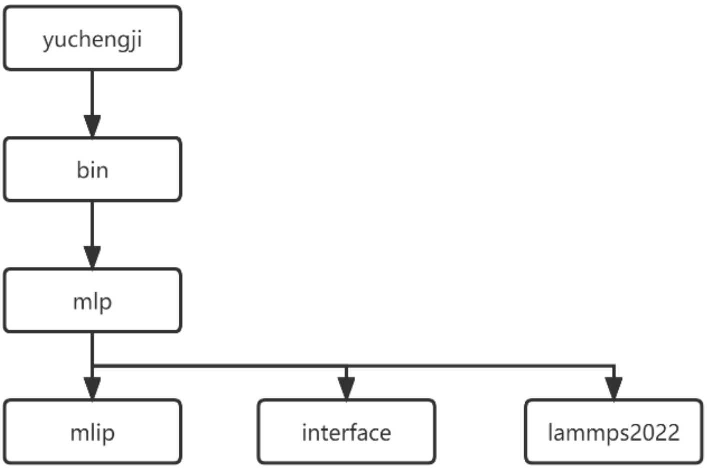
# Installation manual for MLIP (Machine learning interatomic potential)

MLIP developed by Prof. Alexander Shapeev

Manual wrote by Yucheng Ji on November 15th 2022

Update on March 01st 2023

The directory structure:



- MLIP

1. Download the source file from Gitlab (Permission required).

   下载 MLIP 源文件，该文件下载需要经过作者许可；

   https://gitlab.com/ashapeev/mlip-2

2. Upload the source file to the destination path <MLIP> in your account.

   上传源文件至目标路径 <MLIP>

   ```
   tar -zxvf mlip-2-master
   mv mlip-2-master mlip
   ```

3. Prepare the file and environments

   预备文件目录及软件环境

   ```
   module load 2022r2 intel/oneapi-all                    (DelftBlue)
   module load  2022 intel-2022a                          (Snellius)
   ```

   Attention:

   If the version **2017 or 2018** Intel compiler is available, please use load this compiler!

4. Compile the source code

   编译配置文件

```
cd mlip
./configure
```

```
    Configure option:
    --prefix=<path>                installation prefix path (e.g., /home/...)
    --prefix_fortran=<path>        install path to fortran
    --no-mpi                       disable MPI libraries [autodetect]
    --no-selftest                  disable selftest implementation
    --lossless-cfg                  turn on lossless configuration file writing
    --compiler=[intel|gnu]         which compilers to use [autodetect]
    --blas=NAME                    BLAS library [autodetect]
    --program-suffix=<suffix>      program name suffix
```

   successfully created information

   成功创建信息

```
    Generating makefile configuration ...
    Configuration complete.
```

5. compile the binary file

   编译二进制文件

```
make mlp
```

   there is no successfully created information, but the mlp compiled file can be found in the path as follows.

   未有成功创建信息，但在以下路径可查询到 mlp 文件

```
./bin/mlp
```

6. Shortcomings of this solution

   本方案的不足之处

   a. Using mlp command, all testing cannot be passed!

      使用 mlp 命令所有的测试均不通过！

```
make test
```

   The output error is as follows,

   输出错误如下

```
    INTEGRATION TESTING:
    Running test 00.convert_vasp_outcar
    Abort(1090831) on node 0 (rank 0 in comm 0): Fatal error in PMPI_Init: Other MPI
error, error stack:
```

```
MPIR_Init_thread(176):
MPID_Init(1430)......:
MPIR_pmi_init(167)...: PMI2_Job_GetId returned 14
make[1]: *** [Makefile:24: /home/yuchengji/bin/mlp/mlip/test/examples/00.convert_vasp_outcar/test.sh] Error 1
make: *** [Makefile:173: test] Error 2
```

b. The normal mlp command should be used as follows,

常规 mlp 命令可使用如下：

```
mpirun -np 1 mlp convert-cfg --input-format=vasp-outcar OUTCAR out/relax.cfg
```

Instruction:

说明：

All commonly used mlp commands should be preceded by the beginning *mpirun -np 1*, these commands including convert-cfg, calc-error, mindist, calc-efs, calc-grade.

