In this manual we provide a guide on the practical implementation and reproduction of the results presented in our publication entitled: "Exploring Phononic Properties of Two-Dimensional Materials using Machine Learning Interatomic Potentials". We specifically discuss the repository https://gitlab.com/ivannovikov/mlip_phonopy with the MLIP_PHONOPY code—a C++ interface between the MLIP code and the PHONOPY software—which allows one to calculate phonon spectra, group velocities, thermal properties, etc., of a two-dimensional material. Along with the repository description, this manual contains an instruction on quick installation of the stable branch of the MLIP code, a description of VASP input files for ab initio molecular dynamics (AIMD) calculations (namely, the folders Structures and VASP-inputs) and training set preparation, an instruction on passive training of Moment Tensor Potentials (MTPs) using the MLIP code. Finaly, we describe the folders Untrained-MTPs and Examples with the additional files available here: http://dx.doi.org/10.17632/7ppcf7cs27.1.

1 Quick installation of the MLIP code and the mlp binary file

MLIP is a software package implementing MTP. It is distributed upon sending a reasonable request to Alexander Shapeev at a.shapeev@skoltech.ru. In order to download the code through git use

```
git clone https://gitlab.com/ashapeev/mlip-v2.git/
```

The command will create the mlip-v2/ folder with the source files. For the MLIP software installation execute (in the terminal)

```
./configure --no-mpi
make mlp
```

These commands will generate the serial version of the MLIP code, namely, the binary file mlp in the bin/ folder. This file allows to run various MLIP commands. The binary file is self-explanatory, i.e., to list all the MLIP commands execute ./mlp list, to read the manual about the commands ./mlp help. Typical template for executing the command is

```
./mlp [command] [input/output files] [options]
```

2 VASP input files for AIMD calculations and training sets

In the folder VASP-inputs we include the samples of VASP AIMD input files (namely, POSCAR, POTCAR, INCAR and KPOINTS) for the case of graphene. The folder Structures contains POSCAR files for different 2D materials we deal with during this study. After the completion of AIMD simulations, OUTCAR file can be used to create a training set with the following command:

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

Using the aforementioned command all the snapshots will be included in the training set. Shortening of the training set (creating subsamples) can be achieved using the following command:

```
./mlp subsample train.cfg subsample.cfg 10
```

In the mentioned case, each first of every 10 snapshots in the original train.cfg will be written to the subsample.cfg file.

3 Passive training of MTP

For MTP training we use the parallel version of the `MLIP` code (if there is no parallel compiler in a user computer it is also possible to use the serial version for training). In order to compile this version execute:

```
./configure
make mlp
```

These commands will generate the parallel version (the `mlp` binary file) of the `MLIP` code if the `configure` file find an `mpi`-compiler (appropriate `module` should be loaded). We run passive training with the `train` command, here is the example:

```
mpirun -n n_cores ./mlp train p.mtp train.cfg --energy-weight=1 --force-weight=0.1
--stress-weight=0.001 --max-iter=2000 --curr-pot-name=p.mtp --trained-pot-name=p.mtp,
```

where `n_cores` is the number of cores used for parallel passive training of MTP, `p.mtp` is the input/output (`curr-pot-name`/`trained-pot-name`) MTP file (we also added `Untrained-MTPs` folder with the MTPs untrained), `train.cfg` is the training set in internal `*.cfg` MLIP format for configurations (see the `manual.pdf` in the `doc/manual/` folder), the option `max-iter` determines the maximum number of iterations in the optimization algorithm (in our software we use BFGS algorithm). The options `energy-weight`, `force-weight`, and `stress-weight` define the weights w_e, w_f, and w_s in the minimization problem (1):

$$\sum_{k=1}^{K}\left[w_\mathrm{e}\left(E_k^\mathrm{AI}-E_k^\mathrm{MTP}\right)^2+w_\mathrm{f}\sum_{i=1}^{N}\left|f_{k,i}^\mathrm{AI}-f_{k,i}^\mathrm{MTP}\right|^2+w_\mathrm{s}\sum_{i,j=1}^{3}\left(\sigma_{k,ij}^\mathrm{AI}-\sigma_{k,ij}^\mathrm{MTP}\right)^2\right]\to\min, \tag{1}$$

where E_k^AI, $f_{k,i}^\mathrm{AI}$, and $\sigma_{k,ij}^\mathrm{AI}$ are the ab initio energy, forces, and stresses of the k-th configuration in the training set, E_k^MTP, $f_{k,i}^\mathrm{MTP}$, and $\sigma_{k,ij}^\mathrm{MTP}$ are the same quantities calculated with MTP, K is the number of the configurations in the training set, w_e, w_f, and w_s are non-negative weights that express the importance of energies, forces, and stresses in (1).

