Example usage of the ænet-LAMMPS interface for molecular dynamics simulations with artificial neural network (ANN) potentials.

Contact: Michael Stephen Chen (misch@stanford.edu), Tobias Morawietz (tobias.morawietz.nn@gmail.com), Nong Artrith (nartrith@atomistic.net)

If you make use of the aenet-LAMMPS interface, please cite the following reference:

M.S. Chen, T. Morawietz, H. Mori, T.E. Markland, N. Artrith, AENET-LAMMPS and AENET-TINKER: Interfaces for Accurate and Efficient Molecular Dynamics Simulations with Machine Learning Potentials, in preparation (2021).

The database can be obtained from the Materials Cloud repository, DOI: https://doi.org/10.24435/materialscloud:dx-ct

Tutorial: https://colab.research.google.com/drive/1Km8JVFM2DCeElAeE2n_WgMgLa7IU_IFh?usp=sharing or https://github.com/atomisticnet/aenet-lammps/tutorial

1. Downloading and extracting required codes

For this example, we will be using the ænet library version 2.0.4 and the LAMMPS release from 4 Feb 2020. Running the cell below downloads extracts both of these software packages into the working directory.

```
!wget https://github.com/atomisticnet/aenet/archive/refs/tags/v2.0.4.tar.gz
!wget https://download.lammps.org/tars/lammps-4Feb2020.tar.gz

!tar -xzvf lammps-4Feb2020.tar.gz
!tar -xzvf v2.0.4.tar.gz
```

```
Streaming output truncated to the last 5000 lines.
lammps-4Feb20/lib/gpu/lal_eam.cu
lammps-4Feb20/lib/gpu/lal_soft.cu
lammps-4Feb20/lib/gpu/lal_lj_expand.cpp
lammps-4Feb20/lib/gpu/lal_re_squared_ext.cpp
lammps-4Feb20/lib/gpu/lal_tersoff.cu
lammps-4Feb20/lib/gpu/lal_colloid.cpp
lammps-4Feb20/lib/gpu/lal_colloid.cu
lammps-4Feb20/lib/gpu/lal_tersoff_mod.cu
lammps-4Feb20/lib/gpu/lal_sw.cpp
lammps-4Feb20/lib/gpu/lal_dipole_lj_ext.cpp
lammps-4Feb20/lib/gpu/lal_buck.cpp
lammps-4Feb20/lib/gpu/README
lammps-4Feb20/lib/gpu/lal_lj_gromacs.cpp
lammps-4Feb20/lib/gpu/lal_born_coul_long_cs.cpp
lammps-4Feb20/lib/gpu/lal_coul_dsf.h
lammps-4Feb20/lib/gpu/lal_charmm_long.cpp
lammps-4Feb20/lib/gpu/lal_mie.cu
```

```
lammps-4Feb20/lib/gpu/lal_lj_class2_long.cpp
lammps-4Feb20/lib/gpu/lal_buck_coul_long.cu
lammps-4Feb20/lib/gpu/lal_neighbor_shared.cpp
lammps-4Feb20/lib/gpu/lal_lj_coul_long.cu
lammps-4Feb20/lib/gpu/Makefile.lammps.mac_ocl
lammps-4Feb20/lib/gpu/lal_re_squared_lj.cu
lammps-4Feb20/lib/gpu/lal_yukawa_ext.cpp
lammps-4Feb20/lib/gpu/lal_base_charge.h
lammps-4Feb20/lib/gpu/Makefile.xk7
lammps-4Feb20/lib/gpu/lal_lj_coul.cpp
lammps-4Feb20/lib/gpu/lal_gauss_ext.cpp
lammps-4Feb20/lib/gpu/lal_lj_cubic.h
lammps-4Feb20/lib/gpu/Makefile.lammps.mingw-cross
lammps-4Feb20/lib/gpu/lal_eam.cpp
lammps-4Feb20/lib/gpu/lal_coul_long_ext.cpp
lammps-4Feb20/lib/gpu/lal_lj_coul_ext.cpp
lammps-4Feb20/lib/gpu/lal_dipole_lj.cu
lammps-4Feb20/lib/gpu/lal_lj_coul_debye.cu
lammps-4Feb20/lib/gpu/lal_dpd_ext.cpp
lammps-4Feb20/lib/gpu/lal_ufm.cpp
lammps-4Feb20/lib/gpu/lal_device.cu
lammps-4Feb20/lib/gpu/lal_vashishta.cpp
lammps-4Feb20/lib/gpu/lal_lj_class2_long.h
lammps-4Feb20/lib/gpu/lal_neighbor_cpu.cu
lammps-4Feb20/lib/gpu/lal_buck_ext.cpp
lammps-4Feb20/lib/gpu/lal_dipole_lj.cpp
lammps-4Feb20/lib/gpu/lal_ellipsoid_extra.h
lammps-4Feb20/lib/gpu/lal_soft.h
lammps-4Feb20/lib/gpu/lal_lj.cu
lammps-4Feb20/lib/gpu/lal_base_dipole.h
lammps-4Feb20/lib/gpu/lal_table_ext.cpp
lammps-4Feb20/lib/gpu/lal_lj_coul_long_ext.cpp
lammps-4Feb20/lib/gpu/lal_zbl.cpp
lammps-4Feb20/lib/gpu/lal_tersoff_extra.h
lammps-4Feb20/lib/gpu/lal_coul.cu
lammps-4Feb20/lib/gpu/lal_tersoff.cpp
lammps-4Feb20/lib/gpu/lal_base_ellipsoid.cpp
lammps-4Feb20/lib/gpu/Makefile.mac_opencl
lammps-4Feb20/lib/gpu/lal_dipole_lj_sf.cpp
lammps-4Feb20/lib/gpu/Makefile.linux_multi
lammps-4Feb20/lib/gpu/lal_dpd.cpp
```