But the training mlp should be described according to the cores, such as *srun* or *mpirun -np 96*.

c. If the version of Intel compiler is 2017 or 2018, these shortcomings can be avoid.

使用 2017 或 2018 版本的 Intel 编译器可能会避免上述问题

7. Compile the libinterface for lammps

为 lammps 编译 mlip 接口

```
make libinterface
```

successfully created information

成功创建信息

```
ar: creating lib/lib_mlip_interface.a
```

8. Download the interface software from the Gitlab (Permission required)

下载 interface 接口程序，需要作者允许

```
cd /home/yuchengji/bin/mlp/
git clone https://gitlab.com/ashapeev/interface-lammps-mlip-2.git interface
```

9. Move the lib_mlip_interface.a file to the interface directory

移动生成的 mlip 库至 interface 文件夹

```
cd /home/yuchengji/bin/mlp/mlip/lib
mv lib_mlip_interface.a ../../interface/
```

- LAMMPS

1. Download the lammps package

   下载lammps源文件

   ```
   cd /home/yuchengji/bin/mlp/
   git clone -b stable https://github.com/lammps/lammps.git lammps2022
   ```

2. Choose the package required

   选择LAMMPS运行所需的包

   ```
   cd /home/yuchengji/bin/mlp/lammps2022/src
   make package-status
   make yes-MC
   make yes-MEAM
   make yes-manybody
   ```

- Interface

1. Installing the mlip-lammps interface

   安装mlip与lammps接口程序

   ```
   cd /home/yuchengji/bin/mlp/interface
   ./install.sh ../lammps2022 intel_cpu_intelmpi
   ```

   successfully created information

   成功创建信息

   ```
     text      data      bss      dec      hex   filename
   10053427          235288  14312 10303027          9d3633            ../lmp_intel_cpu_intelmpi
   make[1]: Leaving directory
   '/home/yuchengji/bin/mlp/lammps2022/src/Obj_intel_cpu_intelmpi'
   ```

**Yucheng Ji** PhD candidate
**MSE, 3mE, TU Delft**
E-mail: Y.Ji-2@tudelft.nl
**BAIC for MGE, USTB**
E-mail: jiyucheng@xs.ustb.edu.cn

北京科技大学
University of Science and Technology Beijing

TU Delft
Delft University of Technology

# Utilization manual for MLIP (Machine learning interatomic potential)

MLIP developed by Prof. Alexander Shapeev
Manual wrote by Yucheng Ji
November 2022

- Potential training

1. Transfer the VASP/OUTCAR to cfg file

转化 OUTCAR 为 MLIP cfg 文件

```
mlp convert-cfg --input-format=vasp-outcar OUTCAR Sc3Al.cfg
```

Note: All structure optimization step will be extracted into cfg file.

注释：所有的在优化过程中的结构都会被考虑<一定变形>

The file format:

文件格式

```
BEGIN_CFG
 Size
    6
 Supercell
         7.604607      0.000002     -0.000001
         0.000002      7.604608     -0.000007
        -0.000001     -0.000007      7.604623
 AtomData:  id type       cartes_x      cartes_y      cartes_z           fx          fy          fz
             1    0       2.851730      0.950570      2.851730    -0.000019    0.000003    0.000030
             2    0       6.654030      6.654030      0.950570     0.000024   -0.000021    0.000001
             3    0       4.752880      6.654030      2.851730    -0.000039   -0.000039    0.000021
             4    0       0.950580      0.950580      0.950580     0.000030   -0.000015   -0.000009
             5    0       2.851730      4.752880      6.654040    -0.000005   -0.000011   -0.000031
             6    0       6.654030      2.851730      4.752890     0.000005    0.000021   -0.000009
 Energy
       -122.177426730000
 PlusStress:  xx          yy          zz          yz          xz          xy
        -0.33700    -0.33706    -0.33776     0.00029     0.00010    -0.00009
 Feature   EFS_by       VASP
 Feature   mindist      2.688629
END_CFG
```

