

Editor's choice paper

aflow++: A C++ framework for autonomous materials design

Corey Oses^{a,b}, Marco Esters^{a,b}, David Hicks^{c,a,b}, Simon Divilov^{a,b}, Hagen Eckert^{a,b}, Rico Friedrich^{d,e,b}, Michael J. Mehl^{a,b}, Andriy Smolyanyuk^{f,b}, Xiomara Campilongo^b, Axel van de Walle^g, Jan Schroers^h, A. Gilad Kusne^{i,j}, Ichiro Takeuchi^j, Eva Zurek^k, Marco Buongiorno Nardelli^{l,m,b}, Marco Fornari^{n,b}, Yoav Lederer^{o,b}, Ohad Levy^{a,b,o}, Cormac Toher^{p,b}, Stefano Curtarolo^{a,b,*}

^a Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA

^b Center for Autonomous Materials Design, Duke University, Durham, NC 27708, USA

^c LIFT, American Lightweight Materials Manufacturing Innovation Institute, Detroit, MI 48216, USA

^d Institute of Ion Beam Physics and Materials Research, Helmholtz-Zentrum Dresden-Rossendorf, 01328 Dresden, Germany

^e Theoretical Chemistry, Technische Universität Dresden, 01062 Dresden, Germany

^f Institute of Solid State Physics, Technische Universität Wien, A-1040 Wien, Austria

^g School of Engineering, Brown University, Providence, RI 02912, USA

^h Department of Mechanical Engineering and Materials Science, Yale University, New Haven, CT 06511, USA

ⁱ Materials Measurement Science Division, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA

^j Department of Materials Science and Engineering, University of Maryland, College Park, MD 20742, USA

^k Department of Chemistry, State University of New York at Buffalo, Buffalo, NY 14260, USA

^l Department of Physics and Department of Chemistry, University of North Texas, Denton, TX 76203, USA

^m Santa Fe Institute, Santa Fe, NM 87501, USA

ⁿ Department of Physics and Science of Advanced Materials Program, Central Michigan University, Mount Pleasant, MI 48859, USA

^o Department of Physics, NRCN, Beer-Sheva, 84190, Israel

^p Department of Materials Science and Engineering and Department of Chemistry and Biochemistry, University of Texas at Dallas, Richardson, TX 75080, USA

ARTICLE INFO

Dataset link: <https://aflow.org>

Keywords:

AFLOW

Autonomous computation

Machine learning

Workflows

ABSTRACT

The realization of novel technological opportunities given by computational and autonomous materials design requires efficient and effective frameworks. For more than two decades, aflow++ (Automatic-Flow Framework for Materials Discovery) has provided an interconnected collection of algorithms and workflows to address this challenge. This article contains an overview of the software and some of its most heavily-used functionalities, including algorithmic details, standards, and examples. Key thrusts are highlighted: the calculation of structural, electronic, thermodynamic, and thermomechanical properties in addition to the modeling of complex materials, such as high-entropy ceramics and bulk metallic glasses. The aflow++ software prioritizes interoperability, minimizing the number of independent parameters and tolerances. It ensures consistency of results across property sets — facilitating machine learning studies. The software also features various validation schemes, offering *real-time* quality assurance for data generated in a high-throughput fashion. Altogether, these considerations contribute to the development of large and reliable materials databases that can ultimately deliver future materials systems.

* Corresponding author at: Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA.
E-mail address: stefano@duke.edu (S. Curtarolo).

1. Introduction

The Automatic-Flow (aflow++) Framework for Materials Discovery is an interconnected collection of algorithms and workflows that have been developed to address the challenge of accelerated materials' calculation and identifications. We clarify the difference between aflow++ and aflow.org. The subject of this article, aflow++, is a set of codes which enables data generation, materials discovery, analysis, identification and optimization. On the other hand, aflow.org is a web ecosystem of FAIR databases, software and tools, including online machinery to analyze and download data, as well as different educational resources [1–3]. Many of the functionalities of aflow.org rely on performing on-the-fly aflow++ operations, facilitating a more effective, reliable, and reusable development.

The framework aflow++ comprises a comprehensive suite of functionalities that through the years has been successfully applied to the discovery of many new systems, e.g., permanent magnets [4,5], superalloys [6,7], high-entropy high-hardness plasmonic carbides [8–11], and phase-change memory compositions [12]. The framework is written in C++ (~650,000 lines as of version 3.2.12) with a growing Python environment, and operates on UNIX architectures (GNU-Linux, macOS). It automates the input file generation, job submission and queue management, error correction, analysis, storage, and dissemination of the results [1]. Workflows are fully parallelizable, having break-points where independent components can be run simultaneously on a cluster accelerated with multi-threaded functionalities. Integral to aflow++'s automation is the standardization of input parameter sets [13] and structure prototypes [14]. The software offers direct and programmatic access to a broad range of experimentally-observed structures [15–17] with adjustable internal degrees of freedom, enabling the construction of virtually any conceivable periodic structure. aflow++ is fully integrated to work with the Vienna Ab-initio Simulation Package (VASP) [18] and provides some support for QUANTUM ESPRESSO [19], ABINIT [20], the Fritz-Haber-Institut Ab Initio Materials Simulation Package (FHI-AIMS) [21], the ELK Code [22], and the Alloy Theoretic Automated Toolkit (ATAT) [23].

Historically, aflow++ started from the characterization of inorganic intermetallic crystals, e.g., early adoption in machine-learning/data-mining [43,44] and high-throughput [45], in line with the use of a plane-wave basis and the generalized gradient approximation (GGA) pseudopotentials of Perdew, Burke, and Ernzerhof (PBE) [46] by default. Recently, this scope has been extended to include i. ceramics, largely enabled by the coordination corrected enthalpies (aflow-CCE) method [33,35] and ii. structurally and chemically disordered systems, facilitated by the creation of the thermodynamic density of states (DOS) descriptor [3,36] and the Glass-Forming-Ability (aflow-GFA) [36,37], Partial Occupation (aflow-POCC) [38], and Quasi-Chemical Approximation (aflow-QCA) [39] modules. Descriptors are

feasibly-calculated quantities based on microscopic features that offer predictive power of macroscopic properties of the material [47]. Their development and application remain at the heart of aflow++, particularly for the prediction of thermodynamic stability/synthesizability [32], electronic [26,27,48], and thermomechanical [28–30,34,40,41,49] properties.

In this article, we highlight functionality and workflows that have been developed since the original aflow++ report [50] and demonstrate their interoperability within the overall environment. The various modules and tools included in the standard distribution (version 3.2.12, Fall 2022) are listed in Table 1. Examples of their application for the discovery of new materials are presented, providing a practical guide for future materials informatics investigations. Through ongoing innovation and implementation of robust descriptors and workflows, aflow++ continues to deliver valuable solutions [51] as well as playing a role in accelerating the pace of automation in the materials community.

2. Basic usage

Installation instructions and a description of the aflow.rc file are included in the [Appendix](#).

Generating geometry files. Geometry files for crystalline materials can be automatically generated with aflow++ by decorating prototype structures with different elements. The command-line syntax to generate prototypes is:

```
aflow --proto=label.ordering:elements --params=parameters
```

where *label* is the prototype designation (or alias) and *parameters* are the comma-separated degrees of freedom for the prototype. *elements* is a colon-separated list of elements in alphabetical order decorating the structure (e.g., Ag:C:Cu). By default, the structure is decorated with fictitious atoms (i.e., A, B, C, D, ...). *ordering* specifies the site decoration (e.g., ABC vs. BAC), where the placement of the letter corresponds to the site on the prototype and the letter corresponds to the alphabetically-ordered *elements*.

There are two prototype libraries in aflow++: the High-Throughput Quantum Computing (HTQC) library [50] and the aflow++ Prototype Encyclopedia [15–17]. HTQC prototypes are hard-coded structures that do not require any degrees of freedom to be specified, and whose labels are largely ad hoc e.g., 201 and T0001 denote the rocksalt and Heusler structures, respectively. An example command to generate an HTQC structure is:

```
aflow --proto=201:C1:Na
```

For Prototype Encyclopedia prototypes [15–17], the label — or aflow++ prototype label — is an underscore-delimited string describing the symmetry of a crystal structure. For example, corundum has the prototype label A2B3_hr10_167_c_e-001, where:

Table 1

Current aflow++ tools in the standard distribution (version 3.2.12, Fall 2022). Non-C++ modules include AFLOW π : medium-throughput framework for QUANTUM ESPRESSO and the ACBN0 function [24,25], and PAOFLOW: procedure for projecting the full plane-wave solution on a reduced space of pseudoatomic orbitals [26,27], as described in Section 5.

Acronym	Section	Module or Library	Refs.
aflow-AAPL	7.4	<u>Automatic Anharmonic Phonon Library</u>	[28]
aflow-AEL	7.1	<u>Automatic Elasticity Library</u>	[29]
aflow-AGL	7.1	<u>Automatic GIBBS Library</u>	[30]
aflow-APE	9	<u>AFLOW Python Environment</u>	[14,31–33]
aflow-APL	7.2	<u>Automatic Phonon Library</u>	[34]
aflow-CCE	6.2	<u>Coordination Corrected Enthalpies Module</u>	[33,35]
aflow-CHULL	6.1	<u>Convex Hull Module</u>	[32]
aflow-GFA	8.3	<u>Glass-Forming-Ability Module</u>	[36,37]
aflow-POCC	8.1	<u>Partial Occupation Module</u>	[38]
aflow-QCA	8.2	<u>Quasi-Chemical Approximation Module</u>	[39]
aflow-QHA	7.3	<u>Quasi-Harmonic Approximation Library</u>	[40–42]
aflow-SYM	4.2	<u>Symmetry Module</u>	[31]
aflow-XtalFinder	4.3	<u>Crystal Finder Module</u>	[14]

- the first field indicates the reduced stoichiometry (A_2B_3),
- the second field indicates the Pearson symbol (hR10),
- the third field indicates the space group number (167),
- the fourth field indicates the Wyckoff letters associated with species A (c),
- the fifth field indicates the Wyckoff letters associated with species B (e), and
- the sixth field indicates the alias for the corundum parameter-set extracted and generalized by aflow-XtalFinder (001).

In addition to the prototype label, aflow-XtalFinder determines the lattice and Wyckoff parameters that are not fixed by symmetry and returns their values for the particular geometry. For corundum, these degrees of freedom are lattice parameters a and c/a and Wyckoff coordinates x_1 and x_2 in direct/fractional space: 4.7607, 2.7296, 0.3522, and 0.5561, respectively, for Al_2O_3 . This structure can be generated with the parameters provided explicitly:

```
aflow --proto=A2B3_hR10_167_c_e:Al:0 --params
  ↪ =4.7607,2.7296,0.3522,0.5561
```

or using the 001 alias:

```
aflow --proto=A2B3_hR10_167_c_e-001:Al:0
```

Options specific to the Prototype Encyclopedia prototype include:

- `--add_equations`: The symbolic version of the geometry file (in terms of the variable degrees of freedom) is printed after the numeric geometry file.
- `--equations_only`: Only prints the symbolic version of the geometry file (in terms of the variable degrees of freedom).

aflow++ can read and generate geometry files (*GEOM_FILE*) in different file formats: VASP [52], QUANTUM ESPRESSO [53], FHI-AIMS [21], ABINIT [20], ELK [22], and the Crystallographic Information File (CIF) [54]. In the following sections, the variable *GEOM_FILE* can be replaced with files of any name having any of the aforementioned formats; aflow++ will detect their type and process them automatically. To convert between formats, use the following commands, respectively: `--vasp`, `--qe`, `--aims`, `--abinit`, `--elk`, and `--cif`. For example:

```
aflow --qe < POSCAR
```

Note, aflow++ functions are generally overloaded to read inputs from the input stream, enabling commands to be compounded via `bash`'s pipe. For example:

```
aflow --proto=A2B3_hR10_167_c_e-001:Al:0 | aflow --
  ↪ sconv | aflow --aflowSG
```

will create the geometry file for corundum, convert it to the aflow++ Standard Conventional representation [48], and calculate its space group [31].

The aflow.in input file. Calculations performed by aflow++ are controlled via the `aflow.in` file, containing directives to create and run ensembles of density functional theory (DFT) calculations with VASP for the analysis of materials' structural, electronic, thermal, and elastic properties. For a given calculation workflow, options can be specified to control: **i.** the symmetry analyses of the input geometry, **ii.** type and order of VASP calculations, **iii.** schemes fixing VASP errors (and subsequent calculation resubmission), and **iv.** results analysis. Depending on the calculation, VASP input files are generated and organized automatically by aflow++, namely the POSCAR (lattice vectors and atomic positions), INCAR (VASP settings), KPOINTS (k-point grid information), and POTCAR (pseudopotential information). The `aflow.in` file enables high-throughput calculations of material properties in a consistent and repeatable manner, expanding the breadth of materials in the aflow.org repositories.

In general, the `aflow.in` file includes the required machine/compute settings and VASP setup for the calculations. A summary of the specific content in the `aflow.in` file is as follows. First, the system name is given at the top of the file, usually consisting of a string containing the elements, associated pseudopotential designation, and prototype structure. Next, settings to perform the VASP calculation are given, namely the VASP binary name/location, number of computing resources (compute cores/nodes), message passing interface (MPI) settings for calculation parallelization, and commands to launch the VASP application (e.g., `mpirun` and `aprun`). These machine settings are followed by the options for the various aflow++ submodules to calculate different material properties, including:

- crystallographic symmetry (aflow-SYM),
- phonons via the harmonic approximation (aflow-APL),
- phonons via the quasi-harmonic approximation (aflow-QHA),
- anharmonic phonons (aflow-AAPL), and
- thermomechanical properties (aflow-AEL-AGL).

`aflow.in` files can be created automatically, generally by converting the aforementioned `--proto` commands into `--aflow_proto` commands. For example:

```
aflow --aflow_proto=A2B3_hR10_167_c_e-001:Al:0
```

will generate an `aflow.in` within the following directory structure (to avoid writing collisions): `AFLOWDATA/AlO/A2B3_hR10_167_c_e-001.AB`. The first layer, `AFLOWDATA`, is the general container for automatically generated `aflow.in` files. The second layer specifies the chemistry (species), and the third specifies the structure (prototype). The default parameters written inside this automatically-generated `aflow.in` are controlled by the `aflow.rc` and command-line options. More information can be found under the `--aflow_proto` command in the following README:

```
aflow --readme=aconvasp
```

3. Ab-initio calculations

Ab-initio structure-energy calculations remain by far the most time- and resource-intensive component of aflow++'s workflows. As such, substantial efforts have been devoted to integrating with VASP, the default ab-initio software employed by aflow++. VASP offers well-tuned default settings, especially for their pseudopotentials [55], that ensure fast convergence and high accuracy of results without much need for additional customization. aflow++ also provides support for other ab-initio software and frameworks used in the community, especially for structure characterization and manipulation. Beyond this dedicated functionality, aflow++'s alien mode enables the execution of any binary in high-throughput fashion [50].

The full documentation for running automated ab-initio calculations with aflow++, including parameter-tuning and error-handling, can be found in the aflow++ README:

```
aflow --readme=aflow
```

Standard calculation types and protocols. aflow++ offers three basic run schemes for ab-initio calculations with VASP: RELAX, STATIC, and BANDS. By default, two relaxations are performed to ensure structural convergence, which is specified in the `aflow.in` with `[VASP_RUN]RELAX=2`. To incorporate a STATIC and BANDS run into the workflow, the setting should be modified to `[VASP_RUN]RELAX_STATIC_BANDS=2`. The run schemes are described below.

A RELAX run optimizes the geometry of the structure — either by minimizing the energy (default setting) or the forces (changed in the `aflow.in` with `[VASP_FORCE_OPTION]RELAX_MODE=FORCES`) — while trying to converge the electronic charge density at each structure-snapshot. Convergence of the electronic charge density is

facilitated by smearing techniques (ISMEAR and SIGMA in VASP [56]) allowing partial occupancy of orbitals at the Fermi edge, controlled in the `aflow.in` by `[VASP_FORCE_OPTION] TYPE=DEFAULT`, which also takes values of METAL and INSULATOR. For high-throughput calculations, the material assumes the parameters of a metal as the default: using the method of Methfessel–Paxton with first order corrections [57] and a width of 0.1 eV. This sets the following parameters in the INCAR:

```
ISMEAR=1
SIGMA=0.1
```

Within the `aflow++` workflow, the components of the stress tensor are checked after the final relaxation; if any are in excess of 10 kBar, the calculations are automatically rerun with increased precision and cutoffs for the plane-wave basis set [58]. As a standard, `aflow++` runs spin-polarized calculations with initial magnetic moments for all atoms set to 1.0 μ_B /atom; a good default if the magnetic properties of the system are unknown. Spin-polarization is turned off to reduce computational resources if the magnetization resulting from the second relaxation is found to be below 0.05 μ_B /atom. These settings are found in the `aflow.in`:

```
[VASP_FORCE_OPTION] SPIN=ON, REMOVE_RELAX_2
# [VASP_FORCE_OPTION] AUTO_MAGMOM=ON
```

`[VASP_FORCE_OPTION] AUTO_MAGMOM=ON` would change the initial magnetic moments from 1.0 to 5.0 μ_B /atom, and is turned off (commented out) by default. To change how these settings are written with the automatic generation of `aflow.in` files, the following variables should be tuned in the `aflow.rc`:

```
DEFAULT_VASP_FORCE_OPTION_SPIN=1
DEFAULT_VASP_FORCE_OPTION_SPIN_REMOVE_RELAX_1=0
DEFAULT_VASP_FORCE_OPTION_SPIN_REMOVE_RELAX_2=1
DEFAULT_VASP_SPIN_REMOVE_CUTOFF=0.05
DEFAULT_VASP_FORCE_OPTION_AUTO_MAGMOM=0
```

A STATIC run converges the electronic charge density of a fixed geometry with settings that ensure a high-resolution calculation of the total energy and electronic DOS. This is accomplished by running with a high k -point density and performing Brillouin-zone integrations with the tetrahedron method with Blöchl corrections [59] as a default. The k -point density is controlled by the k -points per reciprocal atom (KPPRA) [13] parameter, defining a grid size that scales inversely with the number of atoms. The parameter can be adjusted in the `aflow.in`:

```
[VASP_KPOINTS_FILE] STATIC_KPPRA=10000
```

A BANDS run uses the well-converged electronic charge density from the STATIC run and calculates the energy levels along the `aflow++` standard k -paths to resolve the full electronic band structure and associated properties, such as the band gap. Standard semi-local DFT (with, e.g., the PBE functional) tends to underestimate the band gap [25,60], an issue that is addressed in `aflow++` with the DFT+ U approach based on the formulations developed by Liechtenstein [61] and Dudarev [62]. The relevant `aflow.in` parameters are:

```
[VASP_FORCE_OPTION] LDAU2=ON
[VASP_FORCE_OPTION] LDAU_PARAMETERS=Ga,Sb
↪ ;2,-1;3.9,0;0,0
```

where `[VASP_FORCE_OPTION] LDAUi=ON` chooses the formulation as developed by Liechtenstein ($i=1$) or Dudarev ($i=2$; default). `[VASP_FORCE_OPTION] LDAU_PARAMETERS` is a semicolon-separated string of DFT+ U on-site interaction parameters for each species (comma-separated), namely the l -quantum number and the strengths of the effective Coulomb (U) and exchange (J) interactions [56]. Note the values for Sb for which no DFT+ U corrections will be applied: -1, 0, 0. The aforementioned `aflow.in` parameters would generate the following lines in the INCAR:

```
LDAU=.TRUE.
#LDAU_SPECIES=Ga Sb
LDAUL=2 0
LDAUU=3.9 0
LDAUJ=0 0
LDAUTYPE=2
LMAXMIX=4
```

The parameters for several systems have been defined as part of the `aflow++` standard [13,48]. For alloys having DFT+ U parameters, an `aflow.in` will automatically be generated with `[VASP_FORCE_OPTION] LDAU2=ON`. To turn off this behavior, add `--noldau` to the `--aflow_proto` command.

The default parameters for these run schemes, including convergence tolerances, grid densities, and pseudopotential choices, have been defined as part of the `aflow++` standard detailed in Ref. [13]. These include: i. a VASP precision set to the highest pre-defined setting [56] (`[VASP_FORCE_OPTION] PREC=ACCURATE` in the `aflow.in`, `DEFAULT_VASP_FORCE_OPTION_PREC_SCHEME=ACCURATE` in the `aflow.rc`), ii. a plane-wave basis cutoff increased by a factor of 1.4 above that set by VASP [56] (`[VASP_FORCE_OPTION] ENMAX_MULTIPLY=1.4` in the `aflow.in`, `DEFAULT_VASP_PREC_ENMAX_ACCURATE=1.4` in the `aflow.rc`), and iii. the stable Davidson blocked scheme for diagonalizing the Hamiltonian [63,64] (`[VASP_FORCE_OPTION] ALGO=NORMAL` in the `aflow.in`, `DEFAULT_VASP_FORCE_OPTION_ALGO_SCHEME=NORMAL` in the `aflow.rc`). As demonstrated, settings are highly tunable through a combination of the `aflow.in` and `aflow.rc` files. In addition to the `[VASP_FORCE_OPTION]` keys, the `aflow.in` offers explicit and implicit control of VASP input files. For example,

```
[VASP_INCAR_MODE_EXPLICIT] START
SYSTEM=Ga_hGe_h.11
PSTRESS=000 # for hand modification
#NBANDS=XX # for hand modification
#IALGO=48 # for hand modification
[VASP_INCAR_MODE_EXPLICIT] STOP
[AFLOW] *****
[VASP_KPOINTS_MODE_IMPLICIT]
[VASP_KPOINTS_FILE] KSCHEME=M
[VASP_KPOINTS_FILE] KPPRA=6000
[VASP_KPOINTS_FILE] STATIC_KSCHEME=M
[VASP_KPOINTS_FILE] STATIC_KPPRA=10000
[VASP_KPOINTS_FILE] BANDS_LATTICE=AUTO
[VASP_KPOINTS_FILE] BANDS_GRID=20
```

where `[VASP_INCAR_MODE_EXPLICIT]` allows direct injection of content into the INCAR, and `[VASP_KPOINTS_MODE_IMPLICIT]` defines a namespace of keys that control the creation of the KPOINTS files. These settings are generally overridden by `[VASP_FORCE_OPTION]` settings, unless `[VASP_FORCE_OPTION] NOTUNE` is provided (and uncommented).

Error-handling. `aflow++` offers automatic VASP error detection and correction, with various treatment routes accessible depending on the errors and the order in which they are encountered. `aflow++` is efficient in its application of the corrections, only applying the ones that have not been tried before or would conflict with previous ones. Treatments include modifications of the diagonalization algorithm, precision, k -points grid and scheme, and rescaling of the atomic distances during relaxations. Errors are detected in the `vasp.out` file (VASP's standard output) and corrections are documented in the LOCK file. The treatments and order in which they are applied have been heavily tested and optimized to minimize the need of human intervention. In the event that `aflow++` over-corrects, the following command can be added to the `aflow.in`:

```
[VASP_FORCE_OPTION] IGNORE_AFIX=ERROR:DENTET, FIX:ALGO
↪ =FAST
```

which will ignore VASP's DENTET error and not apply any treatment which includes changing the algorithm to the FAST preset [56]. The full list of errors detected and treatments available can be found under the IGNORE_AFIX section of the aflow++ README:

```
aflow --readme=aflow
```

4. Structural analysis

4.1. aflow++ standard cell representations

Determination of a standard cell representation is essential to an autonomous workflow, and is usually the first step. aflow++ employs the standard primitive and standard conventional cells as defined in Ref. [48].

Standard primitive cell. The aflow++ standard primitive representations for all Bravais lattices have been constructed to have corresponding Minkowski-reduced lattices in the reciprocal space, ensuring both speed and convergence of electronic structure calculations using plane-wave bases [65,66]. For example, the standard primitive representations for fcc copper and β -tin are

```
POSCAR of SPRIM Cu/A_cF4_225_a
1.0
0.000000 1.815202 1.815202
1.815202 0.000000 1.815202
1.815202 1.815202 0.000000
1
Direct(1) [A1]
0.000000 0.000000 0.000000 Cu
```

```
POSCAR of SPRIM Sn/A_tI4_141_a
1.0
-2.933827 2.933827 1.600696
2.933827 -2.933827 1.600696
2.933827 2.933827 -1.600696
2
Direct(2) [A2]
0.125000 0.875000 0.750000 Sn
0.875000 0.125000 0.250000 Sn
```

Such a lattice is guaranteed to be composed of the three smallest linearly-independent vectors — and is thus maximally compact — and have a bounded orthogonality defect, where an orthogonal basis has a defect of zero. The orthogonality defect is associated with the loss of completeness of a truncated plane-wave basis, needing more plane-wave terms (and computational resources) to reach the required accuracy. Note that in the search for the most primitive lattice, vectors defined by the atomic basis are considered, so the shape of the lattice may change in such a way that the symmetry of the crystal is still preserved [31]. A structure can be converted to the aflow++ standard primitive representation with the following command:

```
aflow --sprim < GEOM_FILE
```

Standard conventional cell. The aflow++ standard conventional representations for all Bravais lattices have been constructed to highlight symmetry properties of the lattices (e.g., defining lattice vectors along high-symmetry directions). For example, the standard conventional representations for fcc copper and β -tin are

```
POSCAR of SCONV Cu/A_cF4_225_a
1.0
3.630405 0.000000 0.000000
0.000000 3.630405 0.000000
0.000000 0.000000 3.630405
4
Direct(4) [A4]
0.000000 0.000000 0.000000 Cu
0.500000 0.500000 0.000000 Cu
0.500000 0.000000 0.500000 Cu
0.000000 0.500000 0.500000 Cu
```

```
POSCAR of SCONV Sn/A_tI4_141_a
1.0
5.867655 0.000000 0.000000
0.000000 5.867655 0.000000
0.000000 0.000000 3.201393
4
Direct(4) [A4]
0.750000 0.000000 0.125000 Sn
0.250000 0.500000 0.625000 Sn
0.250000 0.000000 0.875000 Sn
0.750000 0.500000 0.375000 Sn
```

The standard conventional representations expose the 4-fold symmetry axes of the structures, while the standard primitive representations minimize the number of atoms.

A structure can be converted to the aflow++ conventional primitive representation with the following command:

```
aflow --sconv < GEOM_FILE
```

Many aflow++ standard conventional representations match with those defined in the International Tables for Crystallography (ITC) [67], with others making use of different, equally-valid choices; such is the case for the monoclinic system. Since the Wyckoff positions are standardized with respect to the ITC conventional cells, aflow++ is also able to generate structures in the ITC representation with the following command:

```
aflow --itc < GEOM_FILE
```

where the --itc flag can be appended with other output formats, e.g.,

```
aflow --itc --qe < GEOM_FILE
```

to convert the structure to the QUANTUM ESPRESSO geometry format.

By default, aflow++ will convert structures to the aflow++ standard primitive representation before running an ab-initio calculation. This setting is controlled with the following line in the aflow.in:

```
[VASP_FORCE_OPTION]CONVERT_UNIT_CELL=SPRIM
```

which also takes SCONV (aflow++ conventional representation), NIGGLI (Niggli standard form [68,69], which is unique and thus used to determine the lattice type), MINK (Minkowski-reduced lattice), INCELL (moving atoms inside the inequivalent unit cell), COMPACT (moving atoms to reduce distance between them and expose bonds), WS (Wigner-Seitz cell), CART/Frac (Cartesian/direct coordinates), PRES (no modification of input structure). The aflow++ standard conventional representation has been useful for phonon calculations (via finite-displacement), achieving more spherical supercells that include more full coordination shells while keeping cell sizes as small as possible (see Section 7.2) [34].

The various reduction algorithms can yield different primitive cells. Here are some examples based on α -boron structure prototype (A_hR12_166_2h) [15–17]:

```
aflow --proto=A_hR12_166_2h-001:B | aflow --vasp
```

```
POSCAR of SPRIM B/A_hr12_166_2h-001
1.0
1.266100 -0.730983 2.161001
0.000000 1.461966 2.161001
-1.266100 -0.730983 2.161001
12
Direct(12) [A12]
0.010400 0.010400 0.657290 B
0.657290 0.010400 0.010400 B
0.010400 0.657290 0.010400 B
-0.657290 -0.010400 -0.010400 B
-0.010400 -0.010400 -0.657290 B
-0.010400 -0.657290 -0.010400 B
0.220600 0.220600 0.632300 B
0.632300 0.220600 0.220600 B
0.220600 0.632300 0.220600 B
-0.632300 -0.220600 -0.220600 B
-0.220600 -0.220600 -0.632300 B
-0.220600 -0.632300 -0.220600 B
```

```
aflow --proto=A_hr12_166_2h-001:B | aflow --niggli |
↪ aflow --vasp
```

```
POSCAR of NIGGLI B/A_hr12_166_2h-001
1.0
1.266100 2.192949 0.000000
-1.266100 2.192949 0.000000
0.000000 1.461966 2.161001
12
Direct(12) [A12]
0.342710 0.989600 0.678090 B
0.989600 0.342710 0.678090 B
0.989600 0.989600 0.678090 B
0.010400 0.657290 0.321910 B
0.657290 0.010400 0.321910 B
0.010400 0.010400 0.321910 B
0.367700 0.779400 0.073500 B
0.779400 0.367700 0.073500 B
0.779400 0.779400 0.073500 B
0.220600 0.632300 0.926500 B
0.632300 0.220600 0.926500 B
0.220600 0.220600 0.926500 B
```

```
aflow --proto=A_hr12_166_2h-001:B | aflow --mink |
↪ aflow --vasp
```

```
POSCAR of MINK B/A_hr12_166_2h-001
1.0
-1.266100 -0.730983 2.161001
2.532199 0.000000 0.000000
1.266100 2.192949 0.000000
12
Direct(12) [A12]
0.678090 0.010400 0.010400 B
0.678090 0.657290 0.010400 B
0.678090 0.010400 0.657290 B
-0.678090 -0.657290 -0.010400 B
-0.678090 -0.010400 -0.010400 B
-0.678090 -0.010400 -0.657290 B
1.073500 0.220600 0.220600 B
1.073500 0.632300 0.220600 B
1.073500 0.220600 0.632300 B
-1.073500 -0.632300 -0.220600 B
-1.073500 -0.220600 -0.220600 B
-1.073500 -0.220600 -0.632300 B
```

4.2. aflow-SYM: The crystal symmetry module

Identifying crystallographic symmetries. To identify the isometries of a crystal structure, candidate symmetries are applied to the atomic positions in the unit cell ($\{\mathbf{x}\}$). A structure exhibits that symmetry if all transformed atomic positions ($\{\mathbf{x}_{\text{transformed}}\}$) map one-to-one with the original positions ($\{\mathbf{x}_{\text{original}}\}$). In general, the transformed and original atomic positions will: i. match exactly (ideal mapping), ii. significantly differ (no mapping), or iii. slightly differ (possible mapping) (depicted in Fig. 1(a)). To determine whether the transformed and original atoms map, a threshold, ϵ_{sym} , is employed:

$$\|\mathbf{x}_{\text{orig}} - \mathbf{x}_{\text{transformed}}\| \leq \epsilon_{\text{sym}}, \forall \mathbf{x} \in \{\mathbf{x}\}.$$

For crystals, atom mappings in periodic systems are considered via the method of images [70]. However, determination of the minimum distance — required to identify the correct one-to-one mappings — is often computationally expensive in Cartesian (Euclidean) space. Algorithms to minimize distances in fractional (direct, non-Euclidean) space are generally faster; however, incorrect minimum distances (mappings) are possible since the metric tensor in this space is not uniform. The problem is exacerbated in systems with skewed lattices; e.g., a triclinic lattice ($\alpha \neq \beta \neq \gamma \neq 90^\circ$) in Cartesian coordinates will be converted to a unit cube in fractional coordinates, warping the space (as shown in Fig. 1(b)). To determine the impact of the warping, the aflow-SYM module compares the lattice skewness and minimum interatomic distance in Cartesian space $d_c^{\text{nn,min}}$ to the symmetry tolerance:

$$\epsilon_{\text{max}} \equiv [1 - \max(\cos\alpha, \cos\beta, \cos\gamma)] d_c^{\text{nn,min}}.$$

If ϵ_{sym} is below ϵ_{max} , the minimum distance algorithm in fractional space (faster) should yield the correct mappings. Above this value, the warping may yield incorrect mappings and the slower, but more reliable minimization in Cartesian space is required. This heuristic was validated for the more than 3.5 million entries in the aflow.org repositories at the time of writing.