Now we download the ænet-LAMMPS code directly from the GitHub repository.

```
!git clone https://github.com/atomisticnet/aenet-lammps.git
```

```
Cloning into 'aenet-lammps'...
remote: Enumerating objects: 271, done.
remote: Counting objects: 100% (271/271), done.
remote: Compressing objects: 100% (138/138), done.
remote: Total 271 (delta 112), reused 199 (delta 71), pack-reused 0
Receiving objects: 100% (271/271), 10.25 MiB | 10.37 MiB/s, done.
Resolving deltas: 100% (112/112), done.
```

2. Compiling ænet

In the cell below, we build the ænet library that will be called by the ænet-LAMMPS interface. Note that in this example we use the OpenBLAS library, but other Makefiles linking alternative optimized linear algebra libraries are provided as part of ænet. For more details on compiling ænet, please see the install instructions found as part of the ænet documentation. Also, parallelization will be left to LAMMPS so we should only compile a serial version of the ænet library.

```
%cd /content/aenet-2.0.4/lib/
!make
%cd ../src
!make clean
!make -f makefiles/Makefile.gfortran_openblas_serial lib
%cd /content/
```

```
/content/aenet-2.0.4/lib
tar xfvz Lbfgsb.3.0.tar.gz
./._Lbfgsb.3.0
Lbfgsb.3.0/
Lbfgsb.3.0/._algorithm.pdf
Lbfgsb.3.0/algorithm.pdf
Lbfgsb.3.0/blas.f
Lbfgsb.3.0/._code.pdf
Lbfgsb.3.0/code.pdf
Lbfgsb.3.0/driver1.f
Lbfgsb.3.0/driver1.f90
Lbfgsb.3.0/driver2.f
Lbfgsb.3.0/driver2.f90
Lbfgsb.3.0/driver3.f
Lbfgsb.3.0/driver3.f90
Lbfgsb.3.0/._iterate.dat
Lbfgsb.3.0/iterate.dat
Lbfgsb.3.0/lbfgsb.f
Lbfgsb.3.0/License.txt
Lbfgsb.3.0/linpack.f
Lbfgsb.3.0/._Makefile
Lbfgsb.3.0/Makefile
Lbfgsb.3.0/._OUTPUTS
Lbfgsb.3.0/OUTPUTS/
Lbfgsb.3.0/README
Lbfgsb.3.0/timer.f
Lbfgsb.3.0/x.lbfgsb_77_1
Lbfgsb.3.0/x.lbfgsb_77_2
Lbfgsb.3.0/x.lbfgsb_77_3
Lbfgsb.3.0/x.lbfgsb_90_1
Lbfgsb.3.0/x.lbfgsb_90_2
Lbfgsb.3.0/x.lbfgsb_90_3
Lbfgsb.3.0/OUTPUTS/._output_77_1
Lbfgsb.3.0/OUTPUTS/output_77_1
Lbfgsb.3.0/OUTPUTS/._output_77_2
Lbfgsb.3.0/OUTPUTS/output_77_2
```

```
Lbfgsb.3.0/OUTPUTS/._output_77_3
Lbfgsb.3.0/OUTPUTS/output_77_3
Lbfgsb.3.0/OUTPUTS/._output_90_1
Lbfgsb.3.0/OUTPUTS/output_90_1
Lbfgsb.3.0/OUTPUTS/._output_90_2
Lbfgsb.3.0/OUTPUTS/output_90_2
Lbfgsb.3.0/OUTPUTS/._output_90_3
Lbfgsb.3.0/OUTPUTS/output_90_3
gfortran -c -O2 Lbfgsb.3.0/blas.f -o Lbfgsb.3.0/blas.o
gfortran -c -O2 Lbfgsb.3.0/lbfgsb.f -o Lbfgsb.3.0/lbfgsb.o
gfortran -c -O2 Lbfgsb.3.0/linpack.f -o Lbfgsb.3.0/linpack.o
gfortran -c -O2 Lbfgsb.3.0/timer.f -o Lbfgsb.3.0/timer.o
ar -crusv liblbfgsb.a Lbfgsb.3.0/blas.o    Lbfgsb.3.0/lbfgsb.o Lbfgsb.3.0/linpac
ar: `u' modifier ignored since `D' is the default (see `U')
a - Lbfgsb.3.0/blas.o
a - Lbfgsb.3.0/lbfgsb.o
a - Lbfgsb.3.0/linpack.o
a - Lbfgsb.3.0/timer.o
gfortran -c -O2 -fPIC -o Lbfgsb.3.0/blas_pic.o Lbfgsb.3.0/blas.f
gfortran -c -O2 -fPIC -o Lbfgsb.3.0/lbfgsb_pic.o Lbfgsb.3.0/lbfgsb.f
gfortran -c -O2 -fPIC -o Lbfgsb.3.0/linpack_pic.o Lbfgsb.3.0/linpack.f
gfortran -c -O2 -fPIC -o Lbfgsb.3.0/timer_pic.o Lbfgsb.3.0/timer.f
gcc -shared Lbfgsb.3.0/blas pic.o    Lbfgsb.3.0/lbfgsb pic.o Lbfgsb.3.0/linpack
```