2. Summarize all cfg files to one train.cfg

   综合所有的 cfg 文件成为一个文件

```
cat *.cfg >> train.cfg
```

3. Calculate the minimum distance in the train.cfg file

   计算 train.cfg 文件中的原子最小间距

```
mlp mindist train.cfg
```

4. Copy and modify the initial.mtp file to the training directory

   拷贝并编辑初始 mtp 文件

```
cp /home/yuchengji/bin/mlp/mlip/untrained_mtps/16.mtp .
vim 16.mtp
```

   The file format:

   文件格式

```
MTP
version = 1.1.0
potential_name = MTP1m
species_count = 1                # modify the number of elements
potential_tag =
radial_basis_type = RBChebyshev
    min_dist = 2.38362           # modify the number to minimum distance
    max_dist = 5                 # modify the cut off distance
    radial_basis_size = 8        # modify the number of radial basis functions,  8,6,10,4,12
    radial_funcs_count = 2
    radial_funcs_count = 4
    alpha_moments_count = 350
alpha_index_basic_count = 130
```

```
Esc： wq
```

5. Compile the submit script file <Cluster required, e.g. DelftBlue>

   编辑 submit 脚本<依据超算改变>

```
vim sub.sh
```

   The content of submit script:

   提交脚本内容

```
#!/bin/bash -l
#SBATCH --job-name="mlp"
#SBATCH --account=research-3me-mse
#SBATCH --partition=compute
#SBATCH -t 24:00:00
```

```
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=24
#SBATCH --mem-per-cpu=2G

module load 2022r2
module load intel/oneapi-all

export OMP_NUM_THREADS=1                                    # Always be kept equal 1

srun mpirun -np 24 mlp train 16.mtp train.cfg --trained-pot-name=AlScCu.mtp
```

`Esc : wq`

6. Other train options provided in the mlp

   Train 命令中其他选项

```
Options
--energy-weight=<double>: weight of energies in the fitting. Default=1
--force-weight=<double>: weight of forces in the fitting. Default=0.01
--stress-weight=<double>: weight of stresses in the fitting. Default=0.001
--scale-by-force=<double>: Default=0. If >0 then configurations near equilibrium (with roughtly force < <double>) get more weight.
--valid-cfgs=<string>: filename with configuration to validate
--max-iter=<int>: maximal number of iterations. Default=1000
--curr-pot-name=<string>: save potential on each iteration if <string> not empty
--trained-pot-name=<string>: filename for trained potential. Default=Trained.mtp_
--bfgs-conv-tol=<double>: stop if error dropped by a factor smaller than this over 50 BFGS iterations. Default=1e-3
--weighting=<string>: how to weight configuration with different sizes relative to each other. Default=vibrations. Other=molecules, structures
--init-params=<string>: how to initialize parameters if a potential was not pre-fitted. Default is random. Other is same - this is when interaction of all species is the same (more accurate fit, but longer optimization)
--skip-preinit: skip the 75 iterations done when parameters are not given
--update-mindist: updating the mindist parameter with actual minimal interatomic distance in the training set
```

7. Submit the task to the cluster

   提交任务至超算

   `sbatch sub.sh`

   output from mlp

   mlp 程序输出

```
MTPR from 16.mtp, Database: train.cfg
Random initialization of radial coefficients
Rescaling...
  scaling = 0.833333333333333, condition number = 2763.67306759096
  scaling = 0.909090909090909, condition number = 3673.61312437045
  scaling = 1, condition number = 5351.11880703349
  scaling = 1.1, condition number = 7834.47326839507
  scaling = 1.2, condition number = 10362.1760849044
Rescaling to 0.833333333333333... done
Rescaling...
  scaling = 0.694444444444445, condition number = 1637.70358193521
  scaling = 0.757575757575758, condition number = 2042.48285004521
  scaling = 0.833333333333333, condition number = 2763.71502718135
  scaling = 0.916666666666667, condition number = 3778.30974868583
  scaling = 1, condition number = 5351.11880301559
Rescaling to 0.694444444444445... done
Rescaling...
  scaling = 0.578703703703704, condition number = 942.573869338707
  scaling = 0.631313131313131, condition number = 1230.54333481382
  scaling = 0.694444444444445, condition number = 1637.70358201451
  scaling = 0.763888888888889, condition number = 2093.96259982992
```