Symmetry tolerance. To alleviate the burden of identifying suitable symmetry tolerances, aflow++ offers two preset values: $\epsilon_{\text{sym}}^{\text{tight}} = d_c^{\text{nn,min}}/100$ (default) and $\epsilon_{\text{sym}}^{\text{loose}} = d_c^{\text{nn,min}}/10$. These presets can be used in any of the symmetry commands by appending =tight or =loose, respectively (see the aflow-SYM README for examples). Since the tolerance thresholds are based on the minimum interatomic distance in the crystal, they are system-specific and generally more consistent with experimental space group determinations (see the Validation with experiments subsection). Alternatively, users can input their own numerical tolerance values, as long as they are below the minimum interatomic distance in the crystal.

Additionally, to ensure consistent symmetry descriptors for a wide range of tolerance values, aflow-SYM employs an adaptive tolerance scan (Fig. 1(c)). Namely, aflow-SYM checks that descriptors are commensurate with group theory and crystallographic conventions (see Ref. [31] for details). If the checks fail at a particular tolerance value (ϵ_0), aflow-SYM will perform a radial tolerance scan (i.e., in $\pm$ increments around the initial value) and recalculate the symmetries at new values (ϵ_{new}). The process continues until consistent symmetry is found at a new tolerance value. This procedure reduces errors between symmetry descriptions and mitigates the need for users to tune tolerances to obtain accurate results.

Symmetry calculator. aflow-SYM identifies the entire symmetry profile of a crystal in any arbitrary unit cell representation. These routines determine all crystallographic symmetry groups afforded by group theory in both real and reciprocal spaces. In particular, the following groups are calculated: point groups (real lattice, reciprocal lattice, Patterson, and atom-centered), factor group representatives (unit cell), and space groups. The different symmetry groups and their aflow++ aliases (in parentheses) include:

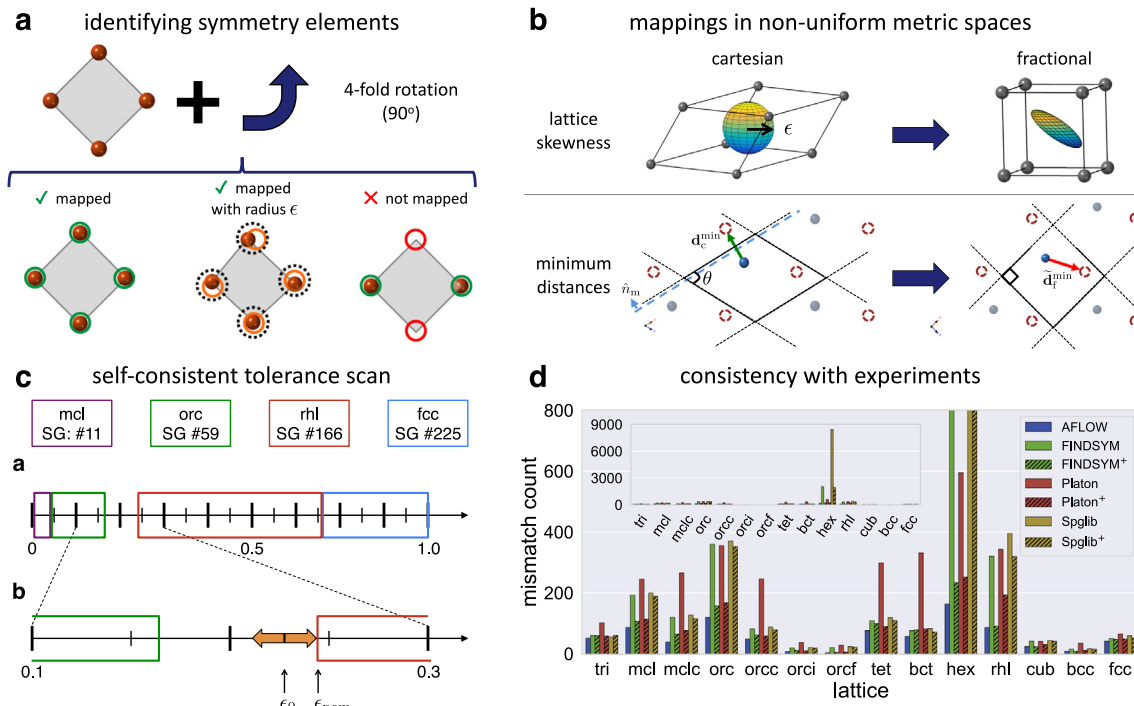


Fig. 1. Overview of aflow-SYM functionality and example of high-accuracy results. (a) A schematic highlighting how a symmetry element is tested on an arrangement of atoms. Outcomes of mapping the original atomic positions (solid circles) to their transformed counterparts (hollow circles) are grouped into the following categories: atom positions exactly map (left), atoms positions map within a tolerance ϵ (middle), or some/all of the atomic positions are not mapped (right). (b) Examples showing how distances and tolerances change between Cartesian and fractional coordinates. aflow-SYM considers lattice skewness in its mapping procedure to ensure it is correct regardless of the coordinate system. (c) An illustration of the different space groups found during a tolerance scan. (d) Results from a benchmark study, highlighting the accuracy of space groups calculated with different software packages compared to experiment (information provided by the Inorganic Crystal Structure Database). aflow-SYM has the fewest mismatches with experiment (best accuracy) across all 14 Bravais lattice types.

- **Lattice point group (pgroup):** Describes the point group symmetry (rotations, inversion, and roto-inversions) of the lattice points.
- **Reciprocal lattice point group (pgroupk):** Describes the point group symmetry (rotations, inversion, and roto-inversions) of the reciprocal lattice (i.e., Brillouin zone).
- **Crystallographic point group (pgroup_xtal):** Describes the point group symmetry of the lattice faces [67,71].
- **Dual of the crystallographic point group (pgroupk_xtal):** Describes the point group symmetry of the dual of the crystallographic point group (i.e., irreducible Brillouin zone).
- **Patterson point group (pgroupk_Patterson):** Describes the Patterson point group symmetry in reciprocal space, i.e., symmetry of the inter-atomic vectors.
- **Factor group representative (fgroup):** Describes the rotations, inversion, roto-inversions, screws, and glides of the unit cell. Note, the factor group representative itself is not a group: the closure axiom is violated since the lattice translations are not present in the group [72].
- **Space group (sgroup):** Describes the rotations, inversion, roto-inversion, screws, and glides of the entire periodic crystal. Since the space group represents the symmetry of an infinite object, a finite number of space group operations are calculated in aflow-SYM (see Ref. [31] for details).
- **Atom-site point group (agroup):** Describes the point group symmetry centered on each atom in the unit cell. To make the calculation more efficient, aflow-SYM only calculates the atom point group operations for symmetrically-inequivalent atoms and transforms them for the other atoms. The symmetrically-equivalent atoms are grouped into an `iatoms` object (analogous to Wyckoff positions).

Note that the Patterson point group is a new addition featured after the original publication of Ref. [31]. All the aforementioned symmetry groups are guaranteed to be commensurate with crystallographic conventions due to a variety of consistency checks implemented into the routines. Any discrepancies initiate the adaptive tolerance scheme to change the symmetry tolerance (ϵ_{sym}) and recalculate all symmetry groups until consistency is achieved.

Symmetry representations. All crystallographic symmetry representations are categorized as either **i.** translation, **ii.** fixed-point (rotations, inversion, and roto-inversions), or **iii.** fixed-point free (screws and glides) operations. For each of these symmetry elements, aflow-SYM provides multiple representations to cater to different applications. Translations are represented as 3×1 vectors, fixed-point operations are represented as 3×3 matrices, and fixed-point free elements are a combination of the two. All of these symmetry elements are returned in both Cartesian and fractional (direct) coordinate systems.

aflow-SYM provides additional representations for pure rotations, comprising the $SO(3)$ Lie group [73], including the axis-angle representation, matrix generator ($so(3)$, Lie algebra), quaternion (4×4 matrix and 4×1 matrix), $SU(2)$ Lie group (2×2 complex matrix), and $su(2)$ Lie algebra (2×2 complex matrix).

Consistency with the ITC. In addition to determining the symmetry of any arbitrary unit cell, aflow-SYM determines the space group symmetry and Wyckoff positions, commensurate with the ITC [67]. From this analysis, aflow-SYM determines the space group number, International (Hermann-Mauguin) designation, Schoenflies designation, and Hall symbol. Furthermore, aflow-SYM identifies the symmetrically-equivalent atomic positions — or Wyckoff positions — and returns the corresponding Wyckoff letter designation, multiplicity, site symmetry, and representative Wyckoff coordinate. Space groups can also be converted into special space group settings or unit cell choices for monoclinic, rhombohedral, and centrosymmetric space groups. Namely, conventional cells and Wyckoff positions can be represented via **i.** unique

axis b or c for monoclinic space groups, **ii.** rhombohedral or hexagonal unit cells, or **iii.** origins centered on inversion sites or other high-symmetry sites for centrosymmetric space groups. Lastly, since the choice of Wyckoff positions can differ with lattice and/or origin choices, aflow-SYM prefers the Wyckoff sequence with the smallest enumerated Wyckoff lettering.

Crystal-spin symmetry. The magnetic moment (spin) associated with each atom in a crystal can impact the properties of a crystal. Thus, aflow-SYM calculates the crystal-spin symmetry by incorporating the magnetic moment into the symmetry analysis, acting as a degree of freedom that can break (lower) symmetry. This is analogous to how decorating a lattice with different atomic species lowers the symmetry. Thus, in general, the crystal-spin symmetry forms a subgroup with respect to the crystal symmetry. The analysis distinguishes the symmetry between varying spin configurations (i.e., ferromagnetic, ferrimagnetic, and antiferromagnetic). This type of symmetry analysis is relevant to ab-initio codes — such as VASP — that break orbital symmetry based on spin considerations. The aflow-SYM crystal-spin symmetry routines are designed for both collinear and non-collinear systems.

Validation with experiments. Other software solutions to calculate symmetry are available, including FINDSYM [74], PLATON [75], and Spglib [76], each catering to different symmetry objectives. Compared to space groups determined by experimental methods, aflow-SYM is the most consistent (Fig. 1(d)). aflow-SYM's high-fidelity results are attributed to its **i.** robust mapping scheme for skewed lattices, **ii.** default tolerance values, and **iii.** adaptive tolerance scheme with integrated consistency checks.

Applications. The symmetry routines discussed herein are used throughout the aflow++ codebase to categorize crystallographic structures and reduce the cost of simulations. For example, the point group symmetries are used to determine the high-symmetry k -paths for electronic and phonon band structure calculations. Furthermore, in phonon simulations, the factor group (fgroup) and atom-site point group (agroup) are used to identify the symmetrically-inequivalent atoms and distortion directions, respectively, to reduce the simulation cost.

To enable adoption into user workflows, aflow-SYM features a Python module to call the major symmetry functions in a Python environment. Furthermore, symmetry results can be printed in either human-readable text or JavaScript Object Notation (JSON) for easy manipulation and extension to other environments.

Command-line options. There are three main functions that provide all symmetry information for a given input structure. These functions allow an optional tolerance value (tol) to be specified, accepting a number (double) or the strings `tight` and `loose` corresponding to ϵ_{tight} and ϵ_{loose} , respectively. To perform the symmetry analysis of a crystal, the functions are called with the following commands:

```
aflow --aflowSYM < GEOM_FILE
```

calculates and returns the symmetry operations for the lattice point group, reciprocal lattice point group, factor group representatives, crystal point group, dual of the crystal point group, Patterson symmetry, site symmetry, and space group. It also returns the unique and equivalent sets of atoms. The tolerance can be appended to the aflowSYM option: `--aflowSYM=tol`. The isometries of the different symmetry groups are saved to their own files: `aflow.group.out` or `aflow.group.json`. The `group` labels are as follows: `pgroup` (lattice point group), `pgroupk` (reciprocal lattice point group), `fgroup` (factor group representatives), `pgroup_xtal` (crystal point group), `pgroupk_xtal` (dual of the crystal point group), `pgroupk_Patterson` (Patterson point group), `agroup` (site symmetry), and `sgroup` (space group).

```
aflow --edata < GEOM_FILE
```

calculates and returns the extended crystallographic symmetry data (crystal, lattice, reciprocal lattice, and superlattice symmetry), while incorporating the full set of checks for robust symmetry determination. The tolerance can be appended to the `edata` option: `--edata=tol`.

```
aflow --sgdata < GEOM_FILE
```

calculates and returns the space group symmetry of the crystal, while only validating the symmetry descriptions matching with the ITC conventions. The tolerance can be appended to the `sgdata` option: `--sgdata=tol`. The `--print` option specifies the output format — accepting `txt` (default) or `json` — and can be appended to the aflowSYM, `edata`, and `sgdata` commands: e.g., `--print=json`.

4.3. aflow-XtalFinder: The crystal prototypes module

Autonomous prototype finder. To identify the prototype of a given crystallographic compound, aflow-XtalFinder computes the structure's Pearson symbol, space group, and Wyckoff positions via aflow-SYM routines [31] (Fig. 2(a)). With the underlying aflow-SYM adaptive tolerance mechanism, prototype designations are guaranteed to be consistent, automatically changing the symmetry tolerance otherwise. The default tolerance for the symmetry analysis is $\epsilon_{\text{sym}} = d_{\text{nn}}^{\text{min}}/100$, where $d_{\text{nn}}^{\text{min}}$ is the minimum interatomic distance within the crystal. Based on benchmarks performed in Ref. [31], this value is consistent with experimentally-resolved space group symmetries.

aflow-XtalFinder casts these structures into their ideal prototype label and degrees of freedom based on the aflow++ Prototype Encyclopedia representation [15–17]. To determine the prototype label, aflow-XtalFinder calculates the reduced stoichiometry, lattice, and Wyckoff positions (via aflow-SYM routines) [31]. Once the symmetry is calculated, aflow-XtalFinder identifies which lattice (i.e., a , b/a , c/a , α , β , and γ) and Wyckoff parameters (i.e., x , y , and z for all Wyckoff positions) are not fixed by symmetry and returns their values for the particular geometry. For example, the wurtzite structure has the prototype label `AB_hP4_186_b_b` and its corresponding degrees of freedom (a , c/a , z_1 and z_2) are 3.82, 1.63, 0.3748, and 0, respectively.

This particular representation of a prototype is advantageous because: **i.** it is consistent with the ITC (the *de facto* standard) [67], **ii.** it allows users to tune the degrees of freedom while preserving the underlying symmetry, and **iii.** geometry files for any arbitrary structure in this designation can be generated using aflow++'s symbolic prototype generator. Furthermore, this representation has been fruitful in performing symmetry-constrained relaxations [77].

Degrees of similarity. aflow-XtalFinder performs symmetry, local atomic geometry, and complete geometric comparisons to analyze structural similarity to varying degrees (Fig. 2(b)). Symmetry comparisons are performed to identify structures that are isopointal (same symmetry). This is done by calculating the space group symmetries and Wyckoff positions of the relevant structures with aflow-SYM [31]. Structures are considered isopointal if **i.** their space groups are the same or form an enantiomorphic pair (mirror image) and **ii.** their Wyckoff sequences are similar (related via an automorphism of the space group) [67,78,79]. aflow-XtalFinder tests Wyckoff similarity by comparing the Wyckoff multiplicities and permuting the site symmetry designation.

Isoconfigurational snapshots: comparing local geometries. Beyond isopointal analyses, structures are further compared by inspecting arrangements of atoms, i.e., local atomic geometries. Routines to quickly identify local geometries are employed here to analyze structures beyond symmetry considerations (i.e., isoconfigurational *versus* isopointal [80]).

Rather than determine the complete local atomic geometry for each atom, XtalFinder builds a reduced representation: neighborhoods comprised of only the least frequently occurring atom (LFA) types. The local LFA geometry analysis provides the connectivity for a subset of atoms

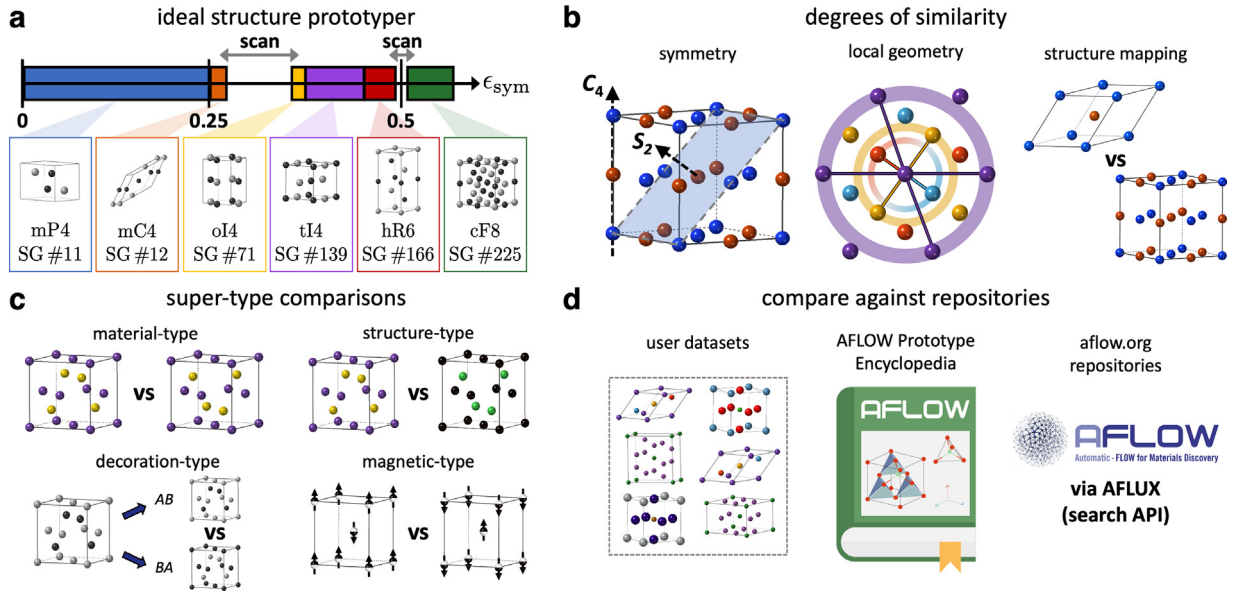


Fig. 2. Ideal prototyper and structure comparison tools in aflow-XtalFinder. (a) The ideal prototyper determines a structure's prototype label and degrees of freedom based on the aflow-SYM symmetry analyses, employing a tolerance scan scheme if incommensurate descriptions are found. (b) Structures are compared to varying degrees of similarity via symmetry, local geometry, and geometric structure mapping. (c) aflow-XtalFinder enables different structure comparison modes: material-type (map like atoms), structure-type (map any atom types of equal stoichiometry), decoration-type (generate and compare different atom colorings on a structure), and magnetic-type (map alike atoms and magnetic moments). (d) Input structures can be automatically compared to user datasets, the aflow++ Prototype Encyclopedia, or the aflow.org repositories.

(i.e., LFA type) to discern if patterns are present in both structures, regardless of cell choice and crystal orientation. This description is preferred over the full local geometry because it is **i.** computationally less expensive to calculate and **ii.** generally less sensitive to coordination cutoff tolerances. The latter is attributed to the fact that LFA geometries are more sparse.

A local LFA atomic geometry (AG) is a set of vectors connecting a central atom (c) to its closest neighbors:

$$AG_c \equiv \{d_{ic}^{\min} \mid \forall i \mid \text{atom}_i \in \{\text{LFAs}\}\},$$

where $d_{ic}^{\min}$ is the minimum distance vector to the i -atom — restricted to LFA types only — and is calculated via the method of images for periodic systems [70]:

$$d_{ic}^{\min} = \min_i \left[\min_{n_a, n_b, n_c} \|\mathbf{x}_i - \mathbf{x}_c + n_a \mathbf{a} + n_b \mathbf{b} + n_c \mathbf{c}\| \right].$$

Here, n_a , n_b , and n_c are the lattice dimensions along the lattice vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$; and $\mathbf{x}_i$ and $\mathbf{x}_c$ are the Cartesian coordinates of the i and c (center) atoms, respectively. A coordination shell with a thickness of $d_{ic}^{\min}/10$ captures other atoms of the same type to control numerical noise in the atomic coordinates (a similar tolerance metric is defined in aflow-SYM, i.e., loose preset tolerance value [31]). This cutoff value yields expected coordination numbers for well-known systems and is comparable to results provided by other atomic environment calculators [36,81]. If there is only one LFA type — e.g., Si in α -cristobalite (SiO_2 , [A2B_tP12_92_b_a](#)) [15–17] — then the distance to the closest neighbor of that LFA type is calculated. If there are multiple LFA types — e.g., four for the quaternary Heusler — then the minimum distances to each LFA type are computed. The local atomic geometry is calculated for each atom of the LFA types in the unit cell, resulting in a list of atomic geometries ($\{AG_c\}$). Therefore, α -cristobalite has a set of four Si LFA geometries (one for each Si in the unit cell: $\{AG_{\text{Si},1}, AG_{\text{Si},2}, AG_{\text{Si},3}, AG_{\text{Si},4}\}$) and the quaternary Heusler has a set of four LFA geometries (one for each element type: $\{AG_{\text{Au}}, AG_{\text{Li}}, AG_{\text{Mg}}, AG_{\text{Sn}}\}$, respectively).

To investigate structural compatibility, local atomic geometry lists for compounds are compared. In general, the local geometry comparisons err on the side of caution. For instance, comparing the cardinality of the coordination is often too strict. Despite a sparser geometry space,

slight deviations in position can move atoms outside the coordination shell threshold, changing the atom cardinality and overlooking potential matches. Local atomic geometries are thus compatible if **i.** the central atoms are of comparable types (i.e., same element and/or stoichiometric ratio in the crystal), **ii.** the neighborhoods of surrounding atoms have distances that match within 20% after normalizing with respect to $\max(AG_c)$ (i.e., the largest distance in the local geometry cluster), and **iii.** the angles formed by two atoms and the center atom match within 10 degrees. To further alleviate the coordination problem, an exact geometry match is not required, i.e., some distances and angles need not match exactly at this stage. This more lenient method favors labeling structures as near-isoconfigurational to mitigate false negatives in subsequent geometric comparisons.

Isoconfigurational structures: comparing full geometry. To resolve a commensurate representation between two structures for geometric comparison, one structure — the reference $\mathbb{X}_{\text{ref}}$ — remains fixed and the other structure — the potential duplicate $\mathbb{X}_{\text{test}}$ — is expanded into a supercell. Lattice vectors are identified within the supercell and compared against the reference structure. For any similar lattices to $\mathbb{X}_{\text{ref}}$, $\mathbb{X}_{\text{test}}$ is transformed into the new lattice representation ($\tilde{\mathbb{X}}_{\text{test}}$). Origin shifts for this cell are then explored in an attempt to match atoms. If one-to-one atom mappings exist between the two structures, then the similarity is quantified with the crystal misfit method (see subsection below) [82]. Misfit values below a given threshold indicate that structures match and the search terminates. Alternatively, misfit values larger than the threshold are disregarded and the search continues until all lattices and origin shifts are exhausted.

Quantitative similarity measure. To compare two crystals in a given representation, a method proposed by Burzlaff and Malinovsky is employed [82]. The similarity between structures is quantified by a misfit value [82]:

$$\epsilon \equiv 1.0 - (1.0 - \epsilon_{\text{latt}}) (1.0 - \epsilon_{\text{coord}}) (1.0 - \epsilon_{\text{fail}}),$$

which incorporates differences between lattice vectors (ϵ_{latt}) and atomic positions (ϵ_{coord} and ϵ_{fail}) and are defined below. The misfit quantity is bound between zero and one, where zero indicates a perfect match. Special misfit ranges defined by Burzlaff and Malinovsky are adopted [82]: match ($0 < \epsilon \leq \epsilon_{\text{match}}$), same family ($\epsilon_{\text{match}} < \epsilon \leq \epsilon_{\text{family}}$),

and no match ($\epsilon_{\text{family}} < \epsilon \leq 1$). The “same family” designation generally corresponds to crystals with common symmetry subgroups. Burzlaff and Malinovsky recommend $\epsilon_{\text{match}} = 0.1$ and $\epsilon_{\text{family}} = 0.2$ based on definitions from Pearson [83] and Parthé [84]. In the XtalFinder article, heuristic misfit thresholds are identified based on the allowed maximum enthalpy differences between similar structures (see Ref. [14] for details).

The deviation of the lattices (ϵ_{latt}) captures the difference between the lattice face diagonals of $\tilde{\mathbb{X}}_{\text{test}}$ and $\mathbb{X}_{\text{ref}}$ [82]:

$$\epsilon_{\text{latt}} \equiv 1 - (1 - D_{12})(1 - D_{23})(1 - D_{31}),$$

$$D_{kl} \equiv \frac{\|\tilde{\mathbf{d}}_{kl}^{\text{test}} - \mathbf{d}_{kl}^{\text{ref}}\| + \|\tilde{\mathbf{f}}_{kl}^{\text{test}} - \mathbf{f}_{kl}^{\text{ref}}\|}{\|\mathbf{d}_{kl}^{\text{ref}} - \mathbf{f}_{kl}^{\text{ref}}\|},$$

where $\mathbf{f}_{kl}$ and $\mathbf{d}_{kl}$ (the diagonals on the kl lattice face) are calculated by adding and subtracting, respectively, the k and l lattice vectors. In the lattice search algorithm, Δl , $\Delta\theta$, and ΔV tolerances are coupled to ϵ_{latt} , and are tuned to ensure $\epsilon_{\text{latt}} \leq \epsilon_{\text{family}}$.

The coordinate deviation — measuring the disparity between atomic positions in the two structures — is based on the mapped atom distances (d_i^{map} or d_j^{map}) and the atoms' nearest neighbor distances in the respective structures (d_{nn}) [82]:

$$\epsilon_{\text{coord}} \equiv \frac{\sum_i^{\tilde{N}^{\text{test}}} (1 - \tilde{n}_i^{\text{test}}) d_i^{\text{map}} + \sum_j^{N^{\text{ref}}} (1 - n_j^{\text{ref}}) d_j^{\text{map}}}{\sum_i^{\tilde{N}^{\text{test}}} (1 - \tilde{n}_i^{\text{test}}) d_{\text{nn},i}^{\text{test}} + \sum_j^{N^{\text{ref}}} (1 - n_j^{\text{ref}}) d_{\text{nn},j}^{\text{ref}}}.$$

$\tilde{N}^{\text{test}}$ and N^{ref} are the number of atoms in the two crystals. If $d^{\text{map}} < d_{\text{nn}}/2$, then a “switch” variable n is set to zero and the mapped atom distance is included in ϵ_{coord} . Otherwise, n is set to one, signifying that the mapped atoms are far apart and not considered in ϵ_{coord} . These atoms are counted in the figure of failure [82]:

$$\epsilon_{\text{fail}} \equiv \frac{\sum_i^{\tilde{N}^{\text{test}}} \tilde{n}_i^{\text{test}} + \sum_j^{N^{\text{ref}}} n_j^{\text{ref}}}{\tilde{N}^{\text{test}} + N^{\text{ref}}}.$$

Super-type comparisons. `afLOW-XtalFinder` offers four modes of comparing crystallographic structures by changing the mapping criteria. These modes are material-type, structure-type, decoration-type, and magnetic-type comparisons (Fig. 2(c)). Material-type comparisons map atoms with matching elements (e.g., Cl→Cl and Na→Na in sodium chloride), revealing duplicate compounds. Structure-type comparisons relax this criteria and attempt to map atoms of any element type as long as they have the same stoichiometric ratio (e.g., Cl→Cl, Cl→Na, or vice versa in sodium chloride). This comparison-type identifies compounds that share the same prototype structure; e.g., periclase (MgO) and sodium chloride (ClNa) are both rocksalt structures. Decoration-type comparisons identify the atom types (or colorings) of a single structure that leave the compound invariant (e.g., swapping Cl and Na in sodium chloride leaves the crystal unchanged). In this analysis, all possible decorations of an n -species structure ($n!$) are created and compared if sites have similar Wyckoff positions. Since this analysis determines the unique atom colorings, prototypes can be decorated such that degenerate compounds are omitted. Lastly, magnetic-type comparisons incorporate the magnetic moment of each atom into the analysis, determining distinct spin configurations (e.g., $\uparrow \rightarrow \uparrow$ and $\downarrow \rightarrow \downarrow$). Comparison of systems with collinear and non-collinear magnetic moments are supported.

Automatic grouping. Automatically comparing structures is necessary for high-throughput classification of unique/duplicate compounds and structure-types. In XtalFinder, compounds are first grouped into isopointal sets by analyzing and comparing the symmetries of the structures, aggregating them by stoichiometry, space groups, and Wyckoff sets (calculated via `afLOW-SYM` [31]). Next, compounds are further partitioned into near-isoconfigurational sets by determining and comparing the local LFA geometries in each structure. Within each near-isoconfigurational group, one representative structure — generally the

first in the set — is compared to the other structures via geometric comparisons and the misfit values are stored. Once the comparisons finish, any unmatched structures (i.e., misfit values greater than ϵ_{match}) are reorganized into new comparison sets. The process is recursive, repeating until all structures have been assembled into matching groups or all comparison pairs have been exhausted. The three comparison analyses are performed in this order for two reasons: **i.** to categorize structural similarity to varying degrees (isopointal, near-isoconfigurational, and isoconfigurational) and **ii.** to efficiently group compounds to reduce the computational cost of the geometric structure comparison. This procedure is the same for material-, structure-, decoration-, and magnetic-type comparisons; however, different atom mapping restrictions are applied depending on the comparison mode.

To enhance calculation speed, multithreading capabilities can be employed. The three computationally intensive procedures — calculating the symmetry, constructing the local LFA geometry, and performing geometric comparisons — are partitioned onto allocated threads, offering significant speed increases for large collections of structures.

Comparison against established repositories. There are three built-in functions to compare multiple structures automatically (Fig. 2(d)): **i.** compare structures provided by a user, **ii.** compare an input structure to prototypes in `afLOW++` [15–17], and **iii.** compare an input structure to entries in the `afLOW.org` repositories.

Compare user datasets. Users can load crystal geometries and compare them automatically with XtalFinder. Options to perform both material-type and structure-type comparisons are available to identify unique compounds and prototypes, respectively. For structure-type comparisons, the unique atom decorations for each representative structure are determined. Once the analysis is complete, XtalFinder groups compatible structures together and returns the corresponding misfit values.