3. Compiling LAMMPS with ænet support

Note, for more detailed instructions on how to patch and compile LAMMPS with ænet support please see the ænet-LAMMPS GitHub repository.

First, let us organize things so that the ænet header and library files are where we expect them to be when compiling things.

```
%cd /content/lammps-4Feb20/lib/
!mkdir -p aenet/lib
!mkdir -p aenet/include
%cd /content/lammps-4Feb20/lib/aenet/include
!ln -s /content/aenet-2.0.4/src/aenet.h
%cd /content/lammps-4Feb20/lib/aenet/lib
!ln -s /content/aenet-2.0.4/src/libaenet.* .
!ln -s /content/aenet-2.0.4/lib/liblbfgsb.* .
```

```
/content/lammps-4Feb20/lib
/content/lammps-4Feb20/lib/aenet/include
/content/lammps-4Feb20/lib/aenet/lib
```

Next, let us patch LAMMPS with the ænet-LAMMPS interface files. This entails copying over the provided USER-AENET folder into the LAMMPS src directory and replacing the LAMMPS Makefile. Since we compiled the ænet library using Makefile.gfortran_openblas_serial, we will use the

corresponding Install.sh file for our custom USER-AENET package so as to make sure that

```
%cd /content/lammps-4Feb20/src/
! cp -r /content/aenet-lammps/USER-AENET .
! cp -f /content/aenet-lammps/Makefile-aenetlammps Makefile
! cp -f USER-AENET/Install.sh-gfortran_openblas_serial USER-AENET/Install.sh
```

```
/content/lammps-4Feb20/src
```

We need to flag the USER-AENET package to be included as part of our LAMMPS compilation and lastly we compile an MPI-parallelized LAMMPS executable.

```
%cd /content/lammps-4Feb20/src/
! make yes-user-aenet
! make mpi
```

```
/content/lammps-4Feb20/src
Gathering installed package information (may take a little while)
make[1]: Entering directory '/content/lammps-4Feb20/src'
Gathering git version information
make[1]: Leaving directory '/content/lammps-4Feb20/src'
Compiling LAMMPS for machine mpi
make[1]: Entering directory '/content/lammps-4Feb20/src/Obj_mpi'
cc -O -o fastdep.exe ../DEPEND/fastdep.c
make[1]: Leaving directory '/content/lammps-4Feb20/src/Obj_mpi'
make[1]: Entering directory '/content/lammps-4Feb20/src/Obj_mpi'
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
```

```
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
mpicxx -g -O3   -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64   -I../../lib/aenet/include
```

4. Running LAMMPS with ænet ANN potentials

An example LAMMPS molecular dynamics simulation of liquid water with an ænet ANN potential is provided as part of the ænet-LAMMPS package.