4 Quick installation and usage of MLIP_PHONOPY code

For calculations of phonon dispersion relations, group velocities and other related thermal properties with MTPs we include the required files and scripts to the repository. In order to download the repository use the command:

```
git clone https://gitlab.com/ivannovikov/mlip_phonopy.git.
```

The command will create the `mlip_phonopy/` folder with the following source files:

`Main.cpp` — a source C++ code which creates the `FORCE_SETS` input files for `PHONOPY`;
`Makefile` — the file allows to compile the `Main.cpp` source file and create the `Main` binary file.

It is necessary to put this folder to the same folder as the `mlip-v2/` folder (with the `MLIP` code). After the compilation of the `MLIP` software described above, one can compile the C++ code by executing the command `make` in the terminal. This command will create the binary file `Main` which allows to generate the `FORCE_SETS` input files for `PHONOPY`.
In the same `mlip_phonopy/` folder there is the folder `Example` with four input files needed to calculate phonon dispersions, group velocities, etc. of a material using the combination of MTPs and `PHONOPY`:

`getPhonon.sh` — the main shell script which calculates various properties of a material using the `Main` file and `PHONOPY`, input arguments are path to the folder with the `phonopy` and `phonopy-bandplot` binary files and path to the folder with the `Main` binary file (see the description of the script in the section below);
`p.mtp` — the file with the MTP trained for a specific material;
`band.conf` — the file with the k-path and various settings for `PHONOPY` (see the structure of this file in the next section);
`POSCARunitcell` — the file with unit cell in `VASP POSCAR` format (in direct coordinates).

In order to calculate properties of any material (in the repository we consider WTe_2) run the main script :

```
./getPhonon.sh path_to_phonopy/ path_to_Main_file_in_mlip_phonopy/.
```

5 Practical informations to acquire phononic properties

To facilitate the future studies we include the sample of `band.conf` file we used for the case of graphene (see PHONOPY documentation https://atztogo.github.io/phonopy/ for detailed information):

```
ATOM_NAME = C
DIM = 8 8 1
BAND = 0.0 0.0 0.0    0.5 0.0 0.0   0.666666 0.33333 0.0    0.0 0.0 0.0
GROUP_VELOCITY = .TRUE.
BAND_CONNECTION = .TRUE.
BAND_POINTS = 101
WRITE_FORCE_CONSTANTS = .TRUE.
MESH = 18 18 1
GAMMA_CENTER = .TRUE.
TPROP = .TRUE.
TMAX = 1000
TSTEP= 20
```

As discussed in the previous section, to calculate phononic properties of any material using trained MTPs, the following files should exist in the desired folder:
`p.mtp`;
`band.conf`;
`POSCARunitcell`;
`getPhonon.sh` (the shell script).
Here we briefly explain the procedure in the shell script for the case of graphene. Using the following commands the properties (e.g., phonon spectra, group velocity) specified in `band.conf` file will be calculated by PHONOPY:

```
"$PHONOPY_PATH"phonopy -d --dim="8 8 1" -c POSCARunitcell
mv SPOSCAR POSCAR
num=$(find -type f -name "POSCAR-*"|wc -l)
"$MLIP_PHONOPY_PATH"Main $num
cp POSCARunitcell POSCAR
"$PHONOPY_PATH"phonopy  -p -s band.conf
```

To extract the phonon dispersion to a text file, one can use the `phonopy-bandplot` executable file provided by PHONOPY:

```
"$PHONOPY_PATH"phonopy-bandplot  --gnuplot band.yaml > Band.txt
```

The group velocities at every frequency can be also extracted using:

```
grep " frequency" band.yaml  > frequency.txt
grep "group_velocity" band.yaml  > gv.txt
```

We have included the folder `Examples` which contains the main results for all the considered examples in our manuscript. Therefore, using these data one can reproduce the results presented in our manuscript. For each considered monolayer, apart from the required files explained in the previous section, we included:

`vasprun.xml` — a VASP output file for the DFPT calculations;
`train.cfg` — a training set created with the AIMD simulations.