Compare to `afLOW++` prototypes libraries. Given an input structure, this routine returns similar `afLOW++` prototypes along with their misfit values. `afLOW++` contains structural prototypes that can be rapidly decorated for high-throughput materials discovery: 1100 in the Prototype Encyclopedia [15–17] and 1492 in the HTQC [50]. In this method, `afLOW++` prototypes are extracted — based on similar stoichiometry, space group, and Wyckoff positions to the input — and compared to the user's structure. Since only matches to the input are relevant, the procedure terminates before regrouping any unmatched prototypes. The attributes of matched prototypes are also returned, including the prototype label, mineral name, *Strukturbericht* designation, and links to the corresponding Prototype Encyclopedia webpage. The scheme identifies common structure-types with the `afLOW++` prototype libraries or — if no matches are found — reveals new prototypes. Absent prototypes can be characterized automatically in the `afLOW++` standard designation with XtalFinder's prototyping tool.

Compare to `afLOW.org` repositories. Compounds are compared to entries in the `afLOW.org` repositories using the `afLOW++` REST- and AFLUX Search-APIs [58,85]. An AFLUX query (i.e., matchbook and directives) is generated internally and returns database compounds similar to the input structure based on species, stoichiometry, space group, and Wyckoff positions. With the AURL from the AFLUX response, structures for the entry are retrieved via the REST-API. The most relaxed structure is extracted by default; however, options are available to obtain structures at different ab-initio relaxation steps. The set of entries from the database is then compared to the input structure. Similar to the `afLOW++` prototype comparisons, candidate entries are only compared against the input structure, i.e., the procedure terminates without regrouping unmatched entries.

With the underlying AFLUX functionality, material properties can also be extracted, highlighting the structure-property relationship amongst similar materials. For instance, the enthalpy per atom (H_{atom}) for matching database entries are printed by including the `enthalpy_atom` API keyword in the query. Any number or combination of

properties can be queried; available API keywords are located in Ref. [1].

This routine reveals equivalent aflow.org compounds if similar materials exist in the database. As such, it can estimate structural properties *a priori*; before performing any calculations. The estimation is based on the following assumptions: **i.** the matching aflow++ material resides at a local minimum in the energy landscape and **ii.** the input structure relaxes to the same geometry as the matching aflow++ compound, given comparable calculation parameters. The functionality can explore properties that are not calculated for a given entry, but are calculated for an equivalent entry. For example, compounds in aflow++'s prototype catalogs (LIB1, LIB2, LIB3, etc.) do not usually have band structure data; however, corresponding ICSD entries can be found which do provide band structure information. Finally, the method can identify compounds that are absent from the database and prioritize them for future calculation, enhancing the diversity of the aflow.org repositories.

New features. Since the publication of Ref. [14], additional functionality has been added to the XtalFinder module. The transformations for mapped structures can be returned to users by appending the `--print_mapping` option to a comparison command (e.g., `--compare_materials` or `--compare_structures`). The transformation information includes **i.** the basis transformation (3×3 matrix), **ii.** the rotation of the coordinate system (3×3 matrix), **iii.** the origin shift (3×1 vector), and **iv.** the volume scaling factor (scalar) between the two structures. Along with changes to the lattice vectors, the basis transformation accounts for changes in the unit cell and can describe mappings between smaller and larger cells. Furthermore, the atom mapping information is included, indicating which atoms are mapped and their relative distances between the two structures.

Command-line interface. The XtalFinder command-line calls are detailed below. Function descriptions and options are provided following each command.

Prototype commands. A structure (*GEOM_FILE*) is converted into its standard aflow++ prototype label with the command

```
aflow --prototype < GEOM_FILE
```

The parameter variables (degrees of freedom) and corresponding values are also listed. Information about the label and parameters are described in the Refs. [13–15]

Options for this command include `--setting=SETTING`, which specifies the space group setting for the conventional cell/Wyckoff positions. Possible values for *SETTING* include: 1, 2, or aflow. Setting values 1 and 2 generally correspond to the first and second choice listed in the ITC, respectively. The aflow setting follows the choices of the Prototype Encyclopedia: *axis-b* for monoclinic space groups, rhombohedral setting for rhombohedral space groups, and origin centered on the inversion site for centrosymmetric space groups (default: aflow).

Comparison commands.

```
aflow --compare_materials=GEOM_FILES
```

compares a comma-separated list of geometry files of compounds comprised of the same elements and with commensurate stoichiometric ratios, i.e., material-type comparison, returning their level of similarity (misfit value). This method identifies unique and duplicate materials.

```
aflow --compare_structures=GEOM_FILES
```

compares a comma-separated list of geometry files of compounds with commensurate stoichiometric ratios with no requirement of the element type, i.e., structure-type comparison, and returns their level of similarity (misfit value). This method identifies unique and duplicate prototypes. For the material- and structure-type comparisons, there are three input types (examples for `--compare_materials` are shown below):

- `aflow --compare_materials=GEOM_FILES` : *GEOM_FILES* is a comma-separated list of geometry files to compare.
- `aflow --compare_materials -D path` : Path to directory (*path*) containing geometry files to compare.
- `aflow --compare_materials -F=filename` : File (*filename*) containing compounds between delimiters `[VASP_POSCAR_MODE_EXPLICIT]START` and `[VASP_POSCAR_MODE_EXPLICIT]STOP`.

To do the same for structure-type comparisons, swap `--compare_materials` with `--compare_structures` in the commands above.

```
aflow --compare2database < GEOM_FILE
```

compares a structure (*GEOM_FILE*) to aflow.org repositories entries, returning similar compounds and quantifying their levels of similarity (misfit values). Material properties can be extracted from the database (via AFLUX) and printed, highlighting structure–property relationships. This function can perform either material-type comparisons or structure-type comparisons (by adding the `--structure_comparison` option). Options specific to this command include:

- `--properties=keywords` : Specifies the comma-separated properties via their API keyword to print the corresponding values with the comparison results.
- `--catalog=string` : Restricts the database entries to a specific catalog/library (e.g., lib1, lib2, lib3, and icsd).
- `--geometry_file=string` : Compares geometries from a particular DFT relaxation step (e.g., POSCAR.relax1, POSCAR.relax2, and POSCAR.static).

```
aflow --compare2prototypes < GEOM_FILE
```

compares a structure (*GEOM_FILE*) against the aflow++ prototype libraries, returning similar structures and quantifying their levels of similarity (misfit values). Adding the `--catalog=string` flag to this command restricts the prototypes to a specific catalog/library (e.g., aflow or htqc).

```
aflow --isopointal_prototypes < GEOM_FILE
```

returns prototype labels that are isopointal (i.e., similar space group and Wyckoff positions) to the input structure (*GEOM_FILE*). The `--catalog=string` flag also works for this function.

```
aflow --unique_atom_decorations < GEOM_FILE
```

determines the unique and duplicate atom decorations for a given structure.

A full list of the possible commands and options is available in the XtalFinder README, which is printed with the command

```
aflow --readme=xtalfinder
```

5. PAOFLOW electronic analysis

PAOFLOW [26,27] is a software tool to efficiently post-process standard first-principles electronic structure plane-wave pseudopotential calculations. From interpolated band structures and DOS, it promptly computes several quantities that provide insight into transport, optical, magnetic and topological properties like anomalous and spin Hall conductivities, magnetic circular dichroism, spin circular dichroism, and topological invariants. The methodology is based on the projection of the electronic wavefunctions of the system on a local atomic orbitals basis (PAO) [86–88] and is part of the aflow++ software infrastructure [50,89]. Currently, PAOFLOW is interfaced with QUANTUM ESPRESSO [19,53] and the proposed procedure is completely general and can be implemented with any DFT electronic structure engine. Accurate PAO Hamiltonian matrices can be built from the direct projection of the Kohn–Sham Bloch states $|\psi_{nk}\rangle$ onto a chosen basis set

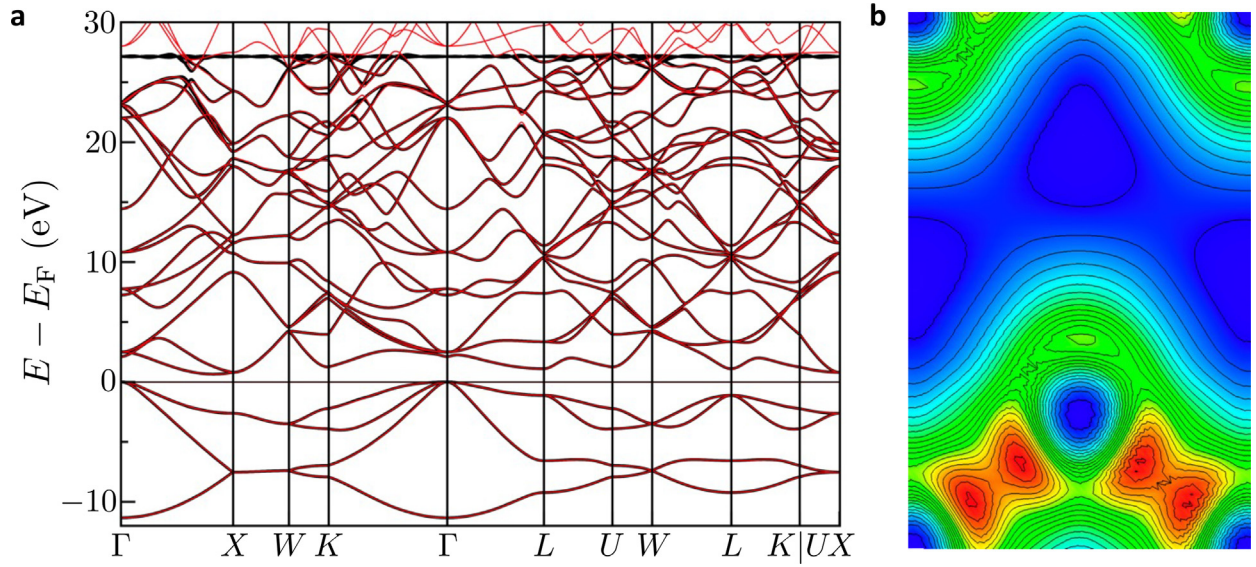


Fig. 3. Electronic band structure and density calculated with PAOFLOW. (a) Silicon band structure calculated with PAOFLOW (black) projecting on the basis set comprising the $3s$, $3p$, $3d$, $4s$, $4p$, $4d$, and $4f$ orbitals, compared to the DFT band structure (red). (b) Silicon electron density reconstructed with PAOFLOW on the same basis set (contour plot on the $[1, 0, -1]$ plane).

of fixed localized functions, as we discussed extensively in Ref. [86–88]. The real space Hamiltonians $H(\mathbf{R})$ ($\mathbf{R}$ is a lattice vector) can be directly calculated using atomic orbitals or pseudo atomic orbitals from the pseudopotential of any given element [86,87]. The crucial quantities that measure the accuracy of the basis set are the projectabilities $p_{nk} = \langle \psi_{nk} | P | \psi_{nk} \rangle \geq 0$ (P is the operator that projects onto the space of the PAO basis set, as defined in Ref. [87]), which indicate the representability of a Bloch state $|\psi_{nk}\rangle$ on the chosen PAO set. Maximum projectability, $p_{nk} = 1$, indicates that the Bloch state can be perfectly represented in the chosen PAO set; inversely, $p_{nk} \approx 0$ indicates that the PAO set is insufficient and should be augmented. Once the Bloch states with good projectabilities have been identified, the PAO Hamiltonian is constructed as

$$H(\mathbf{k}) = AEA^\dagger + \kappa \left[I - A(A^\dagger A)^{-1} A^\dagger \right],$$

as in Ref. [87]. Here E is the diagonal matrix of Kohn–Sham eigenenergies and A is the matrix of coefficients obtained from projecting the Bloch wavefunctions onto the PAO set [87]. The expression above can be understood as a filtering procedure on the PAO Hamiltonian: the states with good projectability are kept while all others are relegated to an orthogonal null space (the second term in the equation above). The null eigenvalues can be moved out from the physically relevant energy spectrum via a shifting parameter κ . An example of this is illustrated in Fig. 3(a), where the null space eigenvalues appear as a straight line at 27.5 eV. This procedure provides an accurate real space representation of the ab-initio Hamiltonian $H(\mathbf{r})$ as a tight-binding (TB) matrix of very small dimension, an advantage for the calculation of any physical property requiring precise integration in the reciprocal space.

By exploiting the PAO projection scheme described above, we can easily Fourier transform ($\mathcal{F}[\cdot]$) the PAO real space representation and interpolate to arbitrary precision; i.e., $H(\mathbf{r}_a) \rightarrow H(\mathbf{k})$. This procedure is computationally inexpensive because of the small dimension of the TB Hamiltonian, and it is performed using a zero-padding algorithm that operates globally on the PAO Hamiltonian with a Fourier transform. Simply from the knowledge of the PAO $H(\mathbf{k})$, one can directly construct the real-space localized Hamiltonian as

$$H(\mathbf{R}) = \mathcal{F}^{-1} [H(\mathbf{k})].$$

$H(\mathbf{R})$ is then zero-padded to i. increase the resolution in $\mathbf{k}$ -space upon inverse $\mathcal{F}[\cdot]$ and ii. obtain the interpolated PAO Hamiltonian for any

arbitrary $\mathbf{k}$ -vector mesh with the same accuracy defined by the projectability number.

From here, it is a simple task to evaluate the expectation value of the momentum operator — the main quantity in the definition of many property descriptors. The momentum operator is defined as [90]:

$$p_{nm}(\mathbf{k}) = \langle \psi_n(\mathbf{k}) | p | \psi_m(\mathbf{k}) \rangle = \left\langle u_n(\mathbf{k}) \left| \left(m_0 / \hbar \right) \nabla_{\mathbf{k}} H(\mathbf{k}) \right| u_m(\mathbf{k}) \right\rangle,$$

where

$$\nabla_{\mathbf{k}} H(\mathbf{k}) = \sum_{\mathbf{R}} \mathbf{R} \exp(i\mathbf{k} \cdot \mathbf{R}) H(\mathbf{R}),$$

$H(\mathbf{R})$ being the real space PAO matrix, and $|\psi_n(\mathbf{k})\rangle = \exp(-i\mathbf{k} \cdot \mathbf{r}) |u_n(\mathbf{k})\rangle$ the Bloch's functions [91]. This procedure can be applied multiple times to evaluate higher order derivatives (effective masses, etc.) [92].

In the original formulation of the PAOFLOW method, the atomic orbital basis was built from the radial pseudowavefunctions of the pseudopotential used in the DFT calculation. This “minimal basis set” approach has proved satisfactory in achieving accurate TB matrices for periodic systems. However, if more unoccupied bands are needed for a particular application, it can be achieved by progressively increasing the size of the atomic orbital basis set, effectively increasing the number of states with high projectability and thus the spectrum of $H(\mathbf{k})$.

We have recently developed an alternative approach that achieves this while maintaining the high accuracy of the minimal basis set. Our approach is entirely independent of the choice (or the availability) of the pseudopotential's radial functions. We generate the basis function by solving the all-electron atomic problem and building the basis set from the atomic radial functions, consistent with the valence states present in the pseudopotential [27]. In this way, we can increase the size of the basis set and construct Hamiltonians that reproduce exactly the electronic states for energies high in the conduction band. As an example, we show in Fig. 3(a) the band structure of silicon generated by a PAO Hamiltonian with a basis set comprising the $3s$, $3p$, $3d$, $4s$, $4p$, $4d$, and $4f$ orbitals. The accuracy of the representation, measured as the average difference between the original DFT bands and the PAOFLOW ones across the whole first Brillouin zone, is of the order of 10^{-3} for energies up to 20 eV. Moreover, the introduction of an explicit basis set promotes the PAOFLOW method beyond a simple tight-binding representation. We are now able to reconstruct the true electronic wavefunctions fully and thus the electronic density of the system (see Fig. 3(b)) — the essential quantity to evaluate a plethora of properties in their real space representation.

PAOFlow can be installed with `pip: pip install paoflow`. A set of examples and tutorials are provided through the PAOFlow website: <https://aflow.org/src/paoflow>. More technical details can be found in Refs. [26,27].

6. Thermodynamics

The energetics of the formation and decomposition reactions of a material influence whether it can be made and its stability. Ideally, the full reaction pathway should be considered (kinetics), including intermediate products which may be energetically inaccessible, rendering the reaction pathway unfeasible. These considerations are highly dependent on the experimental method and conditions, and are thus difficult to generalize. Instead, formation and stability analyses first focus only on the energy difference between the reaction endpoints (thermodynamics), which assumes the system can freely explore all possible outcomes to realize the minimum energy one (ergodicity).

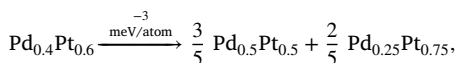
6.1. aflow-CHULL: The convex hull module

The aflow++ Convex Hull Module (aflow-CHULL) [32] can be used to construct ground-state ($T = 0$) ab-initio phase diagrams offering a wealth of thermodynamic data. Identification of stable phases involves the calculation of the convex hull in the free-energy-concentration space. Compounds having different structures (e.g., rock-salt vs. wurtzite) and concentrations (A_xB_{1-x}) are sampled from the aflow.org repositories. These configurations have been generated through a combination of structure prototyping of naturally occurring compounds [15–17] and structure enumeration algorithms [14]. Stability ($\Delta G < 0$) is achieved by minimizing the enthalpy and, in multi-phase regions, can involve changing concentrations (phase-separation and tie-line construction), as dictated by the minimum energy surface. The collection of stable phases and the tie-lines connecting them is determined by the convex hull: the set of outer-most points outlining the smallest convex shape enclosing the data, as illustrated in Fig. 4(a). The convex hull defines stability and thermodynamic pathways to it, and its construction is based solely on the geometry (positions) of the data. The energy axis is the formation enthalpy (H_f), which is defined as:

$$H_{f,A_xB_{1-x}} = H_{A_xB_{1-x}} - [xH_A + (1-x)H_B],$$

where $H_{A_xB_{1-x}}$ is the enthalpy of a binary compound, and H_A is the enthalpy of the A -reference-state. This compositionally-weighted shift of the raw DFT-enthalpies fixes the reference-states ($x = [0, 1]$) to zero, so only the lower convex hull needs to be calculated.

Compounds below the zero-formation-enthalpy tie-line ($H_f < 0$) are only stable with respect to decomposition to the reference states. The phases on the convex hull are globally stable (will not decompose) and thus are expected to form under similar experimental conditions [93–101]. Compounds above the hull will decompose into a linear combination of the stable phases defining the tie-line directly below it. For example, the decomposition reaction of Pd_2Pt_3 (highlighted in Fig. 4(a)) is:



where the distance to the hull (H_{hull}) is the energy difference between the products and the reactants, and the products define the tie-line directly below Pd_2Pt_3 . H_{hull} can be used as a measure of metastability: compounds close to the hull may stabilize at finite temperatures (room temperature: ~ 25 meV).

Analogous to the distance to the hull, the stability criterion [4,32] (ΔH_{sc}) was devised to gauge the relative stability of compounds on the hull. This descriptor is defined as the distance of the compound from the pseudo-hull constructed without it (Fig. 4(b)), quantifying the effect of the phase on the convex hull and its susceptibility to destabilization

by a new phase that has yet to be explored. The descriptor helped guide the design of two Heusler magnetic compounds, the first magnets discovered by computational approaches [4].

aflow-CHULL can construct and analyze convex hulls for arbitrary numbers of components: e.g., Ref. [8] presents distances to the hull and decomposition reactions for 5-metal high-entropy carbides (6D hulls). The module offers illustrations of the hull for binary and ternary systems (Fig. 4(c–d)), as well as a PDF report summarizing the data used to construct the hull and the results of the analysis (Fig. 4(e)). Entries are organized by arity (ternaries first, then binaries) and concentration. The report presents “unstable” vs. “ground-state” designations, distances to the hull, decomposition reactions, and stability criteria. For each stable phase, the report also provides the set of phases with which it is in equilibrium, i.e., the set of vertices for all the facets the stable phase defines. For example, Ag_2AuCd in Fig. 4(c) (lower left) is a vertex for eight facets, which are enumerated in Fig. 4(d) under “vertex of facets”. This information was used to discover two cobalt-based superalloys, where candidate impurity-phase compositions potentially form during age-hardening were screened for not being in two-phase equilibrium with the fcc host matrix [6,7].

The $N + 1$ enthalpy gain ($\Delta H[N|1, \dots, N - 1]$) has also been implemented within aflow-CHULL [102]. The descriptor quantifies an N -compound’s distance from the hull constructed only of $\{1, \dots, N - 1\}$ -compounds (Fig. 5), where, e.g., binaries are 2-compounds and ternaries are 3-compounds. The $N + 1$ enthalpy gain for 1-compounds is the cohesive energy, and for 2-compounds is the formation enthalpy (Fig. 5(a)). An analysis of the aflow.org repositories for metal compositions reveals that with an increasing number of species there is a diminishing enthalpy gain, which can be rapidly overcome by the configurational entropy gain ($N \geq 4$), see Fig. 5(c). This validates the unavoidability of disorder in multi-component systems [102].

Command-line options. There are several ways to interact with the aflow-CHULL module, including through the web [1], with the aflow++ binary installed locally, and through Python/Jupyter wrappers (calling a local install of the aflow++ binary). The binary offers full access to functionality and options, as well as various output formats, including plain text, JSON, PDF, and Jupyter notebooks. The PDF output requires the $\text{\LaTeX}$ package. See the Supporting Information of Ref. [32] for version and package details. The primary aflow-CHULL command

```
aflow --chull --alloy=MnPdPt
```

queries the aflow.org repositories for entries containing {Mn, Pd, Pt}, calculates the convex hull, and returns the information as a PDF (default, see --print). The flags and options include:

- `--chull`: Necessary flag for entering the convex-hull module.
- `--alloy=alloy`: Necessary argument, specifies the system. This code is not dimension limited, i.e., any N -ary system can be calculated. There are two input modes: raw (comma-separated) and combinatorial (colon- and comma-separated).
Raw input: `--alloy=MnPdPt,AlCuZn`.
Combinatorial input: `--alloy=Ag,Au:Mn`. This is interpreted as `--alloy=AgMn,AuMn`.
- `--np=npus`: Number of threads for calculation of multiple convex hulls. Default is `--np=1` (serial).
- `--print=format`: Selects the output format, options include: pdf, png, json, txt, jupyter2, and jupyter3. json and txt have the following extensions: .json and .txt. jupyter2/jupyter3 create a Python2/Python3 Jupyter notebook JSON file that plots a convex hull for the specified alloy. Default is pdf.
- `--dist2hull=aflow:bb0d45ab555bc208`: Returns the distance from the hull for entry aflow:bb0d45ab555bc208, specified by the AUID.

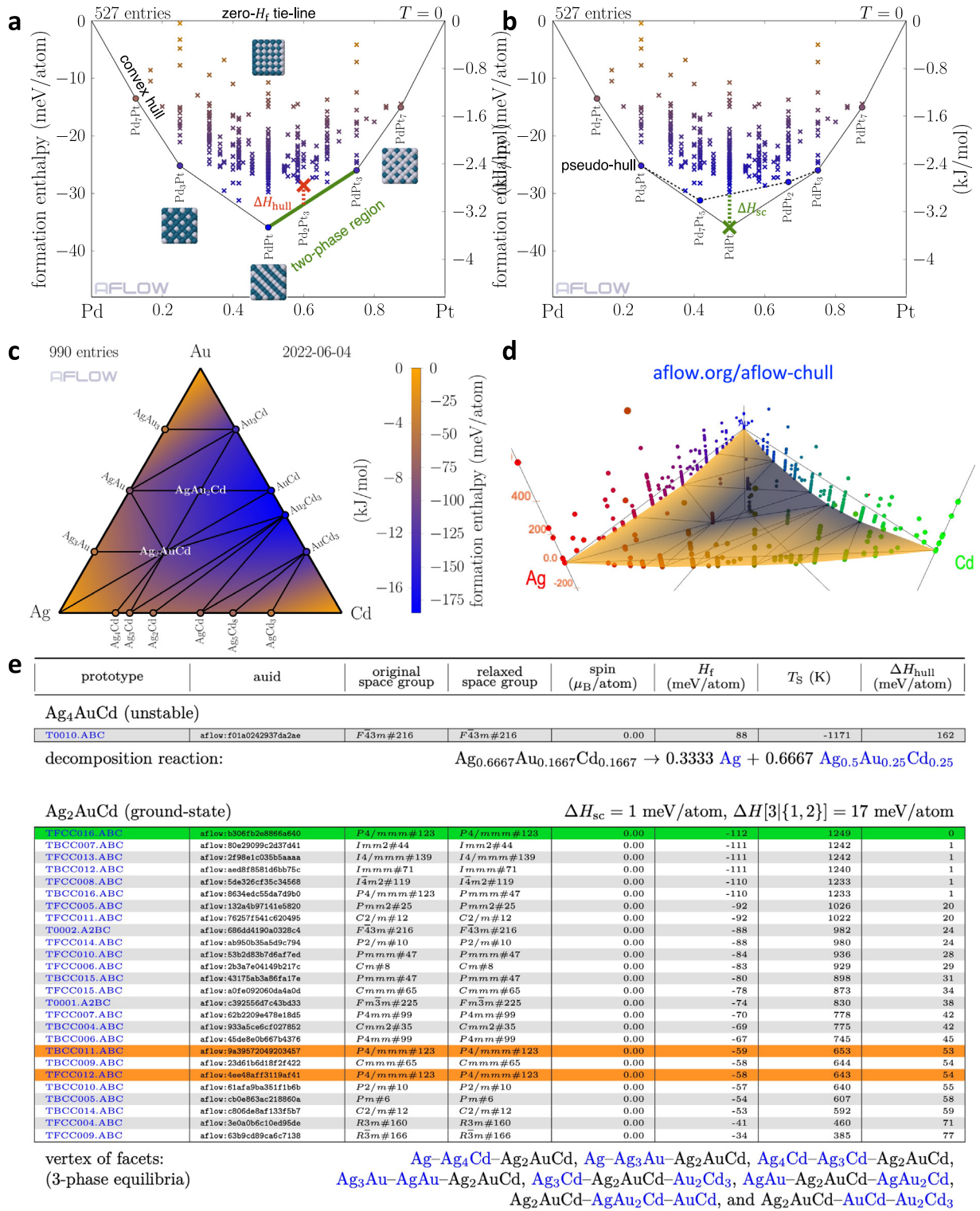


Fig. 4. Results and output from the aflow++ Convex Hull Module. (a) Illustration of the PdPt convex hull, highlighting the various compounds sampled from the aflow.org repositories having different structures and concentrations, the zero-formation-enthalpy tie-line, a two-phase region (hull facet), and the decomposition of Pd_2Pt_3 . (b) Illustration of the stability criterion analysis of PdPt, highlighting the construction of the pseudo-hull (dotted-line) from which the distance is measured. (c) Illustration of the AgAuCd convex hull, where the colors (orange to blue) indicate depth (lower formation enthalpy), as indicated by the color bar on the right. Only stable compounds (on the hull) are shown. (d) 3D rendering of the AgAuCd convex hull from the aflow-CHULL web application offering perspective on depth and coloring [1]. (e) An excerpt from the full AgAuCd PDF report, which organizes all the data used to construct the convex hull and presents the results of the analysis.

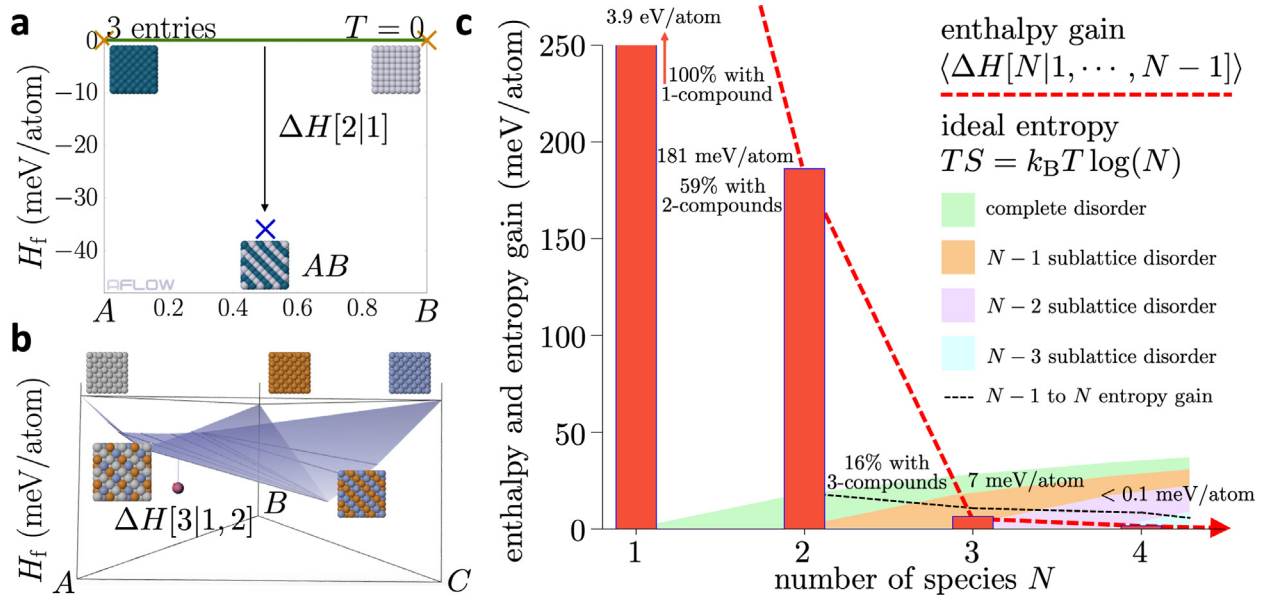


Fig. 5. The $N + 1$ enthalpy gain descriptor. (a) For binary compounds, the $N + 1$ enthalpy gain descriptor is the distance of the compound from the zero-formation-enthalpy tie-line (highlighted in green), which is trivially the formation enthalpy. (b) For ternary compounds, the $N + 1$ enthalpy gain descriptor is the distance of the compound from the convex hull constructed of unary (reference) and binary compounds (highlighted in blue). (c) A plot comparing the enthalpy gains averaged over N -compounds having metallic components in the *aflow.org* repositories with the configurational entropy gains for increasing numbers of species.

- `--scriterion=aflow:bb0d45ab555bc208`: Returns the stability criterion for entry `aflow:bb0d45ab555bc208`, specified by the AUID. The entry must be a ground-state structure; a warning will be issued otherwise. *aflow-CHULL* removes the point from the hull, calculates the pseudo-hull, and determines the distance of this point from below the pseudo-hull.
- `--nplus1=aflow:bb0d45ab555bc208`: Returns the $N + 1$ enthalpy gain for entry `aflow:bb0d45ab555bc208`, specified by the AUID. The entry must be a ground-state structure; a warning will be issued otherwise. *aflow-CHULL* removes all points having the same dimensionality as the input entry from the hull, calculates the pseudo-hull, and determines the distance of this point from below the pseudo-hull.
- `--hull_enthalpy=0.25,0.25`: Returns the value of the convex hull surface at the specified coordinate/concentration. Users should provide the composition in reduced form, e.g., the Mn_2PdPt composition is specified by `--hull_enthalpy=0.5,0.25`, where the last component is implicitly $1 - \text{sum}(0.5 + 0.25)$.

For the full set of options and additional information, see the *aflow-CHULL* README:

```
aflow --readme=chull
```

6.2. *aflow-CCE*: The coordination corrected enthalpies module

While there have been significant advances in calculating finite temperature effects from first principles [28,103,104], the computational modeling of formation enthalpies — the enthalpy difference between the material and its elemental references — still poses a fundamental challenge. Standard (semi-)local and even currently available advanced ab-initio approaches yield inaccurate predictions [105–112], with errors of several hundred meV/atom in particular for ionic systems, which inhibits materials design. The problem is intimately connected to the fact that computing reliable formation enthalpies ab-initio eventually requires accurate total energies for all systems involved [33,35,108]. This is generally not possible within a (semi-)local approximation. To date, it even remains unknown what level of DFT-based theory would be needed to achieve satisfactory accuracy for

formation enthalpies, given that exact Quantum Monte Carlo results are only available for a few special systems, such as MgH_2 [113,114].

