for phonons. *Comput. Phys. Commun.* **185**, 1747–1758 (2014).