- MD Simulation without Active Learning

1. File required

   所需文件

| | | |
|---|---|---|
| AlScCu.mtp | # potential | |
| mlip.ini | # potential | control file |
| train.cfg | # potential | original training file |
| | | |
| data.Al | # lammps | structure file |
| rdf.in | # lammps | control file |

2. The content of mlip.ini

   mlip.ini 文件内容

```
mtp-filename    all.mtp
calculate-efs    FALSE
select           FALSE
#      select:threshold   2.1
#      select:threshold-break  100.0
#      select:save-selected preselect.cfg
#       select:load-state state.als
```

3. Compile the submit script for MLIP-MD simulation

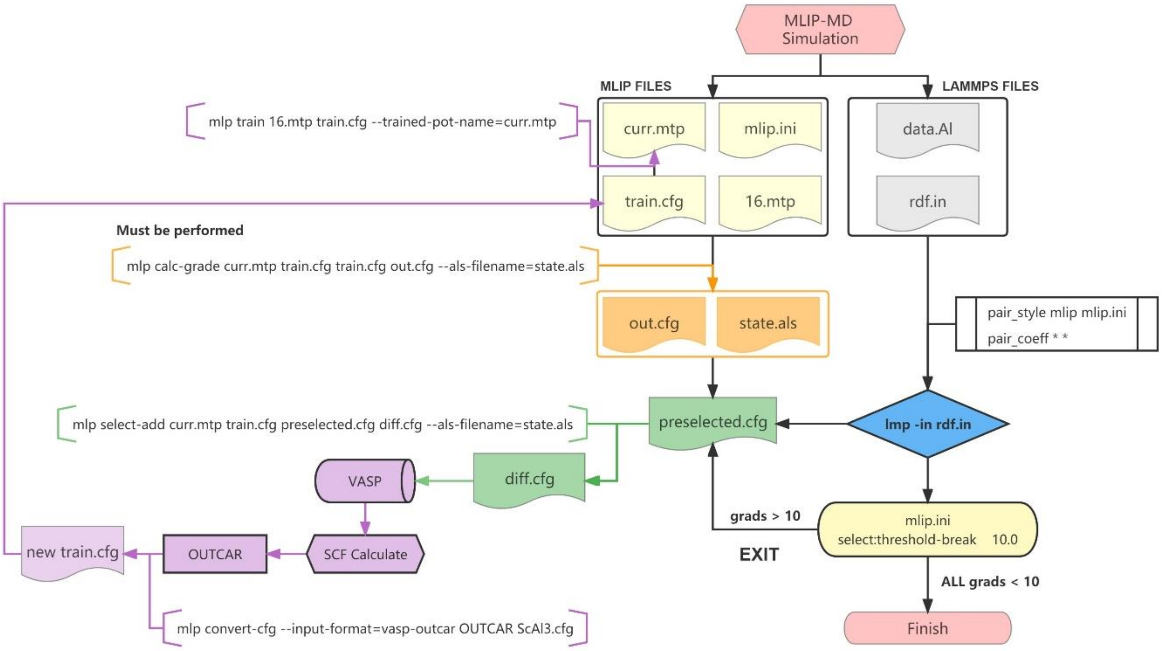
   MLIP-MD 模拟的提交脚本

```
#!/bin/bash
#SBATCH --job-name="lmp_no_al"
#SBATCH --account=research-3me-mse
#SBATCH --partition=compute
#SBATCH -t 72:00:00
#SBATCH --nodes=4
#SBATCH --ntasks-per-node=12
#SBATCH --mem-per-cpu=5G

module load 2022r2
module load intel/oneapi-all

export OMP_NUM_THREADS=1

srun lmp_intel_cpu_intelmpi -in rdf.in
```

## ➢ MD Simulation with Active Learning

The flow of MLIP active learning.