Physically motivated empirical correction schemes parameterizing (semi-)local DFT errors with respect to measured values are hence the only feasible option to enable materials design. Several correction methods based solely on the composition of the materials were established [105–108,115]. These approaches were a major step forward, but their accuracy is limited and the relative stability of polymorphs — sometimes erroneously predicted by DFT [109] — cannot be corrected. Moreover, correction methods based only on composition can lead to incorrect thermodynamic behavior when considering activity vs. concentration [35].

The coordination corrected enthalpies method. The method of coordination corrected enthalpies (CCE) [35] presents a solution to this problem, improving on the established approaches both qualitatively and quantitatively. CCE is the first scheme to leverage structural information. Incorrect thermodynamic behavior is avoided by construction. The method is based on the following physical insight: bonding is central to capturing the properties of a material, with DFT revealing specific errors for each bonding type. Thus, the number of bonds in a material is a suitable descriptor to parameterize DFT errors. The correction is hence developed per bond and per oxidation state. The latter dependence ensures optimal transferability of the corrections since the energetic position of valence states is usually well characterized by the oxidation state. The idea is further illustrated in the example of Figs. 6(a–c): in CaO , the Ca cations are sixfold (octahedrally) coordinated by oxygen anions. Ti is also sixfold coordinated by oxygen in rutile TiO_2 . In perovskite CaTiO_3 , the coordination number for Ti stays the same, but the number of Ca–O bonds changes to eight. A variation in the number of bonds for a given cation between different materials is common and signifies that corrections to calculated DFT formation enthalpies should be obtained and applied per bond.

With binary compounds $A_{x_1}Y_{x_2}$ used as the fit set, the CCE corrections $\delta H_{A-Y}^{T_r, A+\alpha}$ per cation–anion A – Y bond and cation oxidation state $+\alpha$ are obtained from the difference between (zero-temperature and zero-pressure) DFT and experimental standard room temperature formation enthalpies [33,35]:

$$H_{f,A_{x_1}Y_{x_2}}^{0,\text{DFT}} - H_{f,A_{x_1}Y_{x_2}}^{o,T_r,\text{exp}} = x_1 N_{A-Y} \delta H_{A-Y}^{T_r, A+\alpha},$$

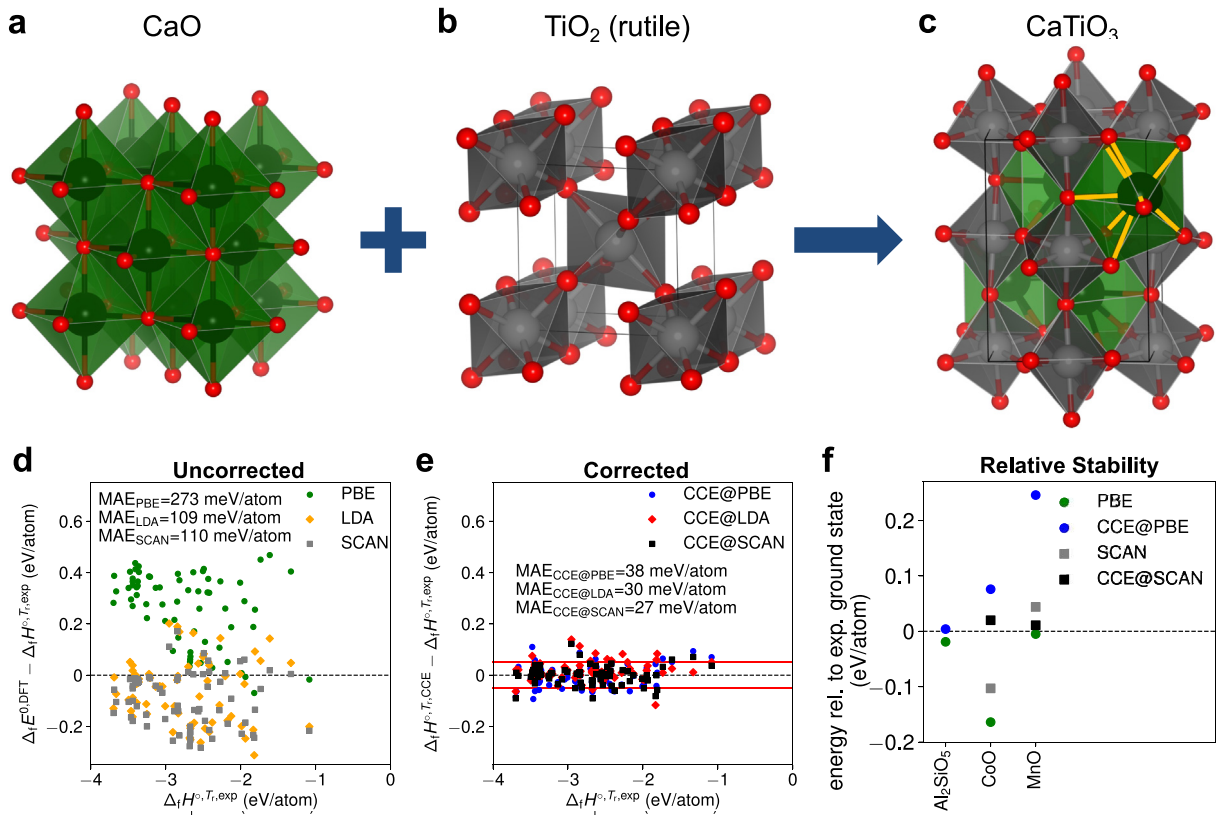


Fig. 6. Motivation and validation of CCE. Crystal structures of (a) CaO, (b) rutile TiO₂, and (c) perovskite CaTiO₃. While Ti is sixfold coordinated by O anions in both rutile and perovskite structures, Ca changes its coordination from sixfold in CaO to eightfold in CaTiO₃ (Ca–O bonds highlighted in yellow). This indicates an important change in the number of bonds critical for the thermodynamic stability of a material. Deviations between (d) calculated ($H_{f,A_{x_1}Y_{x_2}}^{0,DFT}$) and experimental ($H_{f,A_{x_1}Y_{x_2}}^{o,T_r,exp}$) as well as (e) corrected ($H_{f,A_{x_1}Y_{x_2}}^{o,T_r,CCE}$) and experimental formation enthalpies for 71 ternary oxides. Red lines at ± 50 meV/atom are visual guides representing the typical MAE of previous methods [107,108]. (f) Correction of the relative stability of Al₂SiO₅, CoO, and MnO polymorphs. Color code: Ca, black; Ti, light gray; O, red [116].

where $H_{f,A_{x_1}Y_{x_2}}^{0,DFT}$ is the DFT formation enthalpy, $H_{f,A_{x_1}Y_{x_2}}^{o,T_r,exp}$ is the measured standard formation enthalpy at the reference temperature $T_r = 298.15$ K, and N_{A-Y} is the number of nearest neighbor A–Y bonds of element A in oxidation state + α . Note that the DFT formation enthalpies strictly include only the internal energy contributions to the enthalpies and the small pressure–volume terms are neglected [35].

The corrections can then be applied to any multinary compound $A_{x_1}B_{x_2}\dots Y_{x_n}$ to obtain the CCE formation enthalpy $H_{f,A_{x_1}B_{x_2}\dots Y_{x_n}}^{o,T_r,CCE}$ at no additional computational cost compared to DFT:

$$H_{f,A_{x_1}B_{x_2}\dots Y_{x_n}}^{o,T_r,CCE} = H_{f,A_{x_1}B_{x_2}\dots Y_{x_n}}^{0,DFT} - \sum_{i=1}^{n-1} x_i N_{i-Y} \delta H_{i-Y}^{T_r,i+\alpha},$$

where N_{i-Y} is the number of nearest neighbor bonds between the cation i and anion Y species. Temperature effects can also be parameterized on a per bond basis such that, in addition to room temperature values, also 0 K formation enthalpies are computed [33].

The predictive power of CCE was validated on a test set of 71 ternary oxides. DFT yields, for PBE [46], LDA [117,118] and SCAN [119], mean absolute errors (MAEs) of the calculated enthalpies of at least 100 meV/atom (Fig. 6(d)). The CCE values show an improvement by a factor of 4–7 to 38, 30 and 27 meV/atom (Fig. 6(e)). These mean deviations are significantly smaller than the ones of 45 and 48 meV/atom predicted by previous methods [107,108]. The general applicability of CCE was benchmarked on a set of ternary halides, achieving the same accuracy [35]. CCE is also capable of correcting the relative stability of polymorphs at fixed composition — a qualitative advantage versus all earlier schemes — as demonstrated for several minerals and transition metal systems [35]. In Fig. 6(f) this is indicated in three examples. While plain PBE predicts the andalusite polymorph

of Al₂SiO₅ to be more stable than the experimentally known ground-state kyanite (green dot), the application of CCE shifts andalusite energetically above kyanite and retains the correct energetic ordering (blue dot). Similarly, for CoO and MnO, it predicts the correct ground-state rocksalt structure, while DFT erroneously yields zincblende with only four Co/Mn–O bonds [35]. Other correction methods based on only composition cannot rectify the relative stability. CCE also gives accurate defect energies evidenced from investigating crystallographic shear compounds [35].

afLOW-CCE. CCE is integrated into and is fully interoperable with existing afLOW++ functionality [33]. afLOW-CCE returns the correction and the corrected formation enthalpies for a given structure. It features a command-line tool, a web interface, and a Python environment [33]. As shown in Fig. 7(a), the workflow analyzes the structure for cation coordination numbers, automatically determines oxidation numbers based on Allen electronegativities [120–122], includes temperature effects, and calculates the corrected formation enthalpies for the selected functionals. The oxidation state determination includes the treatment of mixed-valence systems, such as Ti–O Magnéli phases and alkali-metal sesquioxides. The software is also readily applicable to (su-)peroxides and multi-anion systems. The algorithms of the implementation are detailed in Ref. [33].

The command-line tool (Figs. 7(b–g)) delivers the CCE cation coordination numbers, oxidation numbers, and corrections, as well as formation enthalpies for the given structure file that can be in any format recognizable by afLOW++, like VASP POSCAR [52], QUANTUM ESPRESSO [53], FHI-AIMS [21], ABINIT [20], ELK [22] and CIF [54]. For VASP, a VASP5 POSCAR is required, or if a VASP4 POSCAR is used, the species must be written on the right side next to the coordinates for each atom just as for the example input structure obtained from the

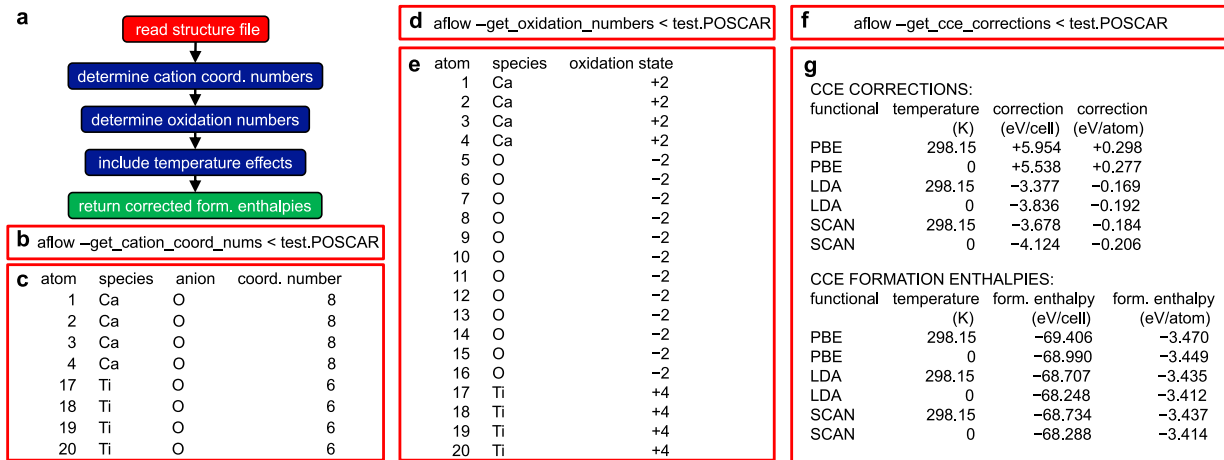


Fig. 7. aflow-CCE implementation. (a) Workflow of the CCE implementation. (b and d) Example commands for the aflow-CCE command-line tool using the input structure file `test.POSCAR` (perovskite CaTiO_3) to determine cation coordination numbers and oxidation states. (c and e) When executed, the output is presented in tabular form. (f) Example command for determining CCE corrections and formation enthalpies. Several other options to use extended functionality are also summarized in the main text [33]. (g) When executed, the output includes the CCE corrections and formation enthalpies at both 298.15 and 0 K for various functionals.

option `--cce`. If desired, oxidation numbers for all atoms can also be provided upon input. The available options include:

```
aflow --cce
```

prints instructions and an example input structure.

```
aflow --cce=GEOM_FILE
```

prints the results of the full CCE analysis, i.e., cation coordination numbers, oxidation numbers, and CCE corrections and formation enthalpies for the given structure.

```
aflow --get_cce_corrections < GEOM_FILE
```

determines the CCE corrections and formation enthalpies for the structure.

```
aflow --get_oxidation_numbers < GEOM_FILE
```

determines the oxidation numbers for all atoms of the structure.

```
aflow --get_cation_coord_nums < GEOM_FILE
```

determines the number of anion neighbors for each cation of the structure.

Options for `--cce=GEOM_FILE` and `--get_cce_corrections < GEOM_FILE`:

- `--enthalpies_formation_dft=enthalpies`: *enthalpies* is a comma-separated list of precalculated DFT formation enthalpies. They are assumed to be negative for compounds lower in enthalpy than the elemental references and in eV/cell. Currently, corrections are available for PBE, LDA, and SCAN.
- `--functionals=functionals`: *functionals* is a comma-separated list of functionals for which corrections should be returned. If used together with `--enthalpies_formation_dft`, the functionals must be in the same sequence as the corresponding formation enthalpies. Available functionals are PBE, LDA, and SCAN. Default is PBE (if only one DFT formation enthalpy is provided).
- `--oxidation_numbers=oxidation_numbers`: *oxidation_numbers* is a comma-separated list of oxidation numbers. It is assumed that one is provided for each atom of the structure and they are in the same sequence as the corresponding atoms in the provided structure file.

The output of these results can be returned as plain text or a JSON format: `--print=format`, where *format* can be *txt* or *json*, respectively. For the full set of options and additional information, see the aflow-CCE README:

```
aflow --readme=cce
```

The aflow-CCE implementation enables the enthalpy correction of an extensive library of materials as well as the accurate and quick generation of convex hull phase diagrams [32]. The tool is also readily applicable to reduced-dimensionality, e.g., 2D systems. aflow-CCE thus provides a valuable infrastructure for the scientific community to obtain the CCE corrections and corrected formation enthalpies for a given input structure. It can also expedite various materials design applications, such as the discovery of novel 2D systems and high-entropy ceramics.

7. Thermomechanical analysis

7.1. aflow-AEL-AGL: The automatic elasticity and GIBBS libraries

AEL: Elastic constants. Thermomechanical properties of materials, such as elastic moduli, Debye temperature, heat capacity and thermal expansion, can be obtained using the aflow++ Automatic Elasticity Library (aflow++-AEL [29]) and the aflow++ Automatic GIBBS Library (aflow++-AGL [30]) modules based on calculations of strained primitive cells. These methods are generally computationally less costly than the APL and AAPL phonon calculations. However, APL and AAPL generally give more quantitatively accurate results, particularly for properties where anharmonic effects are important. AEL and AGL have been combined into a single automated workflow, which has been used to calculate the thermomechanical properties of over 6000 materials in the aflow.org repositories.

The AEL module applies a set of independent normal and shear strains to the primitive cell of a material [29,49] as depicted in Fig. 8(a), and uses DFT to calculate the resulting stress tensors. This set of strain–stress data is used to generate the elastic stiffness tensor, i.e., the elastic constants:

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{12} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ c_{13} & c_{23} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{14} & c_{24} & c_{34} & c_{44} & c_{45} & c_{46} \\ c_{15} & c_{25} & c_{35} & c_{45} & c_{55} & c_{56} \\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & c_{66} \end{pmatrix} \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{pmatrix}$$

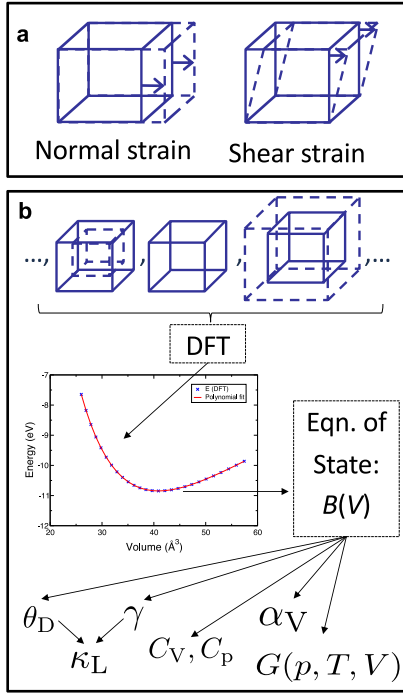


Fig. 8. Calculations of thermomechanical properties in aflow++. (a) AEL uses the stresses from a set of normal and shear strained cells to obtain the elastic constants. (b) AGL calculates the energies of a set of isotropically compressed and expanded unit cells, and fits the resulting $E(V)$ data by a numerical polynomial or by an empirical equation of state to obtain the bulk modulus, and hence other thermal and mechanical properties.

written in the 6×6 Voigt notation using the mapping [123]: $11 \mapsto 1$, $22 \mapsto 2$, $33 \mapsto 3$, $23 \mapsto 4$, $13 \mapsto 5$, $12 \mapsto 6$. The elastic constants are combined to calculate the bulk (B) and shear (G) elastic moduli in the Voigt approximation, which assumes a uniform strain with the stress supported by the individual grains in parallel, giving the upper bound on the elastic moduli:

$$B_{\text{Voigt}} = \frac{1}{9} [(c_{11} + c_{22} + c_{33}) + 2(c_{12} + c_{23} + c_{13})],$$

$$G_{\text{Voigt}} = \frac{1}{15} [(c_{11} + c_{22} + c_{33}) - (c_{12} + c_{23} + c_{13}) + 3(c_{44} + c_{55} + c_{66})];$$

the Reuss approximation, which assumes a uniform stress so that the strain is the sum of the strains of the individual grains in series, giving the lower bound on the elastic moduli:

$$B_{\text{Reuss}}^{-1} = (s_{11} + s_{22} + s_{33}) + 2(s_{12} + s_{23} + s_{13}),$$

$$G_{\text{Reuss}}^{-1} = \frac{1}{15} [4(s_{11} + s_{22} + s_{33}) - 4(s_{12} + s_{23} + s_{13}) + 3(s_{44} + s_{55} + s_{66})];$$

and Voigt–Reuss–Hill (VRH, B_{VRH} and G_{VRH} [124]) average:

$$B_{\text{VRH}} = (B_{\text{Voigt}} + B_{\text{Reuss}}) / 2,$$

$$G_{\text{VRH}} = (G_{\text{Voigt}} + G_{\text{Reuss}}) / 2.$$

The Poisson ratio ν is given by:

$$\nu = \frac{3B_{\text{VRH}} - 2G_{\text{VRH}}}{6B_{\text{VRH}} + 2G_{\text{VRH}}}. \quad (1)$$

Performing AEL calculations. The settings for AEL calculations are provided in the aflow.in file. To run AEL, the line [AFLOW_AEL]CALC needs to be present and uncommented — an appropriate aflow.in can be generated by including the --module=aelflag to the file generation command (--aflow_proto). AEL can reduce the number of required strains by setting [AFLOW_AEL]STRAIN_SYMMETRY=ON. The number and size of the strains in each independent direction can be controlled using the parameters:

```
[AFLOW_AEL]NNORMAL_STRAINS=value
[AFLOW_AEL]NSHEAR_STRAINS=value
[AFLOW_AEL]NORMAL_STRAIN_STEP=value
[AFLOW_AEL]NORMAL_STRAIN_STEP=value
```

A full list of parameters is available in the AEL README, which can be printed using the command

```
aflow --readme=aelf
```

AGL: Debye–Grüneisen model. The AGL module is based on the GIBBS [125,126] quasi-harmonic Debye–Grüneisen method, and calculates the energy as a function of volume $E(V)$ for a set of isotropically compressed and expanded strains of the primitive cell, as illustrated in Fig. 8(b). The $E(V)$ data are fitted by either a numerical polynomial or an empirical equation of state to obtain the adiabatic bulk modulus $B_S(V)$. The Debye temperature $\theta_D(V)$ as a function of volume is then calculated using the expression:

$$\theta_D = \frac{\hbar}{k_B} [6\pi^2 V^{1/2} n]^{1/3} f(\nu) \sqrt{\frac{B_S}{M}}, \quad (2)$$

where n is the number of atoms per unit cell, M is the unit cell mass, and $f(\nu)$ is a function of the Poisson ratio ν :

$$f(\nu) = \left\{ 3 \left[2 \left(\frac{2}{3} \cdot \frac{1+\nu}{1-2\nu} \right)^{3/2} + \left(\frac{1}{3} \cdot \frac{1+\nu}{1-2\nu} \right)^{3/2} \right]^{-1} \right\}^{1/3},$$

where ν can be obtained from Eq. (1) using AEL, or set directly by the user ([AFLOW_AGL]POISSON_RATIO=value). The vibrational contribution to the free energy, F_{vib} , is given by:

$$F_{\text{vib}}(\theta_D; T) = nk_B T \left[\frac{9}{8} \frac{\theta_D}{T} + 3 \log(1 - e^{-\theta_D/T}) - D \left(\frac{\theta_D}{T} \right) \right],$$

where $D(\theta_D/T)$ is the Debye integral:

$$D(\theta_D/T) = 3 \left(\frac{T}{\theta_D} \right)^3 \int_0^{\theta_D/T} \frac{x^3}{e^x - 1} dx.$$

The Gibbs free energy is obtained from:

$$G(V; p, T) = E_{\text{DFT}}(V) + F_{\text{vib}}(\theta_D(V); T) + pV.$$

The volume which minimizes $G(V; p, T)$ at a given pressure p and temperature T is the equilibrium volume V_{eq} , which is used to evaluate $\theta_D(V_{\text{eq}})$ and the Grüneisen parameter γ , as defined by:

$$\gamma = - \frac{\partial \log(\theta_D(V))}{\partial \log V}.$$

Finally, θ_D and γ are used to calculate other thermal properties [30, 125] including specific heat capacity at constant volume C_V :

$$C_V = 3nk_B \left[4D \left(\frac{\theta_D}{T} \right) - \frac{3\theta_D/T}{\exp(\theta_D/T) - 1} \right];$$

volumetric thermal expansion α_V :

$$\alpha_V = \frac{\gamma C_V}{B_T V},$$

where B_T is the isothermal bulk modulus; specific heat capacity at constant pressure C_p :

$$C_p = C_V(1 + \gamma \alpha_V T);$$

and lattice thermal conductivity κ_L [127–129]:

$$\kappa_L(\theta_a) = \frac{0.849 \times 3^{3/4}}{20\pi^3(1 - 0.514\gamma_a^{-1} + 0.228\gamma_a^{-2})} \times \left(\frac{k_B \theta_a}{\hbar} \right)^2 \frac{k_B m V^{1/3}}{\hbar \gamma_a^2},$$

where θ_a and γ_a is the Debye temperature and Grüneisen parameter calculated from only the acoustic modes.

Performing AGL calculations. The settings for AGL calculations are provided in the aflow.in file. To run AGL, the line [AFLOW_AGL]CALC needs to be present and uncommented — an appropriate aflow.in

in can be generated by including the `--module=agl` flag to the file generation command (`--aflow_proto`). AGL can run AEL to obtain the Poisson ratio for use in Eq. (2) to calculate the Debye temperature by setting `[AFLOW_AGL] AEL_POISSON_RATIO=ON`. The number of different volumes and the size of the strain steps can be controlled using the parameters:

```
[AFLOW_AGL] NSTRUCTURES=value
[AFLOW_AGL] STRAIN_STEP=value
```

The number of temperature and pressure points and the corresponding step sizes for the post-processing can be controlled using the parameters:

```
[AFLOW_AGL] NTEMP=value
[AFLOW_AGL] STEMP=value
[AFLOW_AGL] NPRESSURE=value
[AFLOW_AGL] SPRESSURE=value
```

Note that the post-processing can be run multiple times with different temperature and pressure ranges for the same set of DFT calculations. A full list of parameters is available in the AGL README, which can be printed using the command:

```
aflow --readme=agl
```

7.2. aflow-APL: The automatic phonon library

Phonons in the harmonic approximation. Phonons are the basis for many finite-temperature processes in solids. They can contribute to the stabilization of a material, determine thermophysical properties such as heat capacities and thermal expansion, and are responsible for transport phenomena such as thermal conductivity. The `aflow++` Automatic Phonon Library (APL) calculates phonon modes in the harmonic approximation by diagonalizing the dynamical matrix $D(\mathbf{q})$ [34]:

$$D(\mathbf{q}) \mathbf{e}_\lambda = \omega_\lambda^2 \mathbf{e}_\lambda,$$

where the phonon mode $\lambda = \{\mathbf{q}, j\}$ is a combined index consisting of the reciprocal space point $\mathbf{q}$ and the branch index j . ω_λ and $\mathbf{e}_\lambda$ are the frequency and the eigenvector of the mode, respectively. The components of the dynamical matrix are [130]:

$$D_{\alpha\beta}(\kappa\kappa'|\mathbf{q}) = \frac{1}{\sqrt{m_\kappa m_{\kappa'}}} \sum_{ll'} \Phi_{\alpha\beta}(l\kappa; l'\kappa') \times \exp[i\mathbf{q} \cdot (\mathbf{R}_{l'} - \mathbf{R}_l)], \quad (3)$$

with the Cartesian indices α and β , the atomic indices κ and κ' , and the supercell indices l and l' . m_κ is the mass of atom κ and $\mathbf{R}_l$ is the vector connecting the origin of the crystal to the origin of supercell l . $\Phi_{\alpha\beta}(l\kappa; l'\kappa')$ are the harmonic interatomic force constants (IFCs), the calculation of which is the central problem for phonon calculations. APL can obtain them either from Γ -point density functional perturbation theory or through the finite displacement method.

The finite displacement method applies small distortions to the atomic positions inside a supercell and calculates the forces using DFT. For a full set of IFCs, each symmetrically-inequivalent atom needs to be displaced along three linearly-independent directions. To minimize the number of calculations, APL uses the following algorithm:

- Create test displacements along the unit cell axes, face diagonals, and body diagonal.
- Generate a set of displacement vectors that are orthogonal and symmetrically equivalent to the test displacement using the site point group of the atom [31] and Gram–Schmidt orthogonalization.
- Sort these sets of displacements by the number of equivalent vectors from highest to lowest.

- Take the displacements inside the first set of this sorted list. If there are less than three, add the displacements in the next set of the list and use the Gram–Schmidt method and the site point groups to create orthogonal vectors. Repeat until three linearly-independent directions are found.

This algorithm not only reduces computational requirements by minimizing the number of calculations, using site point groups to generate the displacements also leads to supercells with the highest possible symmetry. The calculated forces are then used to determine the IFCs through finite differences. These “raw” force constants do not generally fulfill the acoustic sum rule $\sum_{\kappa'} \Phi_{\alpha\beta}(l\kappa; l'\kappa') = 0$ nor are they commensurate with the site point groups of the atoms in the crystal. APL automatically enforces these properties when calculating the IFCs. The dynamical matrix can then be constructed to solve Eq. (3).

The resulting frequencies are used to obtain phonon dispersions and DOS. The DOS $g(\omega)$ can be used to calculate the vibrational free energy F_{vib} , internal energy U_{vib} , and entropy S_{vib} , as well as the isochoric heat capacity C_V :

$$F_{\text{vib}}(T) = k_B T \int_0^\infty \log \left(2 \sinh \frac{\hbar\omega}{2k_B T} \right) g(\omega) d\omega,$$

$$U_{\text{vib}}(T) = \int_0^\infty \frac{\hbar\omega}{2} \coth \left(\frac{\hbar\omega}{2k_B T} \right) g(\omega) d\omega,$$

$$S_{\text{vib}}(T) = \frac{U_{\text{vib}} - F_{\text{vib}}}{T},$$

$$C_V(T) = k_B \int_0^\infty \left(\frac{\hbar\omega}{2k_B T} \right) \text{csch}^2 \frac{\hbar\omega}{2k_B T} g(\omega) d\omega,$$

where k_B and $\hbar$ are the Boltzmann and the reduced Planck constant, respectively. Additionally, APL can calculate group velocities $\mathbf{v}_\lambda$ and mean square atomic displacements $\langle |u|^2 \rangle$ using the eigenvectors:

$$\mathbf{v}_\lambda = \left\langle \mathbf{e}_\lambda \left| \frac{\partial D(\mathbf{q})}{\partial \mathbf{q}} \right| \mathbf{e}_\lambda \right\rangle,$$

$$\langle |u^\alpha(\kappa, T)|^2 \rangle = \frac{\hbar}{N_q m_j} \sum_\lambda \omega_\lambda^{-1} \left(\frac{1}{2} + n_\lambda \right) |\mathbf{e}_\lambda^\alpha(\kappa)|^2,$$

with N_q and n_λ being the number of $\mathbf{q}$ -points and the phonon numbers based on the Bose–Einstein distribution, respectively.

Long-range Coulomb interactions in polar materials cause splitting between the longitudinal and transversal optical phonon modes (LO-TO splitting). This requires a corrective term $\tilde{D}$ to be added to the dynamical matrix. `aflow++` uses the method by Wang et al. to calculate this correction [131]:

$$\tilde{D}_{\alpha\beta}^{\kappa\kappa'} = \frac{4\pi e}{V \sqrt{m_\kappa m_{\kappa'}}} \frac{[\mathbf{q}Z^*(i)]_\alpha [\mathbf{q}Z^*(i)]_\beta}{\mathbf{q} \cdot \boldsymbol{\epsilon}_\infty \mathbf{q}} \times \sum_{ll'} \exp[i\mathbf{q} \cdot (\mathbf{R}_{l'} - \mathbf{R}_l)],$$

where e , Z^* , and $\boldsymbol{\epsilon}_\infty$ are the elemental charge, the Born effective charge tensor, and the dielectric tensor, respectively. The tensors can be directly calculated by VASP.

Performing APL calculations. The parameters for phonon calculations are set in the `aflow.in` file. A full list of parameters is available in the APL README, which can be created using the command:

```
aflow --readme=apl
```

To run APL, the line `[AFLOW_APL] CALC` needs to be present. This can be achieved programmatically by adding `--module=apl` to the input file generation command (`--aflow_proto`). The flowchart of the APL workflow is shown in Fig. 9(a).

