Introduction to CSLD

CSLD, or compressive sensing lattice dynamics, is a comprehensive package to study lattice dynamics from first-principles. The interatomic force constants were fitted using the compressive sensing algorithm. Some of the attractive features of CSLD are

- Accurate: up to 6th order anharmonic terms. Optimized treatment of long-range Coulomb interactions in polar semiconductors.
- Robust: fitting regularized with L1 penalty
- Efficient: using as few training structures as possible
- Versatility: phonon (harmonic) and anharmonic force constants, vibrational free energies, interatomic potentials for solid molecular dynamics simulations.

CSLD is released as an open source package under the MIT license. For issues, please report via github, or contact Fei Zhou mailto:fei.fzhou@gmail.com

Installation

Hereby we refer to all files from the main source code directory. User input commands start with "$", followed by output without leading "$":

```
$ ls setup.py
setup.py
```

- Prerequisite
  - Python 3
  - Python packages: numpy, scipy, matplotlib, ConfigParser
  - Library spglib
  - f2py3 associated with Python3. If it is available in another name, e.g. f2py, make it available by e.g. alias f2py3=f2py
  - C++ and Fortran 90 compilers

  If the prerequisites are not met, please either ask your sys admin to install them, or install your own python 3 environment, e.g. miniconda or anaconda. The codes were tested on conda installations on Linux and MacOS.
- If you have no plan to modify/contribute to the source codes, go to the code directory and install for all users

```
$ python3 setup.py install
```

- Install for yourself without admin rights

```
$ python3 setup.py install --user
```

- If you plan to develop (i.e. modify) the codes

```
$ python3 setup.py develop
```

- Develop without admin rights

```
$ python3 setup.py develop --user
```

- If compilation fails, you may need to modify the Makefile manually.

How to cite

If you use CSLD in your research, please cite the following works:

- Fei Zhou, Weston Nielson, Yi Xia, and Vidvuds Ozolins, Phys. Rev. Lett. 113, 185501, (2014). http://dx.doi.org/10.1103/PhysRevLett.113.185501
- Fei Zhou et al, arxiv:1805.08903, arxiv:1805.08904.

Tutorial

More detailed documentation can be found in the Manual section next. This section provides a short tutorial To get started quickly after installation. The files can be found in the examples/ directory.

Si

Getting started

All input files have been prepared in the examples/Si/ folder.

- Phonon dispersion

```
$ cp -r examples/Si examples/Si-test; cd examples/Si-test
$ csld_main -f csld.in # or just csld_main
```

Then Check out the generated plots.pdf

- Third and fourth order anharmonicity with modified config file

```
$ csld_main -f csld.in-anharmonic
```

The above can be equivalently achieved by modifying the command line rather than the config file

```
$ csld_main --override "[fitting] solution_known=sol_2nd" \
--override "[fitting] submodel1=anh 3 4" \
--override "[fitting] solution_out=solution_all" \
--override "[training] traindat1=fcc333/SPOSCAR fcc333/dir*0.06"
```

Input files preparation

Now let's see how to start from scratch, by copying over the config file control.in and symmetrizing a primitive cell Si/POSCAR (pretending that Si/POSCAR was relaxed without proper symmetry):

```
$ cd ..; mkdir Si-test2; cp Si/csld.in Si-test2; cd Si-test2
$ polaron_main --task primitive --prim ../Si/POSCAR --tol 0.001 >POSCAR
```

Next, prepare training data:

```
$ mkdir fcc333; cd fcc333; echo -e "-3 3 3\n 3 -3 3\n 3 3 -3" > sc.txt
$ polaron_main --task supercell --p1 sc.txt --prim ../POSCAR > SPOSCAR
```

where sc.txt is a 3x3 integer scaling matrix specifying the supercell. Perturbed supercells for phonon calculations can be generated in subdirectories by:

```
$ polaron_main --task rand_disp_dir -R 0.01 --p1 SPOSCAR -N 1
$ ls *
SPOSCAR  sc.txt

dir_00000-disp0.01:
POSCAR
```

where the supercell fcc333/SPOSCAR was perturbed by R=0.01 angstrom randomly for N=1 time. For anharmonic terms, larger displacements should be attempted, e.g.:

```
$ polaron_main --task rand_disp_dir -R 0.06 --p1 SPOSCAR -N 1
```

The user should then run DFT codes (e.g. VASP) to compute the corresponding total forces. Once done, the latter can be collected using the supplied **get-force.sh** script:

```
$ get-force.sh -d dir_00*/
```

NaCl

This is an example that requires long-range forces

Getting started

- Phonon calculation without long range forces. Note the absence of LO-TO splitting in the obtained plots.pdf

```
$ cp -r examples/NaCl examples/NaCl-test; cd examples/NaCl-test
$ csld_main --override '[phonon]nac=-1'
```

- With long range forces

```
$ csld_main
```

Never mind the artifact in the phonon dispersion curve at zone center. It's the non-analytic correction.