MLIP 主动学习流程



```
mlp train 16.mtp train.cfg --trained-pot-name=curr.mtp
```

Must be performed

```
mlp calc-grade curr.mtp train.cfg train.cfg out.cfg --als-filename=state.als
```

```
mlp select-add curr.mtp train.cfg preselected.cfg diff.cfg --als-filename=state.als
```

```
mlp convert-cfg --input-format=vasp-outcar OUTCAR ScAl3.cfg
```

```
pair_style mlip mlip.ini
pair_coeff * *
```

```
mlip.ini
select:threshold-break    10.0
```

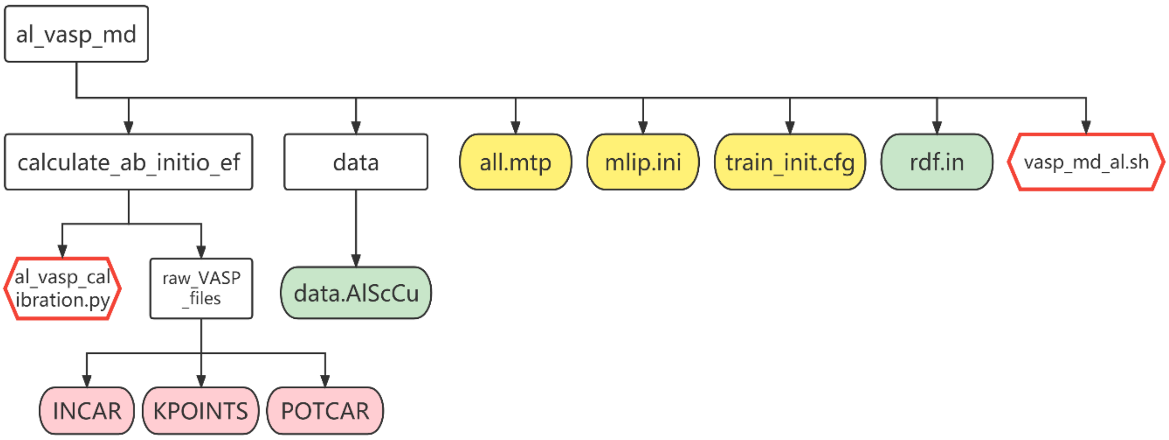
1. File required

所需文件

| | | |
|---|---|---|
| all.mtp | # potential | |
| mlip.ini | # potential | control file |
| train.cfg | # potential | original training file |
| data.Al | # lammps | structure file |
| rdf.in | # lammps | control file |
| vasp_md_al.sh | # script | active learning script |
| al_vasp_calibration.py | # script | create vasp model |
| calculate_ab_initio_ef | # directory | vasp calculation |
| calculate_ab_initio_ef/raw_VASP_files | # directory | vasp INCAR KP… |
| data | # directory | lammps data file |