Since phonon properties require accurate forces, the structure needs to be relaxed so that the forces are near zero. This is done automatically by APL at the beginning of the workflow. The following parameters guide the relaxation calculations:

```
[AFLOW_APL] RELAX=ON
[AFLOW_APL] RELAX_COMMENSURATE=ON
[AFLOW_APL] KPPRA= $N_k$ 
[AFLOW_APL] KPOINTS_GRID= $k_1 \times k_2 \times k_3$ 
```

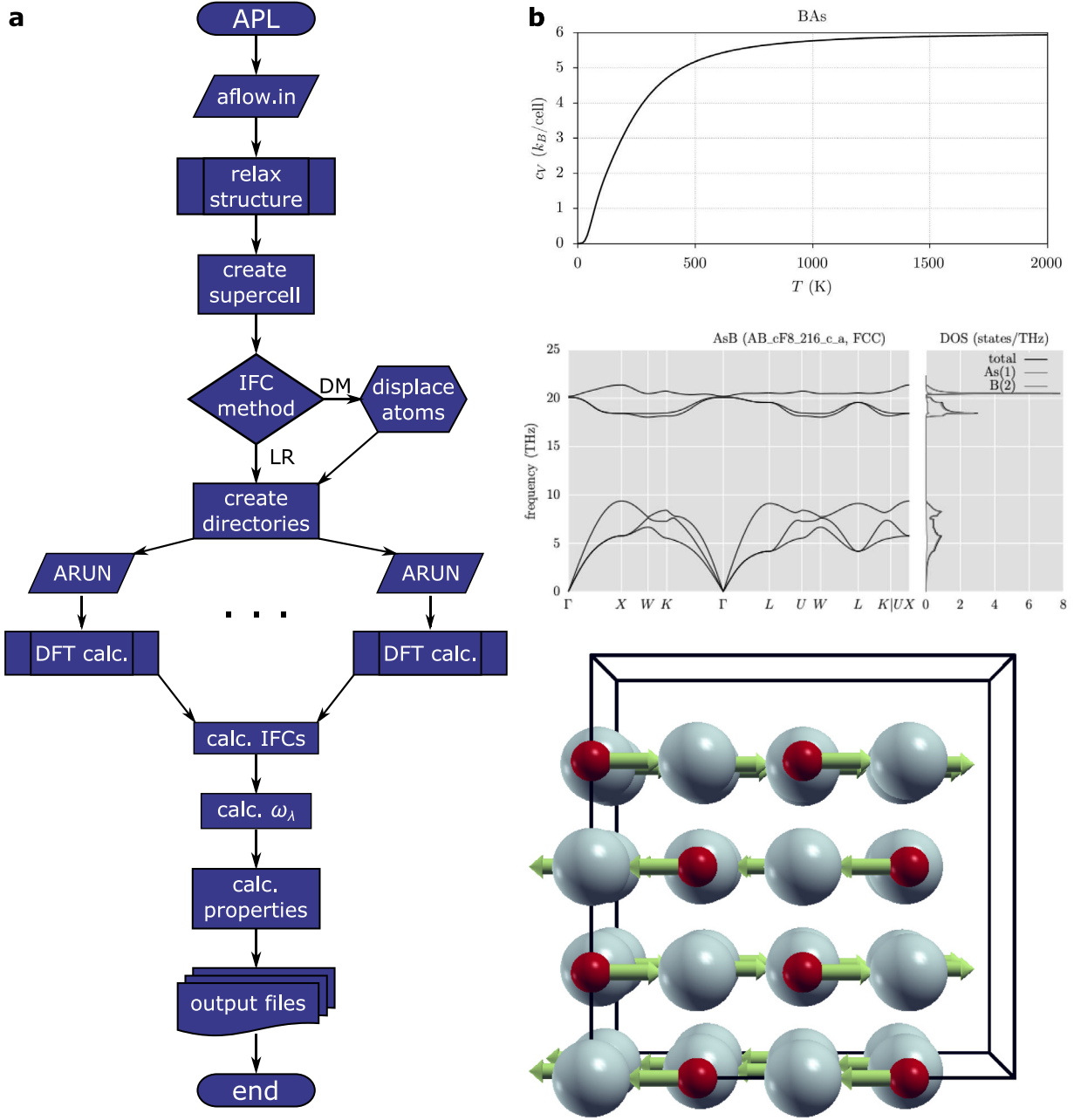


Fig. 9. Workflow and outputs of the Automatic Phonon Library (APL) (a) Flowchart of the APL workflow. (b) Isochoric heat capacity C_V of zincblende BAs. (c) Combined phonon dispersion and projected DOS plot for BAs. (d) Snapshot of a phonon mode visualization at the X point for rocksalt MgO.

`[AFLOW_APL]RELAX_COMMENSURATE=ON` ensures that the $\mathbf{k}$ -point grids of the supercell for the force calculations and the grid of the relaxation calculations are commensurate, which increases the accuracy of the obtained forces. It can be toggled off with `[AFLOW_APL]RELAX_COMMENSURATE=OFF`. To determine the grid size, `[AFLOW_APL]KPPRA` or `[AFLOW_APL]KPOINTS_GRID` are used, which represent the $\mathbf{k}$ -points per reciprocal atom N_k or the $\mathbf{k}$ -point grid dimensions, respectively, for the supercell. Convergence criteria are set such that the forces between two successive ionic steps are below 10^{-3} eV/Å and energy differences are below 10^{-8} eV between electronic steps. The relaxation can be skipped entirely by setting `[AFLOW_APL]RELAX=OFF`.

After the relaxation, APL builds the supercell using either explicit dimensions, a minimum number of atoms, or a number of coordination shells around each atom:

```

[AFLOW_APL]SUPERCELL=3x3x3
[AFLOW_APL]MINATOMS=175
[AFLOW_APL]MINSHELL=10
  
```

SUPERCELL takes priority, followed by MINATOMS. APL then creates subdirectories for the force calculations using DFT. This process is governed by the following `aflow.in` parameters:

```

[AFLOW_APL]ENGINE=method
[AFLOW_APL]DPM=ON
[AFLOW_APL]DMAG= $r_{\text{displ}}$ 
[AFLOW_APL]ZEROSTATE=OFF
[AFLOW_APL]POLAR=ON
  
```


[AFLOW_APL]ENGINE determines the method to calculate the force constants. It can be LR or DM, corresponding to the linear response (Γ -point density functional perturbation theory) and the direct method (finite displacement), respectively. The former generates one subdirectory as the force constants are directly calculated by VASP.

The finite displacement method provides two ways to determine the IFCs: the central and the forward difference methods. [AFLOW_APL]DPM=ON always uses the central difference method, [AFLOW_APL]DPM=OFF always uses forward differences, and [AFLOW_APL]DPM=AUTO determines for each site whether the forward difference is permitted by symmetry. The size of the displacement r_{displ} (in Ångström) is set by DMAG. Additionally, [AFLOW_APL]ZEROSTATE can be used to calculate the forces on the atom in the unperturbed supercell, which can be used to subtract noise for the forward difference method or to test whether virtual forces are present in the supercells. For both methods, [AFLOW_APL]POLAR=ON applies the non-analytical term correction, resulting in an additional subdirectory to calculate Z^* and ϵ_0 .

Each subdirectory contains an aflow.in file for a static DFT calculation and needs to be run separately. These calculations have a convergence criterion of 10^{-8} eV between electronic steps and use the k -point parameters described earlier. Once finished, aflow++ needs to be run in the parent directory again to read the forces, determine the IFCs, and calculate phonon dispersions, DOS, and thermophysical properties. Important settings for this post-processing step include:

```
[AFLOW_APL]DOS_PROJECT=OFF
[AFLOW_APL]TPT= $T_{\text{start}}:T_{\text{end}}:T_{\text{step}}$ 
```

where [AFLOW_APL]DOS_PROJECT determines whether atom-projected DOS are calculated and [AFLOW_APL]TPT sets the start and end temperatures and the temperature step size for thermophysical properties and atomic displacements.

Visualization options. APL provides several output options for its calculations. Thermophysical properties can be plotted and saved as an image by calling:

```
aflow --plotthermo
```

where --plotthermo accepts multiple optional inputs: the *directory* of the phonon calculation (default: current directory), and the minimum (T_{min}) and maximum (T_{max}) temperatures (default: full range of calculated temperatures); i.e., --plotthermo=*directory*, T_{min} , T_{max} . An example plot for the isochoric heat capacity is presented in Fig. 9(b). Moreover, the following additional options may be specified:

- --title=*title* : Title of the plot. If not used, aflow++ will generate a generic one.
- --print=pdf : Output format of the image. Other options include: eps, gif, jpg, png.
- --outfile=*outfile* : Name of the output image. If unused, aflow++ will generate a generic name.

Phonon dispersion and phonon DOS can be plotted in the same manner using aflow --plotphdisp and aflow --plotphdos, respectively, or as a combined plot with aflow --plotphdispdos (see Fig. 9(c)). For these plots, T_{min} and T_{max} are replaced by the minimum and maximum frequency, respectively. Additionally, the following options can be specified:

- --unit=THz : Unit of the frequencies. Other options include: Hz, eV, meV, rcm, and cm $^{-1}$.
- --projection=atoms : Plot atom-projected DOS.

Individual phonon modes can be exported into animation files that can be read by the XCrysDen software and converted into videos or GIFs [132]. This can be done with the aflow --visualize_phonons command. It has the following mandatory options:

- --q=*qpoints* : the q -points as comma-separated triplets in fractional coordinates, e.g., 0.5,0.5,0.5. Multiple q -points can be specified.
- -D *directory* : the directory of the phonon calculation.

The following are optional:

- --amplitude=*amplitudes* : Amplitude of the displacements.
- --branches=*branches* : Branch numbers of the phonon mode as a comma-separated list.
- --periods=*periods* : Number of oscillation periods displayed in the output file.
- --steps=*steps* : Number of steps per period.
- --scell= $s_1s_2s_3$: Supercell dimensions of the structure in the animation.

A snapshot of such a visualization GIF can be found in Fig. 9(d).

7.3. aflow-QHA: The quasi-harmonic approximation library

Thermomechanical properties from phonons. Phonon frequencies strongly depend on the volume of the structure, as shown in Fig. 10(a). In the harmonic approximation, they are calculated at equilibrium volume V_{eq} , i.e., the volume at 0 K. At finite temperatures, the volume of the structure changes, which also changes the phonon frequencies and thus F_{vib} . Knowledge about the volume–temperature curve is thus essential for a more accurate calculation of a material's free energy and thermomechanical properties.

The free energy can be calculated as:

$$F(V, T) = E_0(V) + F_{\text{elec}}(V, T) + F_{\text{vib}}(V, T),$$

where E_0 is the potential energy and F_{elec} the electronic free energy:

$$F_{\text{elec}}(V, T) = \int_0^\infty g_e(\epsilon, V) [f_{\text{FD}} - k_B T s_T(\epsilon)] d\epsilon - \int_0^{E_F} g_e(\epsilon, V) d\epsilon.$$

Here, $g_e(\epsilon, V)$ is the electronic DOS, $f_{\text{FD}} = f_{\text{FD}}(\epsilon, T)$ is the Fermi–Dirac distribution, E_F is the Fermi energy, and $s_T(\epsilon) = f_{\text{FD}} \log(f_{\text{FD}}) + (1 - f_{\text{FD}}) \log(1 - f_{\text{FD}})$ is the electronic entropy at ϵ .

In the quasi-harmonic approximation (QHA) [40,41], $V_{\text{eq}}(T)$ is obtained by calculating $F(T)$ over a set of volumes V_i and fitting the volume-dependent free energy to an equation of state (EOS) for each temperature. aflow-QHA provides the Murnaghan EOS [133]:

$$F(V) = F_{\text{eq}} + \frac{BV_{\text{eq}}}{B'(B'-1)} \left(\frac{V}{V_{\text{eq}}} \right)^{1-B'} + \frac{BV}{B'} - \frac{BV_{\text{eq}}}{B'-1},$$

where B is the bulk modulus and B' its pressure derivative; the n th order Birch–Murnaghan EOS ($2 \leq n \leq 4$) [134]:

$$F(V) = \sum_{i=0}^n f_i V^{-\frac{2}{3}i},$$

where f_i are parameters obtained from a polynomial fit; and the stabilized jellium EOS [135,136]:

$$F(V) = \sum_{i=0}^3 f_i V^{-\frac{1}{3}i}.$$

The fitted EOS can then be used to calculate a variety of thermomechanical properties, such as the volumetric thermal expansion coefficient β , the bulk modulus B , the isochoric and isobaric heat capacities C_V and C_P , and the average Grüneisen parameter γ :

$$\beta(T) = \frac{d \log V_{\text{eq}}(T)}{dT} = \frac{1}{V_{\text{eq}}(T)} \frac{dV_{\text{eq}}(T)}{dT},$$

$$B(T) = V(T) \frac{\partial^2 F(V, T)}{\partial V^2},$$

$$C_V(T) = -T \frac{\partial^2 F(V, T)}{\partial T^2},$$

$$C_P(T) = C_V(T) + V_{\text{eq}}(T) B(T) \beta^2(T) T,$$

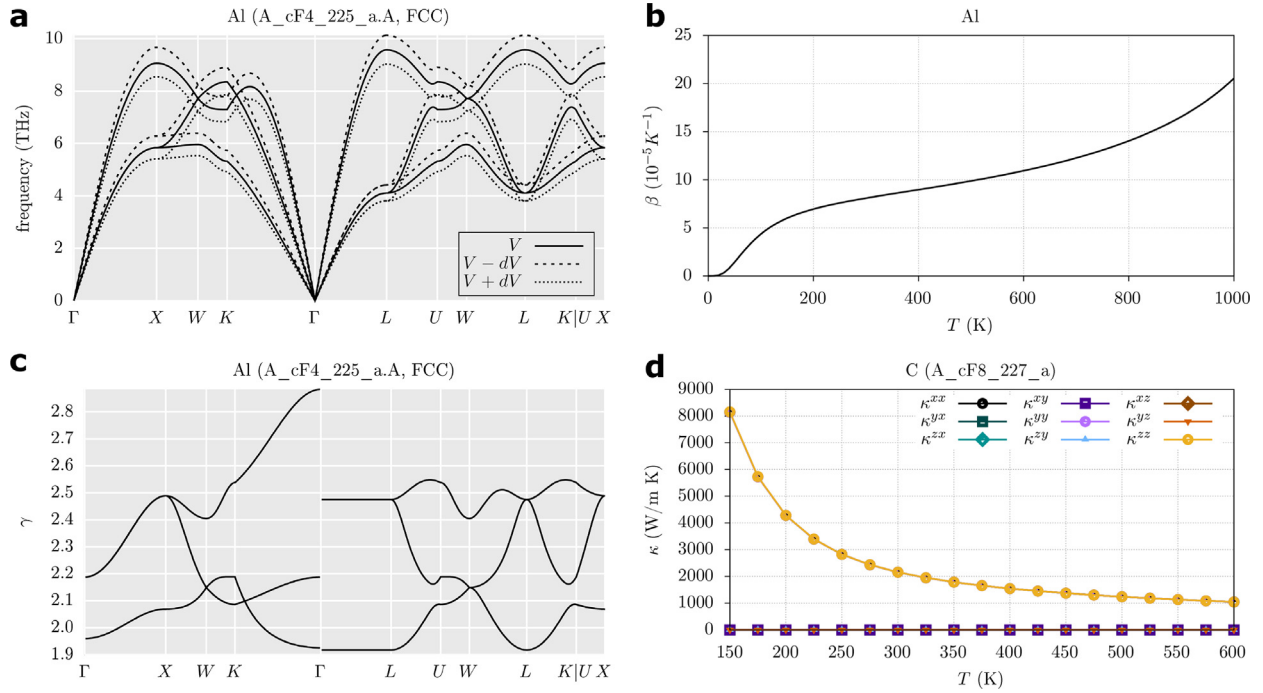


Fig. 10. Properties calculated by aflow-QHA and aflow-AAPL. (a) Volume-dependent phonon dispersion, (b) volumetric thermal expansion coefficient β , and (c) dispersion of Grüneisen parameters of Al. (d) Thermal conductivity tensor components of diamond.

$$\gamma(T) = \beta(T)B(T)V_{\text{eq}}(T)C_V^{-1}(T).$$

The Grüneisen parameter can also be obtained as a mode-dependent quantity:

$$\gamma_\lambda(T) = -\frac{V(T)}{\omega_\lambda(T)} \frac{\partial \omega_\lambda(T)}{\partial V}. \quad (4)$$

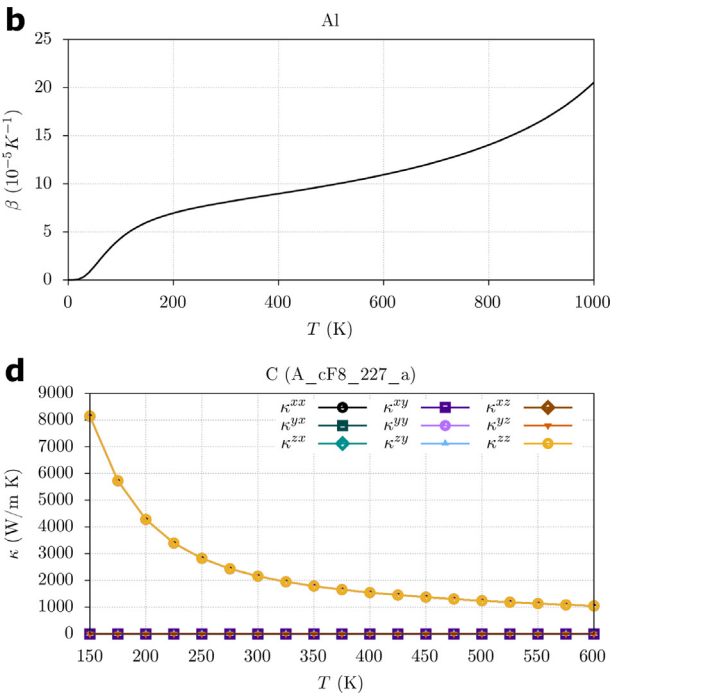
Performing QHA calculations. QHA calculations are activated by adding [AFLOW_QHA] CALC into the aflow.in file or by calling the input file generation command (--aflow_proto) with the --module=qha option. A full list of parameters is available in the APL README, which can be created using the command:

```
aflow --readme=apl
```

The aflow-QHA workflow is similar to aflow-APL in that it consists of a relaxation, a subdirectory creation, and a post-processing step. The subdirectories, however, are different from APL and depend on the properties that are calculated. Calculating thermomechanical properties by fitting $F(V)$ to an EOS is set by [AFLOW_QHA] EOS in the aflow.in file:

```
[AFLOW_QHA] EOS=ON
[AFLOW_QHA] EOS_DISTORTION_RANGE=V_start:V_end:V_step
[AFLOW_QHA] EOS_MODEL=EOS
[AFLOW_QHA] INCLUDE_ELEC_CONTRIB=ON
[AFLOW_QHA] PDIS_T=temperatures
```

This requires phonon and static calculations at multiple volumes, which are determined by [AFLOW_QHA] EOS_DISTORTION_RANGE. [AFLOW_QHA] EOS_MODEL selects the EOS that is used to fit $F(V)$ using the Murnaghan (M), n th order Birch-Murnaghan (BM n), or the stabilized jellium (SJ) model. [AFLOW_QHA] INCLUDE_ELEC_CONTRIB includes or excludes F_{elec} into the free energy. For this procedure, two subdirectories are created for each volume, one for a static calculation to obtain E_0 and F_{elec} and another one for a phonon calculation to determine F_{vib} . The latter is an aflow-APL run without any prior structure relaxation. After finishing all subdirectories, post-processing can be started by running aflow inside the parent directory



again. [AFLOW_QHA] PDIS is a comma-separated list of temperatures for which phonon dispersions are calculated. The temperature range for thermomechanical properties is set via [AFLOW_APL] TPT.

It is also possible to only calculate the Grüneisen dispersion using the following options:

```
[AFLOW_QHA] GP_FINITE_DIFF=ON
[AFLOW_QHA] GP_DISTORTION=ΔV
```

This results in three APL subdirectories at V_{eq} and $\Delta V\%$ compression and expansion.

Visualization options. aflow-QHA provides several output options for its calculated properties. Thermophysical properties can be plotted analogously to --plotthermo for APL:

```
aflow --plotthermoqha
```

where --plotthermoqha accepts multiple optional inputs: the *directory* of the QHA calculation (default: current directory), and the minimum (T_{min}) and maximum (T_{max}) temperatures (default: full range of calculated temperatures); i.e., --plotthermoqha=*directory*, T_{min} , T_{max} . An example plot for the volumetric thermal expansion coefficient of aluminum is shown in Fig. 10(b). The EOS model can be specified using --eosmodel=M which also accepts BM2, BM3, BM4, and SJ.

The Grüneisen parameter dispersion can be plotted the same way using --plotgrdisp. For this plot, T_{min} and T_{max} are replaced by the minimum and maximum Grüneisen parameter. An example plot can be found in Fig. 10(c).

7.4. aflow-AAPL: The automatic anharmonic phonon library

Lattice thermal conductivity. Both aflow-APL and aflow-QHA use harmonic phonons to describe the vibrations inside a material, which neglects phonon-phonon scattering. The aflow++ Automatic Anharmonic Phonon Library (AAPL) includes these interactions explicitly for three-phonon processes, allowing it to calculate the lattice

thermal conductivity of a material [28]. The thermal conductivity tensor k can be calculated as:

$$k^{\alpha\beta} = \frac{\hbar^2}{k_B T^2} \sum_{\lambda} \omega_{\lambda} n_{\lambda} (n_{\lambda} + 1) v_{\lambda\alpha} F_{\lambda\beta}.$$

F_{λ} is the mean-free displacement of the mode and can be expanded as a first-order perturbation:

$$F_{\lambda} = \tau_{\lambda}^0 (\mathbf{v}_{\lambda} + \Delta_{\lambda}),$$

where Δ_{λ} is the perturbation and τ_{λ}^0 is the relaxation time, or inverse scattering rate, of the phonon mode. For lattice thermal conductivity, the scattering rate consists of three components:

$$(\tau_{\lambda}^0)^{-1} = (\tau_{\lambda}^{\text{ph-ph}})^{-1} + (\tau_{\lambda}^{\text{iso}})^{-1} + (\tau_{\lambda}^{\text{grain}})^{-1}.$$

$(\tau_{\lambda}^{\text{grain}})^{-1}$ is the grain boundary scattering rate:

$$(\tau_{\lambda}^{\text{grain}})^{-1} = \frac{|\mathbf{v}_{\lambda}|}{L},$$

with L being the size of the grain. $(\tau_{\lambda}^{\text{iso}})^{-1}$ is the isotope scattering rate:

$$(\tau_{\lambda}^{\text{iso}})^{-1} = \frac{1}{N_q} \sum_{\lambda'} \frac{\pi \omega_{\lambda}^2}{2} \sum_{\kappa} g(\kappa) \left| \mathbf{e}_{\lambda}^*(i) \mathbf{e}_{\lambda}(i) \right|^2 \delta(\omega_{\lambda} - \omega_{\lambda'}),$$

where $g(i)$ is the Pearson deviation coefficient of masses of the isotopes of atom κ . The weights δ can be calculated using the linear tetrahedron method [59,137].

$(\tau_{\lambda}^{\text{ph-ph}})^{-1}$ are the phonon-phonon scattering rates and are the most computationally expensive terms to obtain:

$$(\tau_{\lambda}^{\text{ph-ph}})^{-1} = \frac{1}{N_q} \left[\sum_{\lambda', \lambda''} (n_{\lambda'} - n_{\lambda''}) W_{\lambda\lambda'\lambda''}^+ + \sum_{\lambda', \lambda''} \frac{n_{\lambda'} + n_{\lambda''} + 1}{2} W_{\lambda\lambda'\lambda''}^- \right],$$

where W are intrinsic scattering rates. $W_{\lambda\lambda'\lambda''}^+$ refers to the combination of modes λ and λ' into λ'' and $W_{\lambda\lambda'\lambda''}^-$ to the splitting of λ into λ' and λ'' . They can be calculated using:

$$W_{\lambda\lambda'\lambda''}^{\pm} = \frac{\hbar \pi}{4} \frac{|\mathbf{V}_{\pm}|^2}{\omega_{\lambda} \omega_{\lambda'} \omega_{\lambda''}} \delta(\omega_{\lambda} \pm \omega_{\lambda'} - \omega_{\lambda''}) \times \delta(\mathbf{q} \pm \mathbf{q}' - \mathbf{q}'' + \mathbf{G}).$$

The δ -terms imply that only processes that conserve energy and quasi-momentum (with a phase of a reciprocal lattice vector $\mathbf{G}$) are considered. $\mathbf{V}_{\pm}$ is the scattering matrix:

$$V_{\pm} = \sum \Phi_{\alpha\beta\gamma} (l\kappa; l'\kappa', l''\kappa'') \frac{e_{\pm\lambda}^{\alpha}(\kappa) e_{\pm\lambda'}^{\beta}(\kappa') e_{\pm\lambda''}^{\gamma}(\kappa'')}{\sqrt{m_{\kappa} m_{\kappa'} m_{\kappa''}}} \times \exp[i(\pm \mathbf{q}' \cdot \mathbf{R}_{l'} - \mathbf{q}'' \cdot \mathbf{R}_{l''})].$$

$\Phi_{\alpha\beta\gamma} (l\kappa; l'\kappa', l''\kappa'')$ are the third-order anharmonic force constants and are the central problem for thermal conductivity calculations. AAPL uses the central difference method to calculate them. For this purpose, two atoms need to be displaced along linearly-independent Cartesian coordinates, resulting in many calculations that can be reduced using symmetry.

Performing aflow-AAPL calculations. AAPL calculations are activated by adding [AFLOW_AAPL] CALC to the aflow.in file. As with APL, this can also be achieved by adding --module=aapl to the input generation command (--aflow_proto). The workflow generally follows the APL workflow and uses many of the same parameters, but creates additional subdirectories to calculate forces of supercells with two atoms displaced from their equilibrium position. It also creates the subdirectories required by APL to calculate phonon frequencies and group velocities.

As a preparatory step, AAPL determines the atomic pairs that need to be displaced to calculate anharmonic IFCs. The cutoff distance for these pairs can be set using a radius, a number of coordination shells, or both. For example:

```
[AFLOW_AAPL] CUT_RAD=5.5
[AFLOW_AAPL] CUT_SHELL=6
```

In this case, it will set the cutoff to at least 5.5 Å while also containing at least six coordination shells. To use only one option, the other needs to be set to zero. IFCs between pairs with a distance beyond the cutoff are set to zero. After determining these pairs, they are reduced by symmetry with aflow-SYM to only contain symmetrically-inequivalent ones [31], which results in fewer calculations than in many other packages [28].

After determining the anharmonic IFCs, the Boltzmann Transport Equation is solved to calculate the thermal conductivity tensor. This is a computationally expensive step and is supported by an on-the-fly parallelization scheme inside aflow++. The calculation conditions are set by the following parameters in the aflow.in file:

```
[AFLOW_AAPL] TCT=T_start:T_end:T_step
[AFLOW_AAPL] THERMALGRID=q1xq2xq3
[AFLOW_AAPL] ISOTOPE=ON
[AFLOW_AAPL] BOUNDARY=OFF
[AFLOW_AAPL] NANO_SIZE=d_grain
```

[AFLOW_AAPL] TCT sets the start and end temperatures and the temperature step size; [AFLOW_AAPL] THERMAL_GRID is the dimensions of the $\mathbf{q}$ -point grid for phonon-phonon scattering processes; [AFLOW_AAPL] ISOTOPE and [AFLOW_AAPL] BOUNDARY set whether isotope and grain boundary scattering, respectively, are included; and [AFLOW_AAPL] NANO_SIZE sets the size of the grains to d_{grain} nm. A full list of aflow.in parameters can be found in the APL/AAPL README, which can be displayed using:

```
aflow --readme=aapl
```

Visualization options. aflow-AAPL can plot the thermal conductivity tensor as a function of temperature. The command is analogous to --plotthermo for APL:

```
aflow --plotcond
```

where --plotcond accepts multiple optional inputs: the *directory* of the AAPL calculation (default: current directory), and the minimum ($T_{\min}$) and maximum ($T_{\max}$) temperatures (default: full range of calculated temperatures); i.e., --plotcond=*directory*, $T_{\min}$, $T_{\max}$. An example plot is shown in Fig. 10(d).

8. Modeling disorder

8.1. aflow-POCC: The partial occupation module for chemical disorder

The aflow-POCC module is a framework for modeling chemically disordered systems, also known as substitutionally disordered systems or random alloys [38]. Such systems have sites with partial, probabilistic occupancies, e.g., the high-temperature Cu_3Au phase [138,139] is characterized by a single site on an fcc lattice (A_cF4_225_a) with two occupants having probabilities matching the stoichiometry. The properties of Cu_3Au cannot be calculated directly using standard ab-initio codes like VASP relying on plane wave basis sets. Instead, aflow-POCC represents a random alloy as an ensemble of ordered supercells (an example is illustrated in Fig. 11), the properties of which are thermally averaged to yield that of the disordered system. The workflow is as follows: ensemble set generation (pre-processing), calculation of properties (using, e.g., VASP), and analysis (post-processing). All three steps are performed by aflow++.

PARTCAR file. aflow-POCC takes as input a PARTCAR, a modified POSCAR that specifies the partial occupancies of the sites and some tolerances for the algorithm. An example PARTCAR for the $\text{Ag}_{8.733}\text{Cd}_{3.8}\text{Zr}_{3.267}$ system is provided here, with inputs specific to the PARTCAR file marked with a box.

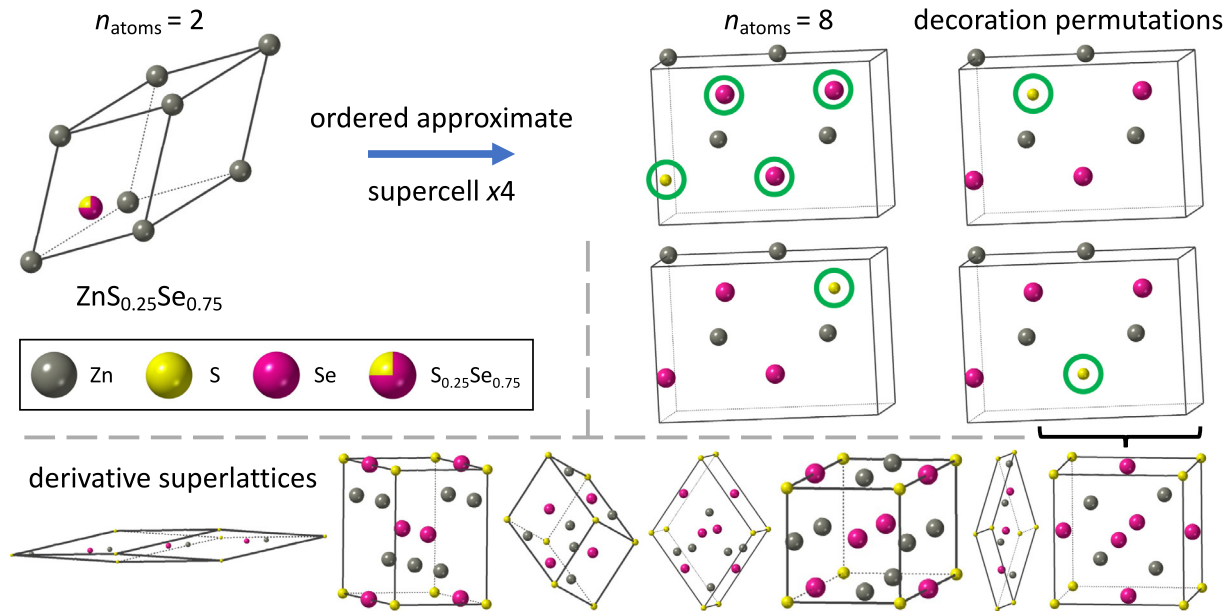


Fig. 11. Generation of ordered representatives for $\text{ZnS}_{0.25}\text{Se}_{0.75}$. The aflow-POCC algorithm constructs the smallest supercell satisfying the stoichiometry of the disordered system. For $\text{ZnS}_{0.25}\text{Se}_{0.75}$, a supercell size of four is needed given the occupancies of the shared site. An illustration of a supercell representative is provided on the right, with the occupants of the shared site highlighted in green. The four decoration-permutations of this supercell are also shown, moving the sulfur (yellow) atom to each of the allowed sites, which in this case all produce symmetrically-equivalent structures and are thus degenerate. This derivative superlattice, a uniform expansion of the original lattice, is only one of seven distinct possibilities producing a supercell size of four, all of which also have four decoration-permutations to consider. This results in 28 total representative structures, seven of which are unique.

```

PARTCAR of Ag8.733Cd3.8Zr3.267
1.0 0.001 0.001
5.76 5.76 5.76 90 90 90
8*1+1*0.733 3*1+1*0.8 3*1+1*0.267
Direct(17) Partial [A8.73B3.8C3.27]
0.25 0.25 0.25 Ag pocc=1
0.75 0.75 0.25 Ag pocc=1
0.75 0.25 0.75 Ag pocc=1
0.25 0.75 0.75 Ag pocc=1
0.25 0.25 0.75 Ag pocc=1
0.75 0.75 0.75 Ag pocc=1
0.25 0.75 0.25 Ag pocc=1
0.75 0.25 0.25 Ag pocc=1
0.50 0.50 0.50 Ag pocc=0.733
0.00 0.50 0.50 Cd pocc=1
0.50 0.00 0.50 Cd pocc=1
0.50 0.50 0.00 Cd pocc=1
0.00 0.00 0.00 Cd pocc=0.8
0.50 0.00 0.00 Zr pocc=1
0.00 0.50 0.00 Zr pocc=1
0.00 0.00 0.50 Zr pocc=1
0.50 0.50 0.50 Zr pocc=0.267

```

The first box is next to the scaling factor and contains the site and stoichiometry tolerances for the supercell size algorithm. The default tolerance is 0.001 for both, which can be changed in the aflow.rc file. The second box marks the line containing the number of each type of atom, which now accepts fractional occupancies. The '+' sign separates occupants of the same type with different occupancy values. The third box is next to the 'Direct' coordinates designation and signals to aflow++ that the sites that follow will have 'Partial' occupancy. Only the 'P' in 'Partial' is required, similar to the 'D' in 'Direct'. The following three boxes mark the two partially occupied sites: one shared between Ag ($x = 0.733$) and Zr ($x = 0.267$) and the other between Cd ($x = 0.8$) and a vacancy ($x = 0.2$). Take note that the order of the sites

matches that specified by the number-of-each-type line with the species listed in alphabetic order.