First we need to make sure that the ænet library file can be dynamically linked when we run LAMMPS.

```
%env LD_LIBRARY_PATH=/content/lammps-4Feb20/lib/aenet/lib:/usr/local/nvidia/lib:/usr/
!echo $LD_LIBRARY_PATH
```

```
env: LD_LIBRARY_PATH=/content/lammps-4Feb20/lib/aenet/lib:/usr/local/nvidia/lib:
/content/lammps-4Feb20/lib/aenet/lib:/usr/local/nvidia/lib:/usr/local/nvidia/lib
```

In the cell below we run the short example simulation. Some details concerning the simulation (see the LAMMPS input file md.lmp for more details):

- NVT simulation of liquid water at 300K
- Initial frame specified by 01_Start/firstframe.start file (64 water molecules in box with side lengths of 12.417 Angstroms)
- H.ann and O.ann are the ænet ANN potential files
- Simulation runs for 100 steps with 0.5 fs spacing between steps and writing out properties every 2.0fs
- Input file flags the use of the "metal" unit system since the ANN potential was trained using eV units for energies and Angstroms units for positions.

- Positions, velocities, forces, and energies are written out to files in the 02_Traj folder (traj.xyz, velocities.xyz, forces.xyz, and analysis_frames.dat respectively)

```
%cd /content/aenet-lammps/examples/water
!/content/lammps-4Feb20/src/lmp_mpi -in md.lmp

     /content/aenet-lammps/examples/water
     LAMMPS (4 Feb 2020)
     Reading data file ...
       orthogonal box = (0 0 0) to (12.4171 12.4171 12.4171)
       1 by 1 by 1 MPI processor grid
       reading atoms ...
       192 atoms
       read_data CPU = 0.000943816 secs
     Neighbor list info ...
       update every 1 steps, delay 10 steps, check yes
       max neighbors/atom: 2000, page size: 100000
       master list distance cutoff = 8.35
       ghost atom cutoff = 8.35
       binsize = 4.175, bins = 3 3 3
       1 neighbor lists, perpetual/occasional/extra = 1 0 0
       (1) pair aenet, perpetual
           attributes: full, newton on
           pair build: full/bin/atomonly
           stencil: full/bin/3d
           bin: standard
     Setting up Verlet run ...
       Unit style    : metal
       Current step  : 0
       Time step     : 0.0005
     Per MPI rank memory allocation (min/avg/max) = 6.705 | 6.705 | 6.705 Mbytes
     Step Temp E_pair E_mol TotEng Press
            0          300   -30056.194            0   -30048.787   -4138.7878
            4    311.16524   -30056.471            0   -30048.789   -2680.0645
            8    325.61646   -30056.821            0   -30048.782    4949.5629
           12    310.57226   -30056.474            0   -30048.806    8251.7059
           16    315.26274   -30056.585            0   -30048.801    2446.1663
           20    306.02986    -30056.37            0   -30048.815   -2670.3051
           24    329.91611   -30056.977            0   -30048.832   -740.68931
           28    330.23232   -30057.005            0   -30048.852    5347.1609
           32    319.03827   -30056.716            0    -30048.84    6956.5274
           36    316.10423   -30056.657            0   -30048.853    3009.7582
           40    299.62656   -30056.248            0    -30048.85    571.70686
           44     326.0215   -30056.911            0   -30048.862    545.76778
           48    312.05352   -30056.516            0   -30048.812    3383.9577
           52    318.83778   -30056.654            0   -30048.782    4847.6457
           56    300.95386   -30056.155            0   -30048.725    6217.3848
           60    302.54665   -30056.278            0   -30048.808    4371.6857
           64      304.476   -30056.331            0   -30048.814    2679.8174
           68    300.04551   -30056.189            0   -30048.781    3053.4613
           72    314.64045   -30056.534            0   -30048.766     5653.102
           76    293.68455   -30055.993            0   -30048.742    8959.0566
           80    324.65481   -30056.783            0   -30048.768     5872.407
           84    305.23793   -30056.259            0   -30048.723    3076.0805
```

```
      88    324.61769   -30056.773            0   -30048.759    1100.5434
      92    311.16735   -30056.459            0   -30048.777    4735.5056
      96    308.70087   -30056.388            0   -30048.767    8820.4214
     100    320.68558   -30056.668            0   -30048.751    7659.6949
Loop time of 805.919 on 1 procs for 100 steps with 192 atoms

Performance: 0.005 ns/day, 4477.326 hours/ns, 0.124 timesteps/s
95.7% CPU use with 1 MPI tasks x no OpenMP threads

MPI task timing breakdown:
Section |  min time  |  avg time  |  max time  |%varavg| %total
```