The directory structure of active learning

主动学习目录结构



1. Compile the vasp_md_al.sh file

编辑主动学习脚本

```
vim vasp_md_al.sh
```

The content of vasp_md_al.sh file

vasp_md_al.sh 文件内容

```
#!/bin/bash
#SBATCH --job-name="AL-AlScCu"
#SBATCH --account=research-3me-mse
```

```
#SBATCH --partition=compute
#SBATCH -t 72:00:00
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=48
#SBATCH --mem-per-cpu=3G

module load 2022r2
module load intel/oneapi-all
export OMP_NUM_THREADS=1

mddir=/scratch/yuchengji/lammps/al_vasp_md
vaspdir=$mddir/calculate_ab_initio_ef
cd $mddir
touch $mddir/log.timestamp

rm -f train.cfg curr.mtp preselect.cfg diff.cfg selected.cfg out.cfg

cp all.mtp curr.mtp
cp train_init.cfg train.cfg

#A. Active set construction
mlp calc-grade curr.mtp train.cfg train.cfg out.cfg --als-filename=state.als
rm out.cfg

while [ 1 -gt 0 ]
do
#B. MD simulations and extrapolative (preselected) configurations
touch preselect.cfg
srun --job-name="int_job" --partition=compute --time=00:30:00 --ntasks=1 --cpus-per-task=8 --mem-per-cpu=1GB --pty lmp_intel_cpu_intelmpi -in rdf.in
#mpirun -np 2 lmp_intel_cpu_intelmpi -in rdf.in

n_preselected=$(grep "BEGIN" preselect.cfg | wc -l)
if [ $n_preselected -gt 0 ]; then
#C. Selection
    mlp select-add curr.mtp train.cfg preselect.cfg diff.cfg --als-filename=state.als
    cp diff.cfg $vaspdir
    rm -f preselect.cfg
    rm -f selected.cfg

#D and E. Ab initio calculations and merging (updating the training set)
    cd $vaspdir
    mlp convert-cfg diff.cfg outmodel --output-format=vasp-poscar
    /home/yuchengji/bin/py39/bin/python3.9 al_vasp_calibration.py
    for file in outmodel*
    do
        cd $vaspdir/mlp_al/$file
        echo -e "VASP Calc   :     \c" >> $mddir/log.timestamp
        date >> $mddir/log.timestamp
        srun /home/yuchengji/bin/vasp/vasp_std_recompiled
        energy=$( grep "TOTEN" OUTCAR | tail -n 1 | awk -F" " '{print $5}')
        if [ $(echo "$energy < 10000 "| bc ) -eq 1 ];then
                echo -e "VASP Calc    : $mfile  $energy     \c" >> $mddir/log.timestamp
                date >> $mddir/log.timestamp
                mlp convert-cfg --input-format=vasp-outcar OUTCAR $mfile.cfg
                cp $mfile.cfg $vaspdir
        else
                echo -e "VASP Calc    : Error --> skipped     \c" >> $mddir/log.timestamp
                date >> $mddir/log.timestamp
        fi
        mlp convert-cfg --input-format=vasp-outcar OUTCAR $file.cfg


        cd $vaspdir
    done
    cat *.cfg >> $mddir/train.cfg
    rm -rf mlp_al diff.cfg outmodel* OUTCAR
    cd $mddir

#F. Training
    echo -e "MLIP Start   :     \c" >> $mddir/log.timestamp
    date >> $mddir/log.timestamp
    srun mlp train curr.mtp train.cfg --trained-pot-name=curr.mtp --update-mindist
    echo -e "MLIP END     :     \c" >> $mddir/log.timestamp
    date >> $mddir/log.timestamp

#A. Active set construction
    mlp calc-grade curr.mtp train.cfg diff.cfg out.cfg --als-filename=state.als
    rm -f diff.cfg
    rm -f out.cfg

elif  [ $n_preselected -eq 0 ]; then
    exit
fi
```

```
done
```

2. Compile the al_vasp_calibration.py file and transfer the cfg file to the POSCAR
   编辑 al_vasp_calibration.py 文件，将 cfg 格式转化为 POSCAR 格式
   vim al_vasp_calibration.py

   The content of al_vasp_calibration.py

   al_vasp_calibration.py 文件内容

```
import os
import sys
import shutil
sys.path.append("/scratch/yuchengji/script/python/")
from lib import VASP_Structure as VS

model_list = []
ele_model = VS.CfgRead('diff.cfg')

for files in os.listdir('.'):
    if 'outmodel' in files:
        model_list.append(files)
if len(model_list) > 10:
    model_num = 10
else:
    model_num = len(model_list)

for i in range(model_num):
    model = model_list[i]
    model_path = './mlp_al/%s' %(model)
    if not os.path.exists(model_path):
        os.makedirs(model_path)
    poscar = VS.Cfg2Poscar(model, ['Al', 'Sc', 'Cu', 'H'], ele_model[i])

    with open('%s/POSCAR' %(model_path), 'w') as psc_w:
        psc_w.write('\n'.join(poscar))
    try:
        shutil.copy('./raw_VASP_files/INCAR', '%s/INCAR' % (model_path))
        shutil.copy('./raw_VASP_files/KPOINTS', '%s/KPOINTS' % (model_path))
        shutil.copy('./raw_VASP_files/POTCAR', '%s/POTCAR' % (model_path))
        print("The %s model has been created" %(model))
    except Exception as err:
        print(err)
        print("The %s model creation failed!" %(model))
```

3. Submit the task to the cluster
   向超算提交任务

   ```
   sbatch vasp_md_al.sh
   ```

Good luck with your potential training!

**Yucheng Ji** PhD candidate
**MSE, 3mE, TU Delft**
E-mail: Y.Ji-2@tudelft.nl
**BAIC for MGE, USTB**
E-mail: jiyucheng@xs.ustb.edu.cn