A PARTCAR file can be constructed automatically by aflow++ using the aforementioned --proto and --afLOW_proto commands, which leverage the extensive prototyping suite [14] and library of naturally-occurring compounds [15–17] to construct the parent structure. The first step is to construct the --proto command for the base structure. For example,

```
afLOW --proto=AB_hP4_186_b_b-001:S:Zn --abccar
```

generates the POSCAR for the high-temperature (wurtzite) phase of ZnS [140].

```

POSCAR of SZn/AB_hP4_186_b_b-001.AB
1.0
3.5155 3.5155 3.5155 90 90 120
2 2
Direct(4) [A2B2]
0.333333 0.666667 0.374800 S
0.666667 0.333333 0.874800 S
0.333333 0.666667 0.000000 Zn
0.666667 0.333333 0.500000 Zn

```

The --abccar flag prints the lattice as $|a|$, $|b|$, $|c|$, α , β , and γ instead of the usual matrix format. Take note that this structure has four sites and two species. To construct the $\text{ZnS}_{0.5}\text{Se}_{0.5}$ structure, where both sulfur sites are partially occupied with selenium, the --proto command is modified as such:

```

afLOW --proto=AB_hP4_186_b_b-001:S:Se:Zn --abccar --
  pocc_params=P0-0.5xA-0.5xB_P1-0.5xA-0.5xB_P2-1
  xC_P3-1xC [--pocc_tol=0.001:0.001]

```

Notice that the changes from the original --proto command include adding selenium to the colon-separated species list (in alphabetic order) and the --pocc_params/--pocc_tol options. The --pocc_params option is a list of underline-separated values for each site (Position) with the first site of the parent structure indicated

by P0. Following the site specification is a list of comma-separated values of the occupancies for that site, with A, B, and C referring to the colon-separated, alphabetically-ordered set of species provided in the --proto option. All sites must be specified with the --pocc_params options. The optional --pocc_tol takes as input a colon-separated list of site and stoichiometry tolerances for the supercell size algorithm. For the site tolerance, a negative integer can be accepted instead, which fixes the desired supercell size and allows the generation of larger ensemble sets (better sampling) beyond what is required to satisfy the stoichiometry exactly. The command generates the following PARTCAR:

```
PARTCAR of S0.5Se0.5Zn/AB_hP4_186_b_b-001.AB
1.0 0.001 0.001
3.6017 3.6017 5.8988 90 90 120
2*0.5 2*0.5 2*1
Direct(6) Partial [A1B1C2]
0.333333 0.666667 0.374800 S pocc=0.5
0.666667 0.333333 0.874800 S pocc=0.5
0.333333 0.666667 0.374800 Se pocc=0.5
0.666667 0.333333 0.874800 Se pocc=0.5
0.333333 0.666667 0.000000 Zn pocc=1
0.666667 0.333333 0.500000 Zn pocc=1
```

The same PARTCAR can be generated with a shorter command:

```
aflow --proto=AB_hP4_186_b_b-001:S:Se:Zn --abccar --
  ↪ pocc_params=S0-0.5xA-0.5xB_S1-1xC
```

where the partial occupancy values are specified by Species, i.e., the group of sites having sulfur atoms, S0, (zinc atoms, S1) in the parent structure. This specification can truncate the input size substantially for larger structures. The two specifications, P and S, can be mixed, but should be avoided in practice as it can be challenging to keep track of the two sets of indices simultaneously.

A PARTCAR can also be converted to a CIF file with partially occupied sites and vice versa. For example:

```
aflow --proto=AB_hP4_186_b_b-001:S:Se:Zn --cif --
  ↪ pocc_params=S0-0.5xA-0.5xB_S1-1xC
```

generates the following output:

```
# AFLOW.org Repositories
# SSeZn/AB_hP4_186_b_b-001.AB:POCC_S0-0.5xA-0.5xB_S1
  ↪ -1xC
data_SSeZn
_pd_phase_name AB_hP4_186_b_b-001.AB:POCC_S0-0.5xA
  ↪ -0.5xB_S1-1xC
_cell_length_a 3.6017
_cell_length_b 3.6017
_cell_length_c 5.8988
_cell_angle_alpha 90.0000
_cell_angle_beta 90.0000
_cell_angle_gamma 120.0000
_symmetry_space_group_name_H-M 'P1'
_symmetry_Int_Tables_Number 1
loop_
_symmetry_equiv_pos_site_id
_symmetry_equiv_pos_as_xyz
  1 x,y,z
loop_
_atom_site_label
_atom_site_occupancy
_atom_site_fract_x
_atom_site_fract_y
_atom_site_fract_z
_atom_site_type_symbol
S1 0.5000 0.3333 0.6667 0.3748 S
S2 0.5000 0.6667 0.3333 0.8748 S
Se3 0.5000 0.3333 0.6667 0.3748 Se
```

```
Se4 0.5000 0.6667 0.3333 0.8748 Se
Zn5 1.0000 0.3333 0.6667 0.0000 Zn
Zn6 1.0000 0.6667 0.3333 0.5000 Zn
```

A CIF file can also be converted to a PARTCAR with the --vasp flag:

```
aflow --proto=AB_hP4_186_b_b-001:S:Se:Zn --cif --
  ↪ pocc_params=S0-0.5xA-0.5xB_S1-1xC | aflow --
  ↪ vasp
```

By switching the --proto to --aflow_proto, aflow++ will generate an aflow.in file containing the PARTCAR inside. The directory structure is largely the same as that specified in Section 2, except in this case, the third (structure) layer is a mix of the parent structure prototype AB_hP4_186_b_b-001.AB and the --pocc_params. The directory tree also protects from the generation of duplicates, although it does not prevent the ones produced from mixing P and S specifications. The aflow-POCC aflow.in contains [AFLOW_POCC] CALC activating the module and the PARTCAR wrapped with [POCC_MODE_EXPLICIT] START.POCC_STRUCTURE and [POCC_MODE_EXPLICIT] STOP.POCC_STRUCTURE. Running aflow --run in the directory with this aflow.in will, in a single step, determine the optimal supercell size, generate the ensemble of ordered representatives, and create a subdirectory for each (denoted with ARUN.POCC). Each directory contains a child aflow.in setting up that particular calculation. The subdirectory structure enables parallelization across the ordered representatives. Note that the characterization of the ordered representatives is not strictly limited to the usual relaxation/static workflows, but can also include the analysis of thermomechanical properties, themselves creating additional parent/child layers which ultimately resolve the properties of the disordered system. These advanced workflows will be covered in subsequent sections.

Creating the right supercell size. Following the creation of a PARTCAR, the aflow-POCC algorithm determines the smallest supercell size that satisfies the site and stoichiometry tolerances specified. The algorithm iterates through incrementally larger supercell sizes, identifying the optimal occupation (f) and the associated site (Δ) and stoichiometry (Δ_x) errors. An example is provided in Table 2 for $\text{Ag}_{8.733}\text{Cd}_{3.8}\text{Zr}_{3.267}$. The stop condition is achieved when both the maximum of the site and stoichiometry errors are below the tolerance specified in the PARTCAR. The default tolerances of 0.001 should find the supercell size that satisfies the stoichiometry exactly. To optimize either the site or stoichiometry errors individually, set the other tolerance to 1.

One challenge that the algorithm overcomes is the filling of sites for supercell sizes that may be too small. For example, consider the first iteration in Table 2. For this case, there is only one site and two possible occupants. The best choice (the one reducing the error) is to fill the site with the occupant having the higher concentration (Ag). If both occupants had equal (50%) concentrations, the algorithm would leave the site unfilled. While filling the site with one of the occupants would yield the same error, the algorithm has no way of deciding which occupant to pick. The issue becomes more problematic with more occupants having equal concentrations. This requires grouping these same-site, equi-concentration occupants and incrementing their fillings simultaneously or not at all. To avoid the generation of vacancies, the input occupancy values should be altered to bias the components of interest.

Ensemble set generation. After the algorithm determines the optimal supercell size, it proceeds to construct all of the possible derivative supercells. An illustration of the two steps — determining the unique derivative superlattices (geometries) and all of the corresponding decorations (colorings) — is shown in Fig. 11. The Hermite normal form matrices [141,142] generate the unique set of derivative superlattices. For the $\text{Zn}_{0.25}\text{Se}_{0.75}$ system shown in Fig. 11, there are seven unique derivative superlattices, each of which has four decoration-permutations, generating 28 total representative structures. aflow-POCC then determines which of these supercells are unique, employing

Table 2

Evolution of the algorithm to determine the optimal supercells size for $\text{Ag}_{0.733}\text{Cd}_{3.8}\text{Zr}_{3.267}$. i is the iteration step of the algorithm, δ is the site occupancy resolution, f is the iteration's choice fraction for the site, Δ is the error in site occupancy, Δ_x is the error in the stoichiometry, and n is the supercell size.

i	δ	$\text{Ag}_{0.733}$		$\text{Zr}_{0.267}$		$\text{Cd}_{0.8}$		$\max(\Delta)$	$\max(\Delta_x)$	n
		f	Δ	f	Δ	f	Δ			
1	1.000	1/1	0.267	0/1	0.267	1/1	0.200	0.267	0.019	1
2	0.500	1/2	0.233	1/2	0.233	2/2	0.200	0.233	0.021	2
3	0.333	2/3	0.066	1/3	0.066	2/3	0.133	0.133	0.006	3
4	0.250	3/4	0.017	1/4	0.017	3/4	0.050	0.050	0.003	4
5	0.200	4/5	0.067	1/5	0.067	4/5	0.000	0.067	0.004	5
6	0.167	4/6	0.066	2/6	0.066	5/6	0.033	0.066	0.005	6
7	0.143	5/7	0.019	2/7	0.019	6/7	0.057	0.057	0.003	7
8	0.125	6/8	0.017	2/8	0.017	6/8	0.050	0.050	0.003	4
9	0.111	7/9	0.045	2/9	0.045	7/9	0.022	0.045	0.004	9
10	0.100	7/10	0.033	3/10	0.033	8/10	0.000	0.033	0.002	10
11	0.091	8/11	0.006	3/11	0.006	9/11	0.018	0.018	0.001	11
12	0.083	9/12	0.017	3/12	0.017	10/12	0.033	0.033	0.002	12
13	0.077	10/13	0.036	3/13	0.036	10/13	0.031	0.036	0.003	13
14	0.071	10/14	0.019	4/14	0.019	11/14	0.014	0.019	0.001	14
15	0.067	11/15	0.000	4/15	0.000	12/15	0.000	0.000	0.000	15

the Universal Force Field method [143] to calculate an approximate energy for each structure which can be quickly compared to resolve whether two structures are identical. For $\text{ZnS}_{0.25}\text{Se}_{0.75}$, seven unique supercells will be passed along to the next workflow stage: DFT calculation with VASP. The algorithm has been optimized for speed and reduction of memory footprint, critical as the number of total derivative structures to consider can easily run into the billions. It is important to mention that aflow-POCC considers all of the possible ordered representatives, and not just a subset limited to a particular superlattice. This ensures proper sampling for the calculation of spectral descriptors like the entropy-forming-ability (EFA) discussed in detail in the next section.

Post-processing analysis. Upon completion of the ARUN.POCC subdirectory calculations, running `aflow --run` again in the parent directory will initiate the post-processing analysis. For the usual relaxation/static runs of the ordered representatives, aflow-POCC will resolve ensemble average DOS, bandgaps, and magnetic moments [38]. A Boltzmann-weight is calculated for the averaging: the i -structure's probability depends on its degeneracy count (g_i), amount of disorder relative to the minimum (ground-state) structure in the set ($H_i - H_{gs}$), and a tuning parameter mimicking temperature (T_{POCC}):

$$P_i = \frac{g_i e^{-(H_i - H_{gs})/k_B T_{\text{POCC}}}}{\sum_i g_i e^{-(H_i - H_{gs})/k_B T_{\text{POCC}}}}.$$

The temperature parameter dictates how much disorder to incorporate in the analysis, and roughly correlates with the synthesis temperature of the material. The ensemble average DOS is written to DOSCAR-type files at various temperature snapshots, e.g., `DOSCAR.pocc_T0300K.xz`. The snapshots can be set from the command line upon creation of the parent `aflow.in` (in combination with the `--aflow_proto` command) with `--temperature=0:2400:300`, or by hand inside the parent `aflow.in` by adding the line `[AFLOW_POCC] TEMPERATURE=0:2400:300`, both of which will perform temperature snapshots between 0–2400 K in 300 K increments. The default temperature snapshots are set inside the `aflow.rc` file. The other properties are written to the `aflow.pocc.out` file containing a summary of the full analysis, including the degeneracy count for each unique structure and associated temperature-specific properties, including the Boltzmann-weighted properties of each structure. The properties are organized by temperature snapshots.

aflow-POCC also calculates the EFA, which has led to the discovery of 10 high-entropy carbides [8,10]. The descriptor is the inverse of the

standard deviation of the energy spectrum of the ordered representatives. Being a spectral descriptor, the EFA analysis is reliant on proper sampling, requiring full exploration of possible ordered representatives.

aflow-POCC has been validated for a number of systems and properties [8,10,34,38,144], including electronic, magnetic, thermodynamic, and thermomechanical properties. It has also been demonstrated that the accuracy improves with larger supercell sizes [38], providing better sampling and capturing longer-range effects. Convergence with supercell size should be checked for each property and balanced with the feasibility of larger supercell calculations.

Comparison to other disordered system models. aflow-POCC is a multiple-supercells approach, which is often compared with special quasirandom structures (SQS), a supercell approach [145]. SQS is, for a given supercell size, the singular supercell structure that minimizes the site correlations to emulate the random alloy, and is thus among the set of structures enumerated by aflow-POCC. Using a single supercell to model the disordered system represents the strictly infinite-temperature solution. aflow-POCC allows finite-temperature modeling, including near the transition temperature, with the introduction of an ensemble set (degeneracies) and tuning parameters into the framework. To best model the properties of the disordered system, the needed SQS is often quite large and very low-symmetry, making it a challenging calculation to converge. Alternatively, aflow-POCC approaches disordered-systems-modeling from the other end, generating many smaller cell representatives, which can be easily parallelized for high-throughput workflows.

Elasticity and GIBBS analysis. The aflow++ Elasticity Library (AEL) and aflow++ GIBBS Library (AGL) modules (see Section 7.1) can be used to calculate the thermal and elastic properties of ordered materials. These methods have now been integrated with the aflow-POCC module to enable calculation of the thermal and elastic properties of configurationally disordered materials.

For every derivative structure or “tile” generated by POCC, full AEL and/or AGL calculations are performed to obtain their thermoelastic properties, including the elastic constants, bulk and shear moduli, Debye temperature, specific heat capacity at constant volume and at constant pressure, and coefficient of thermal expansion. The results are then thermally averaged using the POCC Boltzmann probabilities P_i , a function of the POCC temperature T_{POCC} . For example, the thermally averaged VRH bulk modulus for a configurationally disordered material would be calculated as

$$B_{\text{VRH}}^{\text{avg}}(T_{\text{POCC}}) = \sum_i P_i(T_{\text{POCC}}) B_{\text{VRH},i},$$

where $B_{\text{VRH},i}$ is the VRH bulk modulus for tile i . Since the equations for the bulk and shear moduli are linear in the elastic constants, using the thermally averaged elastic constants to calculate the bulk and shear moduli would give the same result as averaging the bulk and shear moduli for each tile.

Note that in the case of AGL calculations, there are two sets of temperatures: the POCC temperature, which determines the distribution of the tiles present in the material and is often equivalent to a synthesis or annealing temperature; and the AGL temperature, which corresponds to the instantaneous temperature determining the vibrational properties of the material. Both sets of temperatures can be set separately during post-processing, so the same set of DFT calculations can be used to investigate a range of different temperature regimes and combinations.

Performing AGL calculations in the POCC framework requires a completed POCC run with all tiles relaxed and all enthalpies calculated, so that the probabilities needed to ensemble-average the thermal elastic properties are available. To run an aflow-POCC+AEL/AGL calculation, the line `[AFLOW_AGL] CALC` or `[AFLOW_AEL] CALC` should be present in the `aflow.in` file of the parent directory. It is recommended to copy the original `aflow.in` file into a new file (e.g., `aflow_agl.in`) before adding/uncommenting the line, since `aflow.in` files of the same name will be created in the subdirectories.

As in the case of ordered materials, combined AEL-AGL workflows can also be run, e.g., by setting `[AFLOW_AGL]AEL_POISSON_RATIO=ON`. Other AEL and AGL parameters can be set for all tiles by including the appropriate options (described in Section 7.1) in the `aflow.in` file of the parent directory.

Phonon analysis. The `aflow++` Automatic Phonon Library (APL) discussed earlier is limited to calculating phonon properties for ordered materials. Since the POCC method uses a statistical ensemble of ordered structures, APL can be expanded to determine integrated vibrational properties of disordered materials as well [34].

These properties include the vibrational free energy F_{vib} , internal energy U_{vib} , and vibrational entropy S_{vib} . They are calculated from the phonon DOS as shown in the APL section. This opens two avenues to calculate these quantities as ensemble properties: determining them for each derivative structure and ensemble-averaging, or using the ensemble-averaged DOS and integrating it. Due to the linear relationship between the phonon DOS and F_{vib} , U_{vib} , and S_{vib} , the results are independent of that choice:

$$\begin{aligned} F_{\text{vib}}^{\text{avg}}(T) &= \sum_i P_i(T_{\text{POCC}}) F_{\text{vib},i} \\ &= \sum_i P_i(T_{\text{POCC}}) \int_0^\infty f(T, \omega) g_i(\omega) d\omega \\ &= \int_0^\infty f(T, \omega) \sum_i P_i(T_{\text{POCC}}) g_i(\omega) d\omega \\ &= \int_0^\infty f(T, \omega) g^{\text{avg}}(\omega) d\omega. \end{aligned}$$

Here, $F_{\text{vib}}^{\text{avg}}$ and g^{avg} are the ensemble-averaged vibrational free energy and phonon DOS, respectively, and $f(T, \omega) = k_B T \log \left(2 \sinh \frac{\hbar \omega}{2 k_B T} \right)$.

The same relationship can be shown for U_{vib} and S_{vib} . The POCC-APL workflow ensemble averages the DOS, which also provides access to the phonon DOS of the disordered material.

Performing APL calculations in the POCC framework requires a completed POCC run with all derivative structures relaxed and all enthalpies calculated. This ensures that the probabilities needed to ensemble-average the phonon DOS are available. To start the `aflow-POCC` calculation, the line `[AFLOW_APL]CALC` needs to be present in the `aflow.in` file of the parent directory. It is recommended to copy the original `aflow.in` file into a new file (e.g., `aflow_apl.in`) before adding/uncommenting the line since `aflow.in` files of the same name will be created in the subdirectories.

Running `aflow --run` in this directory will create `aflow.in` files for the APL calculations for each derivative structure. All APL options set in the parent input file will be propagated into the subdirectories. The initial structure is taken from the last relaxed run of the ordered representative. The next step is to run `aflow --run` inside each subdirectory, which creates the additional layer of subdirectories needed to calculate force constants and the non-analytical correction, if requested. After finishing all DFT calculations in these directories, `aflow --run` needs to be run again inside the POCC parent directory. It is not required to execute the APL post-processing runs in the individual subdirectories — `aflow-POCC` will calculate the force constants if not present.

The force constants are then determined and used to calculate the phonon frequencies and phonon DOS for each ordered representative. The DOS are ensemble-averaged to calculate thermophysical properties for each POCC temperature. Dynamically unstable derivative structures, i.e., structures with imaginary frequencies in the phonon dispersion, are automatically discarded from the ensemble. This behavior can be turned off via the `aflow.in` option `[AFLOW_POCC]EXCLUDE_UNSTABLE=OFF`.

`aflow-POCC` outputs the phonon DOS for each POCC temperature (in the VASP DOSCAR format), and the vibrational properties calculated from the ensemble-averaged DOS. Both can be plotted the same way as in APL: the DOS plots for all temperatures and the thermophysical properties are generated using `--plotphdos` and `--plotthermo` commands, respectively.

8.2. *aflow-QCA: The quasi-chemical approximation module for chemical disorder*

Solid solution phase transition. Forming solid solution alloys can offer enhanced thermodynamic, chemical and mechanical properties [146, 147]. The `aflow-QCA` module provides a high-throughput [47] *ab-initio* method [39] to predict the temperature at which phase-separated multi-component alloys undergo a transition to become highly disordered solid solutions [148,149]. This method rests on calculating an order parameter, within the generalized quasi-chemical approximation (GQCA) [150,151], whose maximal change, with respect to temperature, defines the order-disorder phase transition at equi-concentration. Then, tracing the locus of the equi-concentration relative entropy [152–154], the phase transition temperature for the whole concentration spectrum is found. In the subsequent sections, the predictive capabilities of this method are demonstrated and corroborated by Monte Carlo simulations [155], CALPHAD calculations [156], and experimental data [149,157–170].

GQCA model. The GQCA model factorizes a parent lattice of N sites and K species into an ensemble of non-overlapping clusters, which are statistically and energetically independent of the surrounding medium. Let an alloy with macroscopic concentration $\{X_1, X_2, \dots, X_K\}$ be characterized by an ensemble of $J+1$ clusters, with each cluster containing n atoms. Here, each j -type cluster has a distinct energy ε_j and concentration $\{x_{1j}, x_{2j}, \dots, x_{Kj}\}$. Then the mixing energy for a given set of clusters is determined by [171]:

$$\Delta E(\mathbf{X}, T) = \sum_{k=1}^K X_k \varepsilon_k^0 + \sum_{j=0}^J P_j(\mathbf{X}, T) \Delta_j,$$

where ε_k^0 are the energies of the pure cluster only containing the k th species, P_j are the probabilities of the j -type cluster in the alloy, and Δ_j are the reduced excess energies defined as:

$$\Delta_j = \varepsilon_j - \sum_{k=1}^K x_{kj} \varepsilon_k^0.$$

Likewise, the mixing entropy is given by:

$$\Delta S(\mathbf{X}, T) = -k_B \left[n \sum_{k=1}^K X_k \log X_k + \sum_{j=0}^J P_j(\mathbf{X}, T) \log \left(\frac{P_j(\mathbf{X}, T)}{P_j^0(\mathbf{X})} \right) \right],$$

where P_j^0 are the probabilities to find the j -type cluster for the ideal solution model [172,173] and the last term is known as the Kullback-Leibler divergence D_{KL} or relative entropy [174].

Finding the cluster probabilities P_j at equilibrium involves minimizing the mixing free energy $\Delta F = \Delta E - T \Delta S$ by solving the set of equations $\partial \Delta F / \partial \mathbf{P} = \mathbf{0}$ with K constraints:

$$\sum_{j=0}^J P_j = 1, \quad \sum_{j=0}^J P_j x_{1j} = X_1, \quad \sum_{j=0}^J P_j x_{2j} = X_2, \quad \dots \quad \sum_{j=0}^J P_j x_{K-1j} = X_{K-1},$$

leaving $J+1-K$ coupled equations. Using the method of Lagrangian multipliers, the solution to this set of equations yields [39,171]:

$$P_j(\mathbf{X}, T) = \frac{P_j^0(\mathbf{X}) e^{n\beta \left[\sum_{k=1}^{K-1} x_{kj} \lambda_k (\mathbf{X}, T) - \Delta_j \right]}}{\sum_{j=0}^J P_j^0(\mathbf{X}) e^{n\beta \left[\sum_{k=1}^{K-1} x_{kj} \lambda_k (\mathbf{X}, T) - \Delta_j \right]}},$$

where $\beta = 1/k_B T$ and λ_k are the Lagrangian multipliers. The energies used in evaluating the previous equation are calculated with the Connolly-Williams method [175], as implemented in Ref. [176,177] using the `aflow.org` repositories [2,13,58,85].

Phase transition order parameter. The order-disorder transition of an alloy can be determined by performing a common tangent construction of the mixing free energy [178]. However, in our model, where the clusters are uncorrelated, this technique is invalid due to the absence of

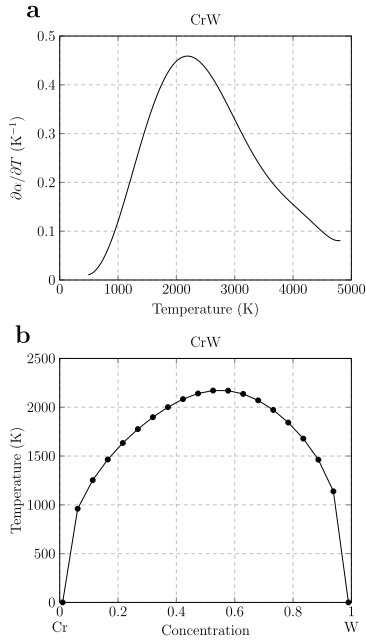


Fig. 12. Properties calculated by aflow-QCA. (a) Partial derivative of the order parameter with respect to temperature and (b) the binodal curve for the CrW alloy.

coherency effects [179]. To overcome this problem, an order parameter is defined by:

$$\alpha(\mathbf{X}, T) = \frac{\mathbf{P} \cdot \mathbf{P}^0}{\|\mathbf{P}\| \|\mathbf{P}^0\|},$$

which measures the deviation of the probability distribution from the high- T limit [180]. Furthermore, α is only evaluated at the equi-concentration $\mathbf{X}^{(ec)}$, where the correlation effects are minimal [181]. Next, the transition temperature at equi-concentration $T_c^{(ec)}$ is defined where the temperature gradient of the order parameter is maximum, similar to Monte Carlo simulations [182], as shown in Fig. 12(a). Finally, an assumption is made: the relative entropy at the transition temperature T_c is independent of the macroscopic concentration, such that:

$$D_{KL}^{(ec)} \equiv D_{KL}(\mathbf{X}^{(ec)}, T_c^{(ec)}) \approx D_{KL}(\mathbf{X}, T_c)$$

yields the transition temperature for the whole concentration spectrum.

Usage. The following command loads the alloy data from the aflow.org repositories, constructs the binodal curve, as shown in Fig. 12(b), and returns the output in plain text (txt) format:

```
aflow --qca --plattice=fcc --elements=Au,Pt
```

- `--qca`: Necessary argument, enters the mode to calculate the phase equilibria properties.
- `--plattice=lattice`: Necessary argument, specifies the parent lattice of the alloy (e.g., fcc).
- `--elements=elements`: Necessary argument, *elements* is a comma-separated list of components present in the alloy.
- `--directory=directory`: Optional argument, specifies the *directory* where to run the calculation. Default is the current directory.

A full list of parameters is available by invoking the command:

```
aflow --qca --usage
```

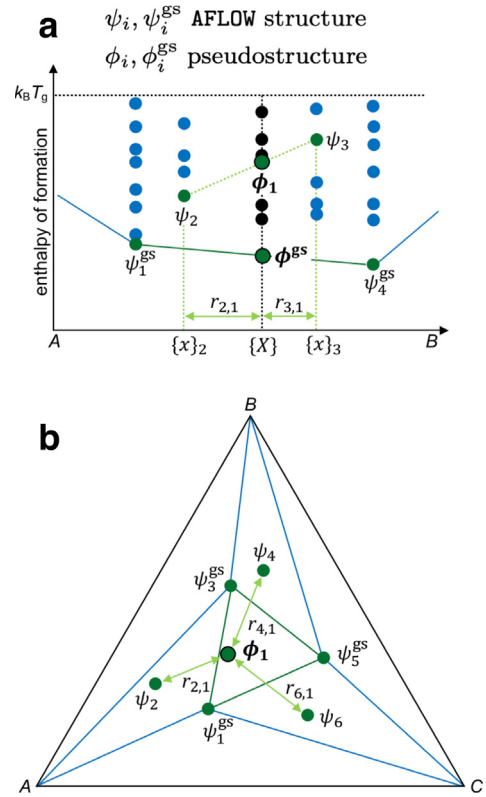


Fig. 13. Schematic of contributions to the GFA for binary (a) and ternary (b) systems. Structures are represented as circles: black circles are at the global stoichiometry, green circles are pairs and triplets that contribute to the GFA at the global stoichiometry, and blue circles are at other stoichiometries. The reference state is the ground state (gs) defined by the convex hull. In the ternary schematic, ϕ^{gs} is located directly beneath ϕ_1 .

8.3. aflow-GFA: The glass-forming-ability module for structural disorder

Metallic glasses are a unique class of materials without the crystalline order typically found in metals [183,184]. Finding new metallic glasses is constrained by the vast combinatorial space [185] and time-consuming experiments. aflow++ can accelerate the discovery of new suitable candidates through the glass-forming ability (GFA) prediction module.

The first iteration of GFA prediction was included in aflow++ by Perim et al. in 2016 [36]. A spectral descriptor was constructed to capture the structural confusion during vitrification into a glass upon cooling of the melt, as described by Greer [186]. It was calculated based on the structures of competing crystalline phases available at a specific stoichiometry in the aflow.org repositories. This first work focused on binary alloy systems, where a discrete exploration along the concentration axis already revealed a good insight into the possible glass-forming structures.