- Third and fourth order anharmonicity trained on fcc333/dir_00000-disp0.06

```
$ csld_main -f csld.in-anharmonic
```

- Export second and third-order force constants files (FORCE_CONSTANTS_2ND and FORCE_CONSTANTS_3RD) for thermal conductivity calculations in ShengBTE

```
$ csld_main --override '[export_potential] export_shengbte=5 5 5 2 3' \
--save_pot_step 1 --phonon -f csld.in-anharmonic
```

Input files for long-range forces

To obtain the Born effective charges and dielectric tensor required for long-range treatment, dielectric calculations should be performed with density functional perturbation theory (DFPT) for the primitive cell before csld fitting. Obtain born_charge.txt and epsilon_inf.txt by

```
$ polaron_main --task born --p1 PATH_TO_DFPT_CALCULATION/OUTCAR
```

Manual

Executable scripts

The main executable is the Python 3 script scripts/**csld_main**. Other executables include the scripts/**polaron_main** script for various auxilliary functions. Help is available via:

```
$ csld_main -h
$ polaron_main -h
```

Input files

The csld_main code takes two basic input files: **POSCAR** and **csld.in**.

- The structure of the primitive cell in VASP 5 format. The file name is specified in the config file (POSCAR, see above). It's important to keep high precision in the structure file. We recommend using the **polaron_main** helper script to symmetrize your primitive cell

```
$ polaron_main --task primitive --prim your_input_POSCAR --tol 0.001 >POSCAR
```

- csld.in, the main configuration file containing essentially all the settings.
    - May be specified with the -f switch (see examples above)
    - The format is the usual configuration file with section headers and variable names under each section. Comment starts with "#". For example, the following specifies "POSCAR" as the primitive cell structure and 0.001 as the tolerance of symmetry finding (using spglib):

      ```
      [structure]
      # primitive cell
      prim = POSCAR
      # symmetry finding tolerance, NOT used yet!
      sym_tol = 1E-3
      ```

    - The tag (e.g. "prim") is case insensitive. Our examples come with all lower case tags.
    - We recommend copying over one of the config files in the test/ example and adapting to your needs.
    - An alternative to editing csld.in is to change settings at the command line via the "--override [HEADER] TAG=VAL" option, where HEADER is a section in csld.in, and TAG and VAL are the desired entry under [HEADER]. This is equivalent to editing the corresponding entry in csld.in. For example, to change the primitive cell to PRIM

```
$ csld_main --override "[structure] prim = PRIM"
```

- Additionally, for polar semiconductors, the Born effective charege and static dielectric tensors are required to describe the long-range interactions. Perform DFPT calculations and extract the required tensors into files born_charge.txt and epsilon_inf.txt by

```
$ polaron_main --task born --p1 PATH_TO_DFPT_CALCULATION/OUTCAR
```

Training data

Additional input files include the supercell structure(s), perturbed structures and associated total forces.

Once the supercell size is determined, the main variable in the above setup is the number of structures (-N switch) to generate.

- For phonon calculations, determine the pair-interaction cutoff distance (say 8 angstrom) and estimate the number perturbations on supercell fcc333

```
$ polaron_main --task nsc --prim POSCAR -R 8 --p1 fcc333/sc.txt
```

- For anharmonic calculations, start with a few supercell calculations with larger displacement, e.g. with the same number as phonon calculations, and incrementally add more calculations as needed. Note that the most significant anharmonic terms are typically short-ranged, so one might try smaller supercells than phonon calculations.

Program flow and settings

The csld_main script runs in the following steps, each controlled by a command-line argument of csld_main with detailed settings under a section of the csld.in config file. The most important settings are given below, too.

- Model setup, i.e. generating clusters, as well as identifying independent model parameters (implemented as computation of a symmetrization matrix *C*)
    - command line switch "--clus_step STEP". STEP=2 means generating clusters, **3** (default) =generate & save to file clusters.out.
    - command line switch "--symC_step STEP". 1=load file, 2=compute, **3** =compute & save to file Cmat.mtx
    - users usually should focus on the following settings rather than changing the command line switches unless they know what they are doing!

"[model]" section of csld.in.

tag	value	description
max_order	int	max order of clusters, e.g. 2 for harmonic model, 4 for up to fourth-order
cluster_diameter	real [real...]	cluster cutoff for pair, triplet, ... e.g. 8.0 4.5
fractional_distance	True|**False**	whether distance is scaled relative to lattice constant (2nd line of POSCAR)

- Training init step, i.e. computation of the correlation matrix
    - switch "--train_step STEP". 1=load file, 2=compute, **3** =compute & save, 4=skip.