In 2019, the GFA module was subsequently expanded to better capture multi-component alloys by Ford et al. [37]. Employing a new approach of global stoichiometry $\{X\}$, the GFA prediction now covers the whole concentration space of an alloy which can comprise of combinations of phases at different compositions from the nominal one and is no longer limited to points with multiple entries in the database. To improve the descriptions of the reference states ψ_i^{gs} and competing structures ψ_i , we include pseudo-structures ϕ_i from neighbors around the target composition. Fig. 13 shows the formation of pseudo-structures based on entries in the aflow.org repositories. Different entries (green circles) are weighted to represent the target composition. Using aflow-CHULL [32] (see Section 6.1) the ground

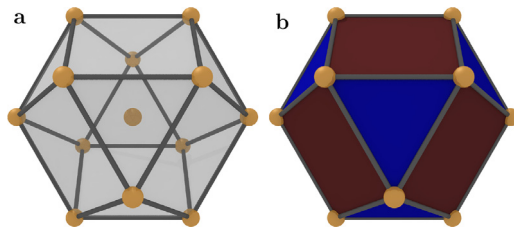


Fig. 14. The Atomic environment of Copper (fcc, cF4). The polyhedron code of this cuboctahedron (a) is 12^{2,2}, as it has 12 vertexes, each connected to two triangles and two quadrangles as highlighted in (b).

states for the investigated concentrations are identified. Based on their energetic distance to the ground state, the analysis will be limited to combinations that could occur at a typical glass transition temperature T_g .

The current implementation relies on two primary sources to predict a GFA score for a specific composition. The first one is the structural similarity of the competing entries. To capture the structural component, the atomic environments (AE) are constructed. The neighbors around each atom in an entry's unit-cell are utilized to form a collection of AEs. The categorization of neighbors is based on a distance histogram, as suggested by Brunner and Schwarzenbach [187], and later applied to AEs by Daams et al. [188]. A categorization scheme is used as the constructed AE are not directly comparable. Each AE is described by a polyhedron code created from the number of vertexes connected to a specific mix of facets [189]. An example is shown in Fig. 14.

This classification describes the structural information of an entry on a high level, and details such as distortions or size differences are not captured. Based on this description, the structural compatibility of entries is expressed by comparing the composition of the AE ensemble for each element type. The descriptor is zero if the structures have the same AEs and reaches a maximum when they have no AEs in common. All possible combinations in a limited area around a given stoichiometry will be combined to create the structural similarity $\bar{s}$ factor. Additionally, the function $f(|\phi_i|)$, describing the direct structural difference between an entry and the ground state, is used in the final representation of the GFA. Both structural descriptors include Gaussian distributed weights w_i assigned to each entry depending on the dimensionless distance in stoichiometry to the global stoichiometry.

The second source of information to predict a GFA is the formation enthalpy, which is captured by the exponential function $g(H_i)$. This function tends towards zero as the difference between the formation enthalpies of a considered entry and the ground state increases. Through $g(H_i)$, entries near the ground state have a bigger impact on the overall result.

Overall the GFA at a global stoichiometry $\{X\}$ is calculated as:

$$\chi_{\text{GFA}}(\{X\}) = \frac{100 \bar{s}^2 \sum_i f(|\phi_i|) g(H_i)}{\sum_i w_i},$$

where 100 is an arbitrary scaling factor. A detailed definition of the different segments is presented in Ref. [37].

The GFA module in *aflow++* can be invoked with:

```
aflow --gfa --alloy=CaCu
```

- `--alloy=alloy`: Sorted, case-sensitive string of the *alloy* system (e.g., CaCu).
- `--ae_file=file`: Optional argument, *file* containing pre-calculated atomic environments (e.g., AE_input.dat).
- `--cutoff_energy=cutoff`: Optional argument, is the formation enthalpy *cutoff* in eV/atom. Default is 0.05 (eV ~ 580 K).

This creates the following outputs:

- `standard output`: Information about the current calculation process.
- `GFA_entries.dat`: Contains the stoichiometries and formation enthalpies of the structures used in the GFA calculation.
- `GFA_alloy.dat`: Contains the calculated GFA for each point on the stoichiometry grid.
- `All_atomic_environments.dat`: Contains the atomic environments (one per species) for each entry used in the GFA calculation.

For the full set of options and additional information, see the *aflow*-GFA README:

```
aflow --readme=gfa
```

9. *aflow*-APE: The *aflow++* python environment

aflow++ is a powerful tool with various fully-automated workflows that can easily be integrated into custom code environments. Python is particularly important to support due to its popularity and because many machine learning frameworks are implemented in this language. To facilitate the usage of *aflow* with Python, the *aflow++* Python Environment (*aflow*-APE), has been developed and includes wrappers for *aflow*-SYM [31], *aflow*-CHULL [32], *aflow*-XtalFinder [14], and *aflow*-CCE [33].

aflow++ Python modules can be installed using the command

```
aflow --python_modules=[modules] -D directory
```

where *modules* are a comma-separated list and *directory* is where the modules are installed, e.g., the site-packages directory of the Python installation. Square brackets [...] indicate optional arguments; the brackets themselves are not part of the command. If no modules are given, *aflow* will install all available ones. A complete installation of *aflow* is required for them to run. They can also be installed into a virtual environment during the installation of *aflow* with the *install-aflow.sh* script by adding the `--venv` option. Some of these modules require additional Python packages. A list of dependencies can be accessed via

```
install-aflow.sh --pip_modules
```

They are automatically installed when using `--venv` with *install-aflow.sh*.

The *aflow*-SYM wrapper (module name: *aflow_sym*) provides functionality to calculate all *aflow++* symmetry groups (lattice point group, reciprocal lattice point group, crystallographic point group, the dual of the crystallographic point group, Patterson point group, factor group, space group, and atom-site point group), extended crystallographic data (edata), and space group data (sgdata). It supports magnetic moments as well. An example python script is provided here:

```
from aflow_sym import Symmetry
from pprint import pprint # pretty print JSON output

# instantiate class and specify path to aflow binary
sym = Symmetry(aflow_executable = '~/bin/aflow')

# indicate geometry file
input_file = 'test.poscar'

# calculate symmetry operators
output = sym.get_symmetry(input_file)
pprint(output)

# calculate space group data
output = sym.get_sgdata(input_file)
pprint(output)
```

```
# calculate extended crystallographic data
output = sym.get_edata(input_file)
pprint(output)
```

afLOW-CHULL consists of two different modules. `afLOW_chull` contains the `CHull` class, which calculates convex hulls, hull energies, distances to the convex hull, and stability criteria. An example python script is provided here:

```
from afLOW_chull import CHull
from pprint import pprint # pretty print JSON output

# instantiate class and specify path to afLOW binary
chull = CHull(afLOW_executable = '~/bin/afLOW')

# calculate PdPt hull and keep log file
alloy = 'PdPt'
output = chull.get_hull(alloy, options = '--keep=log')
pprint(output)

# calculate hull energy at x=0.5
output = chull.get_hull_energy(alloy, [0.5], options = '
    ↪ --keep=log')
pprint(output)
```

`afLOW_chull_plotter` provides a `Plotter` class that can create Jupyter notebooks with convex hull plots.

`afLOW-XtalFinder` is implemented via the `XtalFinder` class inside the `afLOW_xtal_finder` module. It can take a list of files or a directory as input and provide the structure comparison output. Single input files can also be compared to the `afLOW.org` repositories and prototype encyclopedia. Unique atom decorations can be obtained as well. An example python script is provided here:

```
from afLOW_xtal_finder import XtalFinder
from pprint import pprint # pretty print JSON output

# instantiate XtalFinder class
xtal_finder = XtalFinder()

# indicate geometry files
input_files = ['test.poscar1', 'test.poscar2']

# compare two structures
output = xtal_finder.compare_materials(input_files)
print('compare two structures via a material-type
    ↪ comparison (test.poscar1, test.poscar2):')
pprint(output)

# compare input to database entries
output = xtal_finder.compare2database('test.poscar1',
    ↪ '--properties=enthalpy_atom,natoms')
print('compare test.poscar1 to afLOW.org repositories:
    ↪ ')
pprint(output)

# compare input to afLOW prototypes
output = xtal_finder.compare2prototypes('test.poscar1
    ↪ ')
print('compare test.poscar1 to afLOW library of
    ↪ crystallographic prototypes:')
pprint(output)

# cast into afLOW prototype designation
output = xtal_finder.get_prototype_label('test.
    ↪ poscar2')
print('cast test.poscar2 in the afLOW prototype
    ↪ designation:')
pprint(output)
```

```
# get unique permutations for input
output = xtal_finder.get_unique_atom_decorations('
    ↪ test.poscar1')
print('calculate the unique atom decorations for test.
    ↪ poscar1:')
pprint(output)
```

The `afLOW-CCE` module (`afLOW_cce`) contains the `CCE` class and provides the same features as the C++ version of `afLOW++`, i.e., corrections to formation enthalpies, oxidation numbers, and coordination numbers around cations. An example python script is provided here:

```
from afLOW_cce import CCE
from pprint import pprint # pretty print JSON output

# define input arguments for CaTiO3 Pnma perovskite
# AB3C_oP20_62_c_cd_a
struct_file_path = 'test.POSCAR'
functionals = 'PBE,LDA,SCAN'
dft_formation_enthalpies = [-63.452, -72.084, -72.412]
oxidation_numbers = '2,2,2,2,-2,-2,-2,-2,-2,-2,-2,-2,
    ↪ -2,-2,-2,-2,4,4,4,4'

# instantiate class and specify path to afLOW binary
cce = CCE(afLOW_executable = '~/bin/afLOW')

# calculate corrections
output = cce.get_corrections(struct_file_path,
    ↪ functionals = functionals,
    ↪ enthalpies_formation_dft =
    ↪ dft_formation_enthalpies, oxidation_numbers =
    ↪ oxidation_numbers)
pprint(output)
```

With these wrappers, `afLOW++` can be seamlessly integrated into other Python workflows. Most functions return dictionaries, a basic built-in data type in Python. This allows the output to be further processed, e.g., to populate feature vectors in machine learning applications.

10. Summary

This article describes `afLOW++`, an interconnected collection of algorithms and workflows, written in C++, that have been developed to address the challenge of accelerated and autonomous materials' calculation and identifications. The article highlights the upgrades that have been developed since the original `afLOW++` report [50], and demonstrate their interoperability within the overall environment. The code, download/installation instructions and operation manuals describing all the features, are freely available at `afLOW.org`. Through ongoing innovation and implementation of robust descriptors and workflows, `afLOW++` continues to deliver valuable solutions as well as playing a role in accelerating the pace of automation in the materials community.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is open and available in <https://afLOW.org>.

Acknowledgments

The authors thank Adam Zettel, Asa Guest, Doug Wolfe, Don Brenner, Jon-Paul Maria, Bill Fahrenholtz, Douglas Hofmann, Christian Carbogno, Luca Ghiringhelli, Gus Hart, Eric Gossett, Cheryl Li, Harry Wang, Frisco Rose, Mana Rose, Max Brenner, William Schmitt, and Stuart Ki for fruitful discussions. Research sponsored by the Department of Defense Office of Naval Research (DOD-ONR N00014-21-1-2132, N00014-20-1-2525, N00014-20-1-2299, N00014-20-1-2200) and by the National Science Foundation (NSF NRT-HDR DGE-2022040). R.F. acknowledges support from the Alexander von Humboldt Foundation, Germany under the Feodor Lynen research fellowship. A.vdW acknowledges support from DOD-ONR (N00014-20-1-2225). C.T. acknowledges support from NSF (DMR-2219788).

Appendix A. aflow++ Installation

aflow++ was built to run on UNIX-based platforms and thus natively runs on Linux and macOS. It can be compiled from its monolithic source using the provided Makefile and without having to configure other libraries. The location of the executable can then be added to the `$PATH` variable to make it available everywhere for the user. Windows, on the other hand, cannot directly run aflow++ because of its different architecture and system APIs. Short of setting up a virtual machine running Linux, Windows users can compile aflow++ on the Windows Subsystem for Linux (WSL), which provides a GNU/Linux environment for multiple popular distributions, including Ubuntu and Debian [190,191].

To increase portability across platforms, compiling from source is a viable option [192]. Users with limited command-line experience can benefit from an automated installation script (`install-aflow.sh`). It supports many Linux distributions (including Ubuntu and Debian for the WSL), macOS, and Cygwin [193]. It can be downloaded at aflow.org/install-aflow [194]. The script automatically installs the dependencies needed to compile and run all features of aflow++. This includes external packages required for graphics. It then compiles the aflow++ source code and adds it to the `$PATH` variable so that the executable `aflow` can be used after the terminal is restarted. It can also install aflow++'s Python wrappers inside a virtual environment using the `--venv` option.

Some features of the installation can be customized. By default, dependencies for all aflow++ features are installed, which includes `pdflatex` used in some modules. These plotting packages require considerable disk space, and often cannot be installed by individual users on a shared system. They can be skipped entirely by executing the script as:

```
install-aflow.sh --slim
```

Other opportunities to customize the installation include selecting the location of the aflow++ binary by using the option `--ULB=path`, where `path` is the desired path for the aflow++ binary. Similarly, the location of the aflow++ source directory and the installed virtual environment can be set by using `--AWD=path` and `--venv=path`, respectively.

Appendix B. The aflow.rc configuration file

The `aflow.rc` is the configuration file defining the default settings for running within the aflow++ environment. The file emulates the `bashrc` script that runs when initializing an interactive environment in `bash` (Bourne again shell). The file is read upon running the `aflow` binary, with settings that can be overridden by flags passed through the command line or those set in the `aflow.in` file, allowing for calculation-specific customization. Upon running `aflow`, a fresh `aflow.rc` file is created in `$HOME` if one is not already

present. Just like the `bashrc` file, the `aflow.rc` file is hidden (i.e., `$HOME/.aflow.rc`).

Tunable settings include the compression algorithm, output file names, MPI settings, paths for binaries, databases, module-specific settings, and machine settings. Users of VASP might consider configuring the following settings for their machine:

```
// DEFAULT GENERIC MPI
MPI_START_DEFAULT="ulimit -s unlimited"
MPI_STOP_DEFAULT=""
MPI_COMMAND_DEFAULT="mpirun -np"
MPI_NCPUS_DEFAULT=4
MPI_NCPUS_MAX=4

// DEFAULTS BINARY
DEFAULT_VASP_BIN="vasp46s"
DEFAULT_VASP_MPI_BIN="mpivasp46s"
DEFAULT_VASP5_BIN="vasp_std"
DEFAULT_VASP5_MPI_BIN="vasp_std"

// DEFAULTS POTCARS
DEFAULT_VASP_POTCAR_DIRECTORIES="" /src/vasp,/home/
↪ Tools/src/vasp"
DEFAULT_VASP_POTCAR_DATE="current"
DEFAULT_VASP_POTCAR_SUFFIX="/POTCAR"
DEFAULT_VASP_POTCAR_DIR_POT_LDA="pot_LDA"
DEFAULT_VASP_POTCAR_DIR_POT_GGA="pot_GGA"
DEFAULT_VASP_POTCAR_DIR_POT_PBE="pot_PBE"
DEFAULT_VASP_POTCAR_DIR_POTPAW_LDA="potpaw_LDA"
DEFAULT_VASP_POTCAR_DIR_POTPAW_GGA="potpaw_GGA"
DEFAULT_VASP_POTCAR_DIR_POTPAW_PBE="potpaw_PBE"
DEFAULT_VASP_POTCAR_DIR_POTPAW_LDA_KIN="potpaw_LDA
↪ .54"
DEFAULT_VASP_POTCAR_DIR_POTPAW_PBE_KIN="potpaw_PBE
↪ .54"
```

`MPI_START_DEFAULT` and `MPI_STOP_DEFAULT` are commands to run before and after the MPI program. For example, running `ulimit -s unlimited` allows VASP to access as much stack memory as it needs. `MPI_COMMAND_DEFAULT` together with `MPI_NCPUS_DEFAULT` and `MPI_NCPUS_MAX` define the MPI portion of the VASP command, e.g., `mpirun -np 4`. `MPI_COMMAND_DEFAULT` is overridden in the `aflow.in` with the following setting: `[AFLOW_MODE_MPI_MODE] COMMAND="mpirun -np"`. `MPI_NCPUS_DEFAULT` is chosen if no overriding `NCPUS` option is provided on either the command line (`aflow --run --ncpus=8`) or in the `aflow.in` (`[AFLOW_MODE_MPI_MODE] NCPUS=8`), and `MPI_NCPUS_MAX` is chosen if the maximum number of cores are requested via the command line (`--ncpus=max`) or in the `aflow.in` (`[AFLOW_MODE_MPI_MODE] NCPUS=MAX`).

The VASP binary name is specified with the `DEFAULT_VASP_BIN` variable, with corresponding serial and MPI variants (`DEFAULT_VASP_MPI_BIN`). The serial and MPI variants are overridden in the `aflow.in` with `[AFLOW_MODE_BINARY]=vasp46s` and `[AFLOW_MODE_MPI_MODE] BINARY="mpivasp46s"`, respectively. The default can be either a VASP4 or VASP5 binary. There are some cases where the VASP5 binary is required by the workflow (e.g., calculation of force constants using linear-response or to determine longitudinal optic and transverse optical (LO-TO) lattice vibration splitting). aflow++ tries to determine the VASP version automatically and, for these cases, changes the settings and formats to run with VASP5 (`DEFAULT_VASP5_BIN` serial and MPI variants). These binaries should be made accessible in the environment of the compute-node through the `$PATH`. Together with the MPI settings, these variables define the full MPI VASP command to be executed by aflow++:

```
mpirun -np 4 mpivasp46s > vasp.out
```

This can be checked in the calculation's `LOCK` file, looking for the line that starts with:


```
00000 MESSAGE Executing: command
```

Proper organization of the pseudopotential files is critical for enabling *afLOW++* to find and use them automatically. *afLOW++* breaks the subdirectory structure into different variables. For example, the following path `/home/Tools/src/vasp/potpaw_PBE/current/Mn_pv/POTCAR` can be constructed to find the manganese pseudopotential file calculated with the PBE functional [46] using the projector-augmented-wave formalism (PAW) [195]. `DEFAULT_VASP_POTCAR_DIRECTORIES` is a comma-separated list of paths to check for the pseudopotential files. It resembles the `$PATH` environment variable in that it will check each path sequential until a match is found. The level of theory, formalization, and components are specified in the `afLOW.in`:

```
[VASP_FORCE_OPTION] AUTO_PSEUDOPOTENTIALS=mode
[VASP_POTCAR_MODE_IMPLICIT]
[VASP_POTCAR_FILE] Mn
```

mode can be any of the following: `pot_LDA`, `pot_GGA`, `potpaw_LDA`, `potpaw_GGA`, `potpaw_PBE`, `potpaw_LDA_KIN`, `potpaw_PBE_KIN`, where LDA refers to the local density approximation, and the KIN variants refer to the pseudopotentials released with VASP5.4 that include information on the kinetic energy density of the core-electrons. The *mode* chosen corresponds to one of the pseudopotential subdirectories: `DEFAULT_VASP_POTCAR_DIR_MODE`. The `pv` suffix in `Mn_pv` refers to the treatment of *p* semi-core states as valence states, as specified in the VASP Wiki: https://www.vasp.at/wiki/index.php/Available_PAW_potentials. The selection of the right pseudopotential file can be checked in the calculation's LOCK file, looking for the line that starts with:

```
00000 MESSAGE POTCAR FILE: Found potcar FilePotcar=path
```

To keep track of different pseudopotentials released with new versions of VASP, *afLOW++* identifies each reference with its MD5 hash [13, 196] and appends this information to the bottom of the `afLOW.in`:

```
[VASP_POTCAR_AUID] 99be850476e2dfb3
```

References

- [1] M. Esters, C. Oses, S. Divilov, H. Eckert, R. Friedrich, D. Hicks, M.J. Mehl, F. Rose, A. Smolyanyuk, A. Calzolari, X. Campilongo, C. Toher, S. Curtarolo, *afLOW.org*: A web ecosystem of databases, software and tools, *Comput. Mater. Sci.* 216 (2023) 111808, <http://dx.doi.org/10.1016/j.commatsci.2022.111808>.
- [2] W. Setyawan, S. Curtarolo, AFLOWLIB: *Ab-initio* electronic structure library database, 2011, <https://www.afLOW.org>.
- [3] C. Oses, C. Toher, S. Curtarolo, Data-driven design of inorganic materials with the Automatic Flow Framework for Materials Discovery, *MRS Bull.* 43 (2018) 670–675, <http://dx.doi.org/10.1557/mrs.2018.207>.
- [4] S. Sanvito, C. Oses, J. Xue, A. Tiwari, M. Žic, T. Archer, P. Tozcan, M. Venkatesan, J.M.D. Coey, S. Curtarolo, Accelerated discovery of new magnets in the Heusler alloy family, *Sci. Adv.* 3 (2017) e1602241, <http://dx.doi.org/10.1126/sciadv.1602241>.
- [5] S. Sanvito, M. Žic, J. Nelson, T. Archer, C. Oses, S. Curtarolo, Machine learning and high-throughput approaches to magnetism, in: W. Andreoni, S. Yip (Eds.), *Handbook of Materials Modeling: Applications: Current and Emerging Materials*, Springer International Publishing, Cham, Switzerland, 2018, pp. 1–23, http://dx.doi.org/10.1007/978-3-319-50257-1_108-1.
- [6] C. Nyshadham, C. Oses, J.E. Hansen, I. Takeuchi, S. Curtarolo, G.L.W. Hart, A computational high-throughput search for new ternary superalloys, *Acta Mater.* 122 (2017) 438–447, <http://dx.doi.org/10.1016/j.actamat.2016.09.017>.
- [7] F.L.R. Tirado, J.P. Toinin, D.C. Dunand, $\gamma + \gamma'$ Microstructures in the Co-Ta-V and Co-Nb-V ternary systems, *Acta Mater.* 151 (2018) 137–148, <http://dx.doi.org/10.1016/j.actamat.2018.03.057>.
- [8] P. Sarker, T. Harrington, C. Toher, C. Oses, M. Samiee, J.-P. Maria, D.W. Brenner, K.S. Vecchio, S. Curtarolo, High-entropy high-hardness metal carbides discovered by entropy descriptors, *Nature Commun.* 9 (2018) 4980, <http://dx.doi.org/10.1038/s41467-018-07160-7>.
- [9] T.J. Harrington, J. Gild, P. Sarker, C. Toher, C.M. Rost, O.F. Dippo, C. McElfresh, K. Kaufmann, E. Marin, L. Borowski, P.E. Hopkins, J. Luo, S. Curtarolo, D.W. Brenner, K.S. Vecchio, Phase stability and mechanical properties of novel high entropy transition metal carbides, *Acta Mater.* 166 (2019) 271–280, <http://dx.doi.org/10.1016/j.actamat.2018.12.054>.
- [10] K. Kaufmann, D. Maryanovsky, W.M. Mellor, C. Zhu, A.S. Rosengarten, T.J. Harrington, C. Oses, C. Toher, S. Curtarolo, K.S. Vecchio, Discovery of high-entropy ceramics via machine learning, *npj Comput. Mater.* 6 (2020) 42, <http://dx.doi.org/10.1038/s41524-020-0317-6>.
- [11] A. Calzolari, C. Oses, C. Toher, M. Esters, X. Campilongo, S.P. Stepanoff, D.E. Wolfe, S. Curtarolo, Plasmonic high-entropy carbides, *Nat. Commun.* 13 (2022) <http://dx.doi.org/10.1038/s41467-022-33497-1>.
- [12] A.G. Kusne, H. Yu, C. Wu, H. Zhang, J. Hattrick-Simpers, B. DeCost, S. Sarker, C. Oses, C. Toher, S. Curtarolo, A.V. Davydov, R. Agarwal, L.A. Bendersky, M. Li, A. Mehta, I. Takeuchi, On-the-fly closed-loop materials discovery via Bayesian active learning, *Nature Commun.* 11 (2020) 5966, <http://dx.doi.org/10.1038/s41467-020-19597-w>.
- [13] C.E. Calderon, J.J. Plata, C. Toher, C. Oses, O. Levy, M. Fornari, A. Natan, M.J. Mehl, G.L.W. Hart, M. Buongiorno Nardelli, S. Curtarolo, The AFLOW standard for high-throughput materials science calculations, *Comput. Mater. Sci.* 108 (Part A) (2015) 233–238, <http://dx.doi.org/10.1016/j.commatsci.2015.07.019>.
- [14] D. Hicks, C. Toher, D.C. Ford, F. Rose, C. De Santo, O. Levy, M.J. Mehl, S. Curtarolo, AFLOW-XtalFinder: a reliable choice to identify crystalline prototypes, *npj Comput. Mater.* 7 (2021) 30, <http://dx.doi.org/10.1038/s41524-020-00483-4>.
- [15] M.J. Mehl, D. Hicks, C. Toher, O. Levy, R.M. Hanson, G.L.W. Hart, S. Curtarolo, The AFLOW library of crystallographic prototypes: Part 1, *Comput. Mater. Sci.* 136 (2017) S1–S828, <http://dx.doi.org/10.1016/j.commatsci.2017.01.017>.
- [16] D. Hicks, M.J. Mehl, E. Gossett, C. Toher, O. Levy, R.M. Hanson, G.L.W. Hart, S. Curtarolo, The AFLOW library of crystallographic prototypes: Part 2, *Comput. Mater. Sci.* 161 (2019) S1–S1011, <http://dx.doi.org/10.1016/j.commatsci.2018.10.043>.
- [17] D. Hicks, M.J. Mehl, M. Esters, C. Oses, O. Levy, G.L.W. Hart, C. Toher, S. Curtarolo, The AFLOW library of crystallographic prototypes: Part 3, *Comput. Mater. Sci.* 199 (2021) 110450, <http://dx.doi.org/10.1016/j.commatsci.2021.110450>.
- [18] G. Kresse, J. Hafner, *Ab initio* molecular dynamics for liquid metals, *Phys. Rev. B* 47 (1993) 558–561, <http://dx.doi.org/10.1103/PhysRevB.47.558>.
- [19] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. Dal Corso, S. de Gironcoli, P. Delugas, R.A. DiStasio Jr., A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N.L. Nguyen, H.-V. Nguyen, A. Otero-de-la-Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A.P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, S. Baroni, Advanced capabilities for materials modelling with Quantum ESPRESSO, *J. Phys.: Condens. Matter* 29 (2017) 465901, <http://dx.doi.org/10.1088/1361-648X/aa8f79>.
- [20] J. Gonze, J.M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, P. Ghosez, J.Y. Raty, D.C. Allan, First-principles computation of material properties: the ABINIT software project, *Comput. Mater. Sci.* 25 (2002) 478–492, [http://dx.doi.org/10.1016/S0927-0256\(02\)00325-7](http://dx.doi.org/10.1016/S0927-0256(02)00325-7).
- [21] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, M. Scheffler, *Ab initio* molecular simulations with numeric atom-centered orbitals, *Comput. Phys. Comm.* 180 (2009) 2175–2196, <http://dx.doi.org/10.1016/j.cpc.2009.06.022>.
- [22] The ELK code, 2020, <http://elk.sourceforge.net/> (accessed January 6, 2021).
- [23] A. van de Walle, M. Asta, G. Ceder, The alloy theoretic automated toolkit: A user guide, *CALPHAD* 26 (2002) 539–553, [http://dx.doi.org/10.1016/S0364-5916\(02\)80006-2](http://dx.doi.org/10.1016/S0364-5916(02)80006-2).
- [24] A.R. Supka, T.E. Lyons, L.S.I. Liyanage, P. D'Amico, R. Al Rahal Al Orabi, S. Mahatara, P. Gopal, C. Toher, D. Ceresoli, A. Calzolari, S. Curtarolo, M. Buongiorno Nardelli, M. Fornari, AFLOWx: A minimalist approach to high-throughput *ab initio* calculations including the generation of tight-binding hamiltonians, *Comput. Mater. Sci.* 136 (2017) 76–84, <http://dx.doi.org/10.1016/j.commatsci.2017.03.055>.
- [25] L.A. Agapito, S. Curtarolo, M. Buongiorno Nardelli, Reformulation of DFT+*U* as a pseudohybrid hubbard density functional for accelerated materials discovery, *Phys. Rev. X* 5 (2015) 011006, <http://dx.doi.org/10.1103/PhysRevX.5.011006>.
- [26] M. Buongiorno Nardelli, F.T. Cerasoli, M. Costa, S. Curtarolo, R.D. Gennaro, M. Fornari, L. Liyanage, A.R. Supka, H. Wang, PAOFlow: A utility to construct and operate on *ab initio* Hamiltonians from the projections of electronic wavefunctions on atomic orbital bases, including characterization of topological materials, *Comput. Mater. Sci.* 143 (2018) 462–472, <http://dx.doi.org/10.1016/j.commatsci.2017.11.034>.
- [27] F.T. Cerasoli, A.R. Supka, A. Jayaraj, M. Costa, I. Siloi, J. Sławińska, S. Curtarolo, M. Fornari, D. Ceresoli, M. Buongiorno Nardelli, Advanced modeling of materials with PAOFlow 2.0: New features and software design, *Comput.*