"[training]" section

tag	value	description
corr_type	str	Which type of property to fit: f for force, e for energy
cluster_diameter	real [real...]	cluster cutoff for pair, triplet, ... e.g. 8.0 4.5

traindat*	str_SC str_dir1 [str_dir2...]	POSCAR of supercell, followed by list of subdirs, e.g. fcc333/POS fcc333/dir-*. Multiple traindat entries with different supercells may be supplied to mix and match supercells of different sizes, e.g. traindat1= fcc222.... \n traindat2= fcc333....

- Fitting step to obtain model parameters using compressive sensing or other algorithms.
  - switch "--fit_step STEP". 1=file, 2=generate, **3** =generate & save.

"[fitting]" section

tag	value	description
method	int	1=FPC, 3=split Bregman 5=split Bregman + right preconditioning, 201=ridge regression
nsubset	int	number of subset fittings
uscale_list	real [real...]	displacement scale. Approximately 0.01 for phonon calculations, larger values for anharmonic fitting
mulist	real [real...]	list of mu (weight of L1 or L2 norm in penalty) to loop over, e.g. 1e-1 1e-3 1e-4 1e-5 1e-6
submodel*	str int [int]	name of the fitting, and list of orders included, e.g. harmonic 1 2 (fitting the 1st and 2nd-order FCs only such that FCs of other other will be set to zero). Multiple entries may be entered to test different fittings in one run, e.g. submodel0=harmonic 1 2 \n submodel1=up-to-third 1 2 3. If no submodel is supplied, default to fitting all orders up to max_order.
solution_out	str	filename for the obtained vector of solution (independent parameters)
solution_in	str	filename for loading previous solution instead of fitting
solution_known	str	filename for previously obtained parameter phi_in. The force predicted by phi_in will be subtracted from the total force. This is useful in conjunction with submodel to fit in several steps, e.g. assuming max_order=4, first fit harmonic terms with submodel=harmonic 1 2; solution_out=sol_2nd, then fit anharmonic terms with solution_known=sol_2nd; submodel1=anh 3 4; solution_out=solution_all

- Phonon step
  - switch "--phonon_step STEP". 0=skip, **1** = compute.

"[phonon]" section

tag	value	description
nac	int	Method for non-analytic correction. **-1** =disabled, 0=long range treatment in arxiv:1805.08904
wavevector	str	If specified, plot phonon dispersion. wavevector = Auto will turn on automatic generation of special paths in reciprocal space. Manual settings e.g. [[25, [0,0,0],'Gamma', [0,0.5,0.5], 'X']] will add 25 points between zone center and X point
unit	str	Unit for dispersion and DOS. One of THz, meV, eV, cm
dos_grid	int x 3	If specified, plot density states sampled on a grid, e.g. 10 10 10

ismear	int	smearing method of DOS integration. 0=Gaussian, 1=Lorentzian, **-1** =tetrahedron method
epsilon	real	Smearing width for Gaussian/Lorentzian
pdos	True|False	Whether to plot partial DOS
thermal_t_range	real x 3	if specified together with dos_grid, calculate quasi-harmonic thermodynamic properties in the temperature range: begin end increment in Kelvin, e.g. 50 800 50.
thermal_out	str	filename for thermodynamic properties

- Exporting force constants step.
  - controlled by "--save_pot_step STEP". **0** = skip, 1=save.

"[export_potential]" section

tag	value	description
export_shengbte	int x 3 int [int...]	If specified, export force constants for calculation in ShengBTE. First 3 integers designate size of supercell for pair force constants, followed by list of orders to export. e.g. exporting Hessian matrix of a 5x5x5 supercell, as well as 3rd-order FCs with 5 5 5 2 3

- Prediction of supercells forces or energies with known solution
  - switch "--pred_step STEP". **0** =skip, 1=load training setup correlation matrix from file, 2=compute, 3=compute & save. Use "csld_main --predict" to quickly predict supercells"
  - Settings in "[prediction]" with the same sets of tags as "[training]". Usually used together with "[fitting] solution_in = previous_solution"

Output files

- The list of symmetrized force constants: solution.out
- For phonon calculations, the phonon plots (plots.pdf), band dispersion (phonon-dispersion.out) and density of states (phonon-total-dos.out, phonon-partial-dos.out) data files, as well as thermodynamic properties (free energy, vibrational entropy, etc, in thermal.out) in the quasi-harmonic approximation.
- The second and third order FCs can be exported to perform phonon scattering rates and thermal conductivity calculations in ShengBTE. Assuming a 5 x 5 x 5 supercell for the harmonic FCs

```
$ csld_main --override '[export_potential] export_shengbte=5 5 5 2 3' \
--save_pot_step 1 --phonon
```

- The fourth order FCs can be exported to a modified version of ShengBTE that consider frquency shift and scattering due to four-phonon processe

```
$ csld_main --override '[export_potential] export_shengbte=5 5 5 2 3 4' \
--save_pot_step 1 --phonon
```