- Mater. Sci. 200 (2021) 110828, <http://dx.doi.org/10.1016/j.commatsci.2021.110828>.
- [28] J.J. Plata, P. Nath, D. Usanmaz, J. Carrete, C. Toher, M. de Jong, M.D. Asta, M. Fornari, M. Buongiorno Nardelli, S. Curtarolo, An efficient and accurate framework for calculating lattice thermal conductivity of solids: AFLOW-AAPL Automatic Anharmonic Phonon Library, *npj Comput. Mater.* 3 (2017) 45, <http://dx.doi.org/10.1038/s41524-017-0046-7>.
- [29] C. Toher, C. Oses, J.J. Plata, D. Hicks, F. Rose, O. Levy, M. de Jong, M. Asta, M. Fornari, M. Buongiorno Nardelli, S. Curtarolo, Combining the AFLOW GIBBS and elastic libraries to efficiently and robustly screen thermomechanical properties of solids, *Phys. Rev. Mater.* 1 (2017) 015401, <http://dx.doi.org/10.1103/PhysRevMaterials.1.015401>.
- [30] C. Toher, J.J. Plata, O. Levy, M. de Jong, M. Asta, M. Buongiorno Nardelli, S. Curtarolo, High-throughput computational screening of thermal conductivity, Debye temperature, and Grüneisen parameter using a quasiharmonic Debye model, *Phys. Rev. B* 90 (2014) 174107, <http://dx.doi.org/10.1103/PhysRevB.90.174107>.
- [31] D. Hicks, C. Oses, E. Gossett, G. Gomez, R.H. Taylor, C. Toher, M.J. Mehl, O. Levy, S. Curtarolo, AFLOW-SYM: platform for the complete, automatic and self-consistent symmetry analysis of crystals, *Acta Crystallogr. Sect. A* 74 (2018) 184–203, <http://dx.doi.org/10.1107/S2053273318003066>.
- [32] C. Oses, E. Gossett, D. Hicks, F. Rose, M.J. Mehl, E. Perim, I. Takeuchi, S. Sanvito, M. Scheffler, Y. Lederer, O. Levy, C. Toher, S. Curtarolo, AFLOW-CHULL: Cloud-oriented platform for autonomous phase stability analysis, *J. Chem. Inf. Model.* 58 (2018) 2477–2490, <http://dx.doi.org/10.1021/acs.jcim.8b00393>.
- [33] R. Friedrich, M. Esters, C. Oses, S. Ki, M.J. Brenner, D. Hicks, M.J. Mehl, C. Toher, S. Curtarolo, Automated coordination corrected enthalpies with AFLOW-CCE, *Phys. Rev. Mater.* 5 (2021) 0438031, <http://dx.doi.org/10.1103/PhysRevMaterials.5.043803>.
- [34] M. Esters, C. Oses, D. Hicks, M.J. Mehl, M. Jahnátek, M.D. Hossain, J.-P. Maria, D.W. Brenner, C. Toher, S. Curtarolo, Settling the matter of the role of vibrations in the stability of high-entropy carbides, *Nature Commun.* 12 (2021) 5747, <http://dx.doi.org/10.1038/s41467-021-25979-5>.
- [35] R. Friedrich, D. Usanmaz, C. Oses, A. Supka, M. Fornari, M. Buongiorno Nardelli, C. Toher, S. Curtarolo, Coordination corrected *ab initio* formation enthalpies, *npj Comput. Mater.* 5 (2019) 59, <http://dx.doi.org/10.1038/s41524-019-0192-1>.
- [36] E. Perim, D. Lee, Y. Liu, C. Toher, P. Gong, Y. Li, W.N. Simmons, O. Levy, J.J. Vlassak, J. Schroers, S. Curtarolo, Spectral descriptors for bulk metallic glasses based on the thermodynamics of competing crystalline phases, *Nature Commun.* 7 (2016) 12315, <http://dx.doi.org/10.1038/ncomms12315>.
- [37] D.C. Ford, D. Hicks, C. Oses, C. Toher, S. Curtarolo, Metallic glasses for biodegradable implants, *Acta Mater.* 176 (2019) 297–305, <http://dx.doi.org/10.1016/j.actamat.2019.07.008>.
- [38] K. Yang, C. Oses, S. Curtarolo, Modeling off-stoichiometry materials with a high-throughput *Ab-Initio* approach, *Chem. Mater.* 28 (2016) 6484–6492, <http://dx.doi.org/10.1021/acs.chemmater.6b01449>.
- [39] Y. Lederer, C. Toher, K.S. Vecchio, S. Curtarolo, The search for high entropy alloys: A high-throughput *ab-initio* approach, *Acta Mater.* 159 (2018) 364–383, <http://dx.doi.org/10.1016/j.actamat.2018.07.042>.
- [40] P. Nath, J.J. Plata, D. Usanmaz, R. Al Rahal Al Arabi, M. Fornari, M. Buongiorno Nardelli, C. Toher, S. Curtarolo, High-throughput prediction of finite-temperature properties using the quasi-harmonic approximation, *Comput. Mater. Sci.* 125 (2016) 82–91, <http://dx.doi.org/10.1016/j.commatsci.2016.07.043>.
- [41] P. Nath, D. Usanmaz, D. Hicks, C. Oses, M. Fornari, M.B. Nardelli, C. Toher, S. Curtarolo, AFLOW-QHA3P: Robust and automated method to compute thermodynamic properties of solids, *Phys. Rev. Mater.* 3 (2019) 073801, <http://dx.doi.org/10.1103/PhysRevMaterials.3.073801>.
- [42] M. Esters, A. Smolyanyuk, C. Oses, D. Hicks, S. Divilov, H. Eckert, X. Campilongo, C. Toher, S. Curtarolo, QH-POCC: taming tiling entropy in thermal expansion calculations of disordered materials, 2022, submitted for publication.
- [43] S. Curtarolo, Coarse-Graining and Data Mining Approaches to the Prediction of Structures and their Dynamics (Ph.D. Thesis), MIT, 2003.
- [44] S. Curtarolo, D. Morgan, K.A. Persson, J. Rodgers, G. Ceder, Predicting crystal structures with data mining of quantum calculations, *Phys. Rev. Lett.* 91 (2003) 135503, <http://dx.doi.org/10.1103/PhysRevLett.91.135503>.
- [45] S. Curtarolo, D. Morgan, G. Ceder, Accuracy of *ab initio* methods in predicting the crystal structures of metals: A review of 80 binary alloys, *CALPHAD* 29 (2005) 163–211, <http://dx.doi.org/10.1016/j.calphad.2005.01.002>.
- [46] J.P. Perdew, K. Burke, M. Ernzerhof, Generalized gradient approximation made simple, *Phys. Rev. Lett.* 77 (1996) 3865–3868, <http://dx.doi.org/10.1103/PhysRevLett.77.3865>.
- [47] S. Curtarolo, G.L.W. Hart, M. Buongiorno Nardelli, N. Mingo, S. Sanvito, O. Levy, The high-throughput highway to computational materials design, *Nature Mater.* 12 (2013) 191–201, <http://dx.doi.org/10.1038/nmat3568>.
- [48] W. Setyawan, S. Curtarolo, High-throughput electronic band structure calculations: Challenges and tools, *Comput. Mater. Sci.* 49 (2010) 299–312, <http://dx.doi.org/10.1016/j.commatsci.2010.05.010>.
- [49] M. de Jong, W. Chen, T. Angsten, A. Jain, R. Notestine, A. Gamst, M. Sluiter, C.K. Ande, S. van der Zwaag, J.J. Plata, C. Toher, S. Curtarolo, G. Ceder, K.A. Persson, M.D. Asta, Charting the complete elastic properties of inorganic crystalline compounds, *Sci. Data* 2 (2015) 150009, <http://dx.doi.org/10.1038/sdata.2015.9>.
- [50] S. Curtarolo, W. Setyawan, G.L.W. Hart, M. Jahnátek, R.V. Chepulskii, R.H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M.J. Mehl, H.T. Stokes, D.O. Demchenko, D. Morgan, AFLOW: An automatic framework for high-throughput materials discovery, *Comput. Mater. Sci.* 58 (2012) 218–226, <http://dx.doi.org/10.1016/j.commatsci.2012.02.005>.
- [51] Office of Science and Technology Policy, White House, Materials Genome Initiative for Global Competitiveness, 2011, <http://www.whitehouse.gov/mgi>.
- [52] G. Kresse, J. Furthmüller, Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set, *Phys. Rev. B* 54 (1996) 11169–11186, <http://dx.doi.org/10.1103/PhysRevB.54.11169>.
- [53] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G.L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougousis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sciauzero, A.P. Seitsonen, A. Smogunov, P. Umari, R.M. Wentzcovitch, QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials, *J. Phys.: Condens. Matter* 21 (2009) 395502, <http://dx.doi.org/10.1088/0953-8984/21/39/395502>.
- [54] S.R. Hall, F.H. Allen, I.D. Brown, The crystallographic information file (CIF): a New Standard Archive File for Crystallography, *Acta Crystallogr. Sect. A* 47 (1991) 655–685, <http://dx.doi.org/10.1107/S010876739101067X>.
- [55] G. Kresse, D. Joubert, From ultrasoft pseudopotentials to the projector augmented-wave method, *Phys. Rev. B* 59 (1999) 1758–1775, <http://dx.doi.org/10.1103/PhysRevB.59.1758>.
- [56] VASP, The VASP Wiki, 2022, https://www.vasp.at/wiki/index.php/The_VASP_Manual (accessed June 20, 2022).
- [57] M. Methfessel, A.T. Paxton, High-precision sampling for Brillouin-zone integration in metals, *Phys. Rev. B* 40 (1989) 3616–3621, <http://dx.doi.org/10.1103/PhysRevB.40.3616>.
- [58] F. Rose, C. Toher, E. Gossett, C. Oses, M. Buongiorno Nardelli, M. Fornari, S. Curtarolo, AFLUX: The LUX materials search API for the AFLOW data repositories, *Comput. Mater. Sci.* 137 (2017) 362–370, <http://dx.doi.org/10.1016/j.commatsci.2017.04.036>.
- [59] P.E. Blöchl, O. Jepsen, O.K. Andersen, Improved tetrahedron method for Brillouin-zone integrations, *Phys. Rev. B* 49 (1994) 16223–16233, <http://dx.doi.org/10.1103/PhysRevB.49.16223>.
- [60] J.P. Perdew, Density functional theory and the band gap problem, *Int. J. Quantum Chem.* 28 (1985) 497–523, <http://dx.doi.org/10.1002/qua.560280846>.
- [61] A.I. Liechtenstein, V.I. Anisimov, J. Zaanen, Density-functional theory and strong interactions: Orbital ordering in Mott-Hubbard insulators, *Phys. Rev. B* 52 (1995) R5467–R5470, <http://dx.doi.org/10.1103/PhysRevB.52.R5467>.
- [62] S.L. Dudarev, G.A. Botton, S.Y. Savrasov, C.J. Humphreys, A.P. Sutton, Electron-energy-loss spectra and the structural stability of nickel oxide: An LSDA+*U* study, *Phys. Rev. B* 57 (1998) 1505–1509, <http://dx.doi.org/10.1103/PhysRevB.57.1505>.
- [63] B. Liu, The simultaneous expansion method for the iterative solution of several of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices, in: C.B. Moler, I. Shavitt (Eds.), *Numerical Algorithms in Chemistry: Algebraic Methods: Report on the Workshop*, Aug. 9–11, 1978, Lawrence Berkeley Laboratory, University of California, 1978, pp. 49–53.
- [64] E.R. Davidson, Matrix eigenvector methods, in: G.H.F. Diercksen, S. Wilson (Eds.), *Methods in Computational Molecular Physics*, Springer Netherlands, Dordrecht, 1983, pp. 95–113, http://dx.doi.org/10.1007/978-94-009-7200-1_4.
- [65] P.Q. Nguyen, D. Stehlé, Low-dimensional lattice basis reduction revisited, *ACM Trans. Algorithms* 5 (2009) 1–48, <http://dx.doi.org/10.1145/1597036.1597050>.
- [66] P.Q. Nguyen, D. Stehlé, Low-dimensional lattice basis reduction revisited, in: D. Buell (Ed.), *Algorithmic Number Theory*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 338–357, http://dx.doi.org/10.1007/978-3-540-24847-7_26.
- [67] T. Hahn (Ed.), *International Tables of Crystallography. Volume A: Space-Group Symmetry*, Kluwer Academic publishers, International Union of Crystallography, Chester, England, 2002.
- [68] B. Gruber, The relationship between reduced cells in a general Bravais lattice, *Acta Crystallogr. Sect. A* 29 (1973) 433–440, <http://dx.doi.org/10.1107/S0567739473001063>.
- [69] P. Niggli, W. Wien, F. Harms, *Handbuch der Experimentalphysik*, Vol. 7, Akademische Verlagsgesellschaft, 1928.
- [70] M. Hloucha, U.K. Deiters, Fast coding of the minimum image convention, *Mol. Simul.* 20 (1998) 239–244, <http://dx.doi.org/10.1080/08927029808024180>.
- [71] M. Nespolo, B. Souvignier, Point groups in crystallography, *Z. Kristallogr.* 224 (2009) 127–136, <http://dx.doi.org/10.1524/zkri.2009.1107>.
- [72] B.A. DeAngelis, R.E. Newnham, W.B. White, Factor group analysis of the vibrational spectra of crystals: A review and consolidation, *Am. Mineral.* 57 (1972) 255–268.

- [73] R. Gilmore, *Lie Groups, Physics, and Geometry*, Cambridge University Press, 2008.
- [74] H.T. Stokes, D.M. Hatch, FINDSYM: Program for identifying the space group symmetry of a crystal, *J. Appl. Crystallogr.* 38 (2005) 237–238, <http://dx.doi.org/10.1107/S0021889804031528>.
- [75] A.L. Spek, Single-crystal structure validation with the program PLATON, *J. Appl. Crystallogr.* 36 (2003) 7–13, <http://dx.doi.org/10.1107/S0021889802022112>.
- [76] Spglib, 2017, <https://atztogo.github.io/spglib/>.
- [77] M.-O. Lenz, T.A.R. Purcell, D. Hicks, S. Curtarolo, M. Scheffler, C. Carbogno, Parametrically constrained geometry relaxations for high-throughput materials science, *npj Comput. Mater.* 5 (2019) 123, <http://dx.doi.org/10.1038/s41524-019-0254-4>.
- [78] L.L. Boyle, J.E. Lawrenson, The origin dependence of wyckoff site description of a crystal structure, *Acta Crystallogr. Sect. A* 29 (1973) 353–357, <http://dx.doi.org/10.1107/S056773947300094X>.
- [79] E. Koch, W. Fischer, Automorphismengruppen von raumgruppen und die zuordnung von punktlagen zu konfigurationslagen, *Acta Crystallogr. Sect. A* 31 (1975) 88–95, <http://dx.doi.org/10.1107/S0567739475000174>.
- [80] J. Lima-de-Faria, E. Hellner, F. Liebau, E. Makovicky, E. Parthé, Nomenclature of inorganic structure types. Report of the international union of crystallography commission on crystallographic nomenclature subcommittee on the nomenclature of inorganic structure types, *Acta Crystallogr. Sect. A* 46 (1990) 1–11, <http://dx.doi.org/10.1107/S0108767389008834>.
- [81] N. Zimmermann, A. Jain, Local structure order parameters and site fingerprints for quantification of coordination environment and crystal structure similarity, *RSC Adv.* 10 (2020) 6063–6081, <http://dx.doi.org/10.1039/C9RA07755C>.
- [82] H. Burzlaff, Y. Malinovsky, A procedure for the classification of non-organic crystal structures. I. Theoretical background, *Acta Crystallogr. Sect. A* 53 (1997) 217–224, <http://dx.doi.org/10.1107/S0108767396013852>.
- [83] W.B. Pearson, *The Crystal Chemistry and Physics of Metals and Alloys*, Wiley-Interscience, 1972.
- [84] E. Parthé, *Elements of Inorganic Structural Chemistry: A Course on Selected Topics*, K. Sutter Parthé, Petit-Lancy, Switzerland, 1990.
- [85] R.H. Taylor, F. Rose, C. Toher, O. Levy, K. Yang, M. Buongiorno Nardelli, S. Curtarolo, A RESTful API for exchanging materials data in the AFLOWLIB.org consortium, *Comput. Mater. Sci.* 93 (2014) 178–192, <http://dx.doi.org/10.1016/j.commatsci.2014.05.014>.
- [86] L.A. Agapito, A. Ferretti, A. Calzolari, S. Curtarolo, M. Buongiorno Nardelli, Effective and accurate representation of extended Bloch states on finite Hilbert spaces, *Phys. Rev. B* 88 (2013) 165127, <http://dx.doi.org/10.1103/PhysRevB.88.165127>.
- [87] L.A. Agapito, S. Ismail-Beigi, S. Curtarolo, M. Fornari, M. Buongiorno Nardelli, Accurate tight-binding Hamiltonian matrices from *ab initio* calculations: Minimal basis sets, *Phys. Rev. B* 93 (2016) 035104, <http://dx.doi.org/10.1103/PhysRevB.93.035104>.
- [88] L.A. Agapito, M. Fornari, D. Ceresoli, A. Ferretti, S. Curtarolo, M. Buongiorno Nardelli, Accurate tight-binding Hamiltonians for two-dimensional and layered materials, *Phys. Rev. B* 93 (2016) 125137, <http://dx.doi.org/10.1103/PhysRevB.93.125137>.
- [89] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R.H. Taylor, L.J. Nelson, G.L.W. Hart, S. Sanvito, M. Buongiorno Nardelli, N. Mingo, O. Levy, AFLOWLIB.ORG: A distributed materials properties repository from high-throughput *ab initio* calculations, *Comput. Mater. Sci.* 58 (2012) 227–235, <http://dx.doi.org/10.1016/j.commatsci.2012.02.002>.
- [90] A.A. Mostofi, J.R. Yates, G. Pizzi, Y.-S. Lee, I. Souza, D. Vanderbilt, N. Marzari, An updated version of wannier90: A tool for obtaining maximally-localised Wannier functions, *Comput. Phys. Comm.* 185 (2014) 2309–2310, <http://dx.doi.org/10.1016/j.cpc.2014.05.003>.
- [91] P. D'Amico, L.A. Agapito, A. Catellani, A. Ruini, S. Curtarolo, M. Fornari, M. Buongiorno Nardelli, A. Calzolari, Accurate *ab initio* tight-binding Hamiltonians: Effective tools for electronic transport and optical spectroscopy from first principles, *Phys. Rev. B* 94 (2016) 165166, <http://dx.doi.org/10.1103/PhysRevB.94.165166>.
- [92] A. Jayaraj, I. Siloi, M. Fornari, M. Buongiorno Nardelli, Relaxation time approximations in PAOFlow 2.0, *Sci. Rep.* 12 (2022) 4993, <http://dx.doi.org/10.1038/s41598-022-08931-5>.
- [93] O. Levy, G.L.W. Hart, S. Curtarolo, Uncovering compounds by synergy of cluster expansion and high-throughput methods, *J. Am. Chem. Soc.* 132 (2010) 4830–4833, <http://dx.doi.org/10.1021/ja9105623>.
- [94] O. Levy, G.L.W. Hart, S. Curtarolo, Hafnium binary alloys from experiments and first principles, *Acta Mater.* 58 (2010) 2887–2897, <http://dx.doi.org/10.1016/j.actamat.2010.01.017>.
- [95] R.H. Taylor, S. Curtarolo, G.L.W. Hart, Ordered magnesium-lithium alloys: First-principles predictions, *Phys. Rev. B* 81 (2010) 024112, <http://dx.doi.org/10.1103/PhysRevB.81.024112>.
- [96] O. Levy, M. Jahnátek, R.V. Chepulsii, G.L.W. Hart, S. Curtarolo, Ordered structures in rhodium binary alloys from first-principles calculations, *J. Am. Chem. Soc.* 133 (2011) 158–163, <http://dx.doi.org/10.1021/ja1091672>.
- [97] M. Jahnátek, O. Levy, G.L.W. Hart, L.J. Nelson, R.V. Chepulsii, J. Xue, S. Curtarolo, Ordered phases in ruthenium binary alloys from high-throughput first-principles calculations, *Phys. Rev. B* 84 (2011) 214110, <http://dx.doi.org/10.1103/PhysRevB.84.214110>.
- [98] O. Levy, J. Xue, S. Wang, G.L.W. Hart, S. Curtarolo, Stable ordered structures of binary technetium alloys from first principles, *Phys. Rev. B* 85 (2012) 012201, <http://dx.doi.org/10.1103/PhysRevB.85.012201>.
- [99] G.L.W. Hart, S. Curtarolo, T.B. Massalski, O. Levy, Comprehensive search for new phases and compounds in binary alloy systems based on platinum-group metals, using a computational first-principles approach, *Phys. Rev. X* 3 (2013) 041035, <http://dx.doi.org/10.1103/PhysRevX.3.041035>.
- [100] S. Barzilai, C. Toher, S. Curtarolo, O. Levy, Evaluation of the tantalum-titanium phase diagram from *ab-initio* calculations, *Acta Mater.* 120 (2016) 255–263, <http://dx.doi.org/10.1016/j.actamat.2016.08.053>.
- [101] S. Barzilai, C. Toher, S. Curtarolo, O. Levy, Molybdenum-titanium phase diagram evaluated from *ab initio* calculations, *Phys. Rev. Mater.* 1 (2017) 023604, <http://dx.doi.org/10.1103/PhysRevMaterials.1.023604>.
- [102] C. Toher, C. Oses, D. Hicks, S. Curtarolo, Unavoidable disorder and entropy in multi-component systems, *npj Comput. Mater.* 5 (2019) 69, <http://dx.doi.org/10.1038/s41524-019-0206-z>.
- [103] B. Grabowski, L. Ismer, T. Hickel, J. Neugebauer, *Ab initio* up to the melting point: Anharmonicity and vacancies in aluminum, *Phys. Rev. B* 79 (2009) 134106, <http://dx.doi.org/10.1103/PhysRevB.79.134106>.
- [104] B. Grabowski, Y. Ikeda, P. Srinivasan, F. Körmann, C. Freysoldt, A.I. Duff, A. Shapeev, J. Neugebauer, *Ab initio* vibrational free energies including anharmonicity for multicomponent alloys, *npj Comput. Mater.* 5 (2019) 80, <http://dx.doi.org/10.1038/s41524-019-0218-8>.
- [105] L. Wang, T. Maxisch, G. Ceder, Oxidation energies of transition metal oxides within the GGA+U framework, *Phys. Rev. B* 73 (2006) 195107, <http://dx.doi.org/10.1103/PhysRevB.73.195107>.
- [106] S. Lany, Semiconductor thermochemistry in density functional calculations, *Phys. Rev. B* 78 (2008) 245207, <http://dx.doi.org/10.1103/PhysRevB.78.245207>.
- [107] A. Jain, G. Hautier, S.P. Ong, C.J. Moore, C.C. Fischer, K.A. Persson, G. Ceder, Formation enthalpies by mixing GGA and GGA+U calculations, *Phys. Rev. B* 84 (2011) 045115, <http://dx.doi.org/10.1103/PhysRevB.84.045115>.
- [108] V. Stevanović, S. Lany, X. Zhang, A. Zunger, Correcting density functional theory for accurate predictions of compound enthalpies of formation: Fitted elemental-phase reference energies, *Phys. Rev. B* 85 (2012) 115104, <http://dx.doi.org/10.1103/PhysRevB.85.115104>.
- [109] Y. Zhang, D.A. Kitchaev, J. Yang, T. Chen, S.T. Dacek, R.A. Sarmiento-Pérez, M.A.L. Marques, H. Peng, G. Ceder, J.P. Perdew, J. Sun, Efficient first-principles prediction of solid stability: Towards chemical accuracy, *npj Comput. Mater.* 4 (2018) 9, <http://dx.doi.org/10.1038/s41524-018-0065-z>.
- [110] E.B. Isaacs, C. Wolverton, Performance of the strongly constrained and appropriately normed density functional for solid-state materials, *Phys. Rev. Mater.* 2 (2018) 063801, <http://dx.doi.org/10.1103/PhysRevMaterials.2.063801>.
- [111] J. Yan, J.S. Hummelshøj, J.K. Nørskov, Formation energies of group I and II metal oxides using random phase approximation, *Phys. Rev. B* 87 (2013) 075207, <http://dx.doi.org/10.1103/PhysRevB.87.075207>.
- [112] T.S. Jauho, T. Olsen, T. Bligaard, K.S. Thygesen, Improved description of metal oxide stability: Beyond the random phase approximation with renormalized kernels, *Phys. Rev. B* 92 (2015) 115140, <http://dx.doi.org/10.1103/PhysRevB.92.115140>.
- [113] M. Pozzo, D. Alfé, Structural properties and enthalpy of formation of magnesium hydride from quantum Monte Carlo calculations, *Phys. Rev. B* 77 (2008) 104103, <http://dx.doi.org/10.1103/PhysRevB.77.104103>.
- [114] G. Mao, X. Hu, X. Wu, Y. Dai, S. Chu, J. Deng, Benchmark Quantum Monte Carlo calculation of the enthalpy of formation of MgH₂, *Int. J. Hydrog. Energy* 36 (2011) 8388–8391, <http://dx.doi.org/10.1016/j.ijhydene.2011.04.093>.
- [115] M. Aykol, C. Wolverton, Local environment dependent GGA+U method for accurate thermochemistry of transition metal compounds, *Phys. Rev. B* 90 (2014) 115105, <http://dx.doi.org/10.1103/PhysRevB.90.115105>.
- [116] K. Momma, F. Izumi, VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data, *J. Appl. Crystallogr.* 44 (2011) 1272–1276, <http://dx.doi.org/10.1107/S0021889811038970>.
- [117] W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects, *Phys. Rev.* 140 (1965) A1133, <http://dx.doi.org/10.1103/PhysRev.140.A1133>.
- [118] U. von Barth, L. Hedin, A local exchange–correlation potential for the spin polarized case: I, *J. Phys. C: Solid State Phys.* 5 (1972) 1629, <http://dx.doi.org/10.1088/0022-3719/5/13/012>.
- [119] J. Sun, A. Ruzsinszky, J.P. Perdew, Strongly constrained and appropriately normed semilocal density functional, *Phys. Rev. Lett.* 115 (2015) 036402, <http://dx.doi.org/10.1103/PhysRevLett.115.036402>.
- [120] L.C. Allen, Electronegativity is the average one-electron energy of the valence-shell electrons in ground-state free atoms, *J. Am. Chem. Soc.* 111 (1989) 9003–9014, <http://dx.doi.org/10.1021/ja00207a003>.
- [121] J.B. Mann, T.L. Meek, L.C. Allen, Configuration energies of the main group elements, *J. Am. Chem. Soc.* 122 (2000) 2780–2783, <http://dx.doi.org/10.1021/ja992866e>.

- [122] J.B. Mann, T.L. Meek, E.T. Knight, J.F. Capitani, L.C. Allen, Configuration energies of the d-block elements, *J. Am. Chem. Soc.* 122 (2000) 5132–5137, <http://dx.doi.org/10.1021/ja9928677>.
- [123] J.-P. Poirier, *Introduction to the Physics of the Earth's Interior*, second ed., Cambridge University Press, 2000.
- [124] R. Hill, The elastic behaviour of a crystalline aggregate, *Proc. Phys. Soc. Sect. A* 65 (1952) 349, <http://dx.doi.org/10.1088/0370-1298/65/5/307>.
- [125] M.A. Blanco, E. Francisco, V. Luaña, GIBBS: isothermal-isobaric thermodynamics of solids from energy curves using a quasi-harmonic Debye model, *Comput. Phys. Comm.* 158 (2004) 57–72, <http://dx.doi.org/10.1016/j.comphy.2003.12.001>.
- [126] M.A. Blanco, A.M. Pendás, E. Francisco, J.M. Recio, R. Franco, Thermodynamical properties of solids from microscopic theory: Applications to MgF_2 and Al_2O_3 , *J. Mol. Struct.: Theorchem.* 368 (1996) 245–255, [http://dx.doi.org/10.1016/S0166-1280\(96\)90571-0](http://dx.doi.org/10.1016/S0166-1280(96)90571-0).
- [127] G. Leibfried, E. Schlömann, Wärmeleitung in elektrisch isolierenden kristallen, in: *Nachrichten D. Akad. D. Wiss. in Göttingen. Math.-Physik. Kl. 2a. Math.-Physik.-Chem. Abt.*, Vandenhoeck & Ruprecht, 1954.
- [128] G.A. Slack, The thermal conductivity of nonmetallic crystals, in: H. Ehrenreich, F. Seitz, D. Turnbull (Eds.), *Solid State Physics*, Vol. 34, Academic, New York, 1979, p. 1.
- [129] D.T. Morelli, G.A. Slack, High lattice thermal conductivity solids, in: S.L. Shindé, J.S. Goela (Eds.), *High Thermal Conductivity Materials*, Springer, 2006.
- [130] A.A. Maradudin, S.H. Vosko, Symmetry properties of the normal vibrations of a crystal, *Rev. Modern Phys.* 40 (1968) 1–37, <http://dx.doi.org/10.1103/RevModPhys.40.1>.
- [131] Y. Wang, S.L. Shang, H. Fang, Z.-K. Liu, L.Q. Chen, First-principles calculations of lattice dynamics and thermal properties of polar solids, *npj Comput. Mater.* 2 (2016) 16006, <http://dx.doi.org/10.1038/npjcomputmat.2016.6>.
- [132] The XCRYSDEN software, 2022, <http://www.xcrysden.org/>. Accessed April 13, 2022.
- [133] F.D. Murnaghan, The compressibility of media under extreme pressures, *Proc. Natl. Acad. Sci.* 30 (1944) 244–247, <http://dx.doi.org/10.1073/pnas.30.9.244>.
- [134] F. Birch, Finite elastic strain of cubic crystals, *Phys. Rev.* 71 (1947) 809–824, <http://dx.doi.org/10.1103/PhysRev.71.809>.
- [135] D.M. Teter, G.V. Gibbs, M.B. Boisen, D.C. Allan, M.P. Teter, First-principles study of several hypothetical silica framework structures, *Phys. Rev. B* 52 (1995) 8064–8073, <http://dx.doi.org/10.1103/PhysRevB.52.8064>.
- [136] A.B. Alchagirov, J.P. Perdew, J.C. Boettger, R.C. Albers, C. Fiolhais, Energy and pressure versus volume: Equations of state motivated by the stabilized jellium model, *Phys. Rev. B* 63 (2001) 224115, <http://dx.doi.org/10.1103/PhysRevB.63.224115>.
- [137] Ph. Lambin, J.P. Vigneron, Computation of crystal green's functions in the complex-energy plane with the use of the analytical tetrahedron method, *Phys. Rev. B* 29 (1984) 3430–3437, <http://dx.doi.org/10.1103/PhysRevB.29.3430>.
- [138] F.C. Nix, D. MacNair, A dilatometric study of the order-disorder transformation in Cu-Au Alloys, *Phys. Rev.* 60 (1941) 320–329, <http://dx.doi.org/10.1103/PhysRev.60.320>.
- [139] C.H. Johansson, J.O. Linde, Röntgenographische und elektrische Untersuchungen des CuAu-Systems, *Ann. Physics* 417 (1936) 1–48, <http://dx.doi.org/10.1002/andp.19364170102>.
- [140] E.H. Kisi, M.M. Elcombe, u parameters for the wurtzite structure of ZnS and ZnO using powder neutron diffraction, *Acta Crystallogr. Sect. C* 45 (1989) 1867–1870, <http://dx.doi.org/10.1107/S0108270189004269>.
- [141] A. Santoro, A.D. Mighell, Properties of crystal lattices: the derivative lattices and their determination, *Acta Cryst. A* 28 (1972) 284–287, <http://dx.doi.org/10.1107/S0567739472000737>.
- [142] A. Santoro, A.D. Mighell, Coincidence-site lattices, *Acta Cryst. A* 29 (1973) 169–175, <http://dx.doi.org/10.1107/S0567739473000434>.
- [143] A.K. Rappe, C.J. Casewit, K.S. Colwell, W.A. Goddard III, W.M. Skiff, UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations, *J. Am. Chem. Soc.* 114 (1992) 10024–10035, <http://dx.doi.org/10.1021/ja00051a040>.
- [144] M.D. Hossain, T. Borman, C. Oses, M. Esters, C. Toher, L. Feng, A. Kumar, W.G. Fahrenholtz, S. Curtarolo, D. Brenner, J.M. LeBeau, J.-P. Maria, Entropy landscaping of high-entropy carbides, *Adv. Mater.* 33 (2021) 2102904, <http://dx.doi.org/10.1002/adma.202102904>.
- [145] A. Zunger, S.-H. Wei, L.G. Ferreira, J.E. Bernard, Special quasirandom structures, *Phys. Rev. Lett.* 65 (1990) 353–356, <http://dx.doi.org/10.1103/PhysRevLett.65.353>.
- [146] X. Lim, Mixed-up metals make for stronger, tougher, stretchier alloys, *Nature* 533 (2016) 306–307, <http://dx.doi.org/10.1038/533306a>.
- [147] B. Gludovatz, A. Hohenwarther, D. Catoor, E.H. Chang, E.P. George, R.O. Ritchie, A fracture-resistant high-entropy alloy for cryogenic applications, *Science* 345 (2014) 1153–1158, <http://dx.doi.org/10.1126/science.1254581>.
- [148] J.-W. Yeh, S.-K. Chen, S.-J. Lin, J.-Y. Gan, T.-S. Chin, T.-T. Shun, C.-H. Tsau, S.-Y. Chang, Nanostructured high-entropy alloys with multiple principle elements: Novel alloy design concepts and outcomes, *Adv. Eng. Mater.* 6 (2004) 299–303, <http://dx.doi.org/10.1002/adem.200300567>.
- [149] B. Cantor, I.T.H. Chang, P. Knight, A.J.B. Vincent, Microstructural development in equiatomic multicomponent alloys, *Mater. Sci. Eng. A* 375–377 (2004) 213–218, <http://dx.doi.org/10.1016/j.msea.2003.10.257>.
- [150] A. Sher, M. van Schilfgaarde, A.-B. Chen, W. Chen, Quasichemical approximation in binary alloys, *Phys. Rev. B* 36 (1987) 4279, <http://dx.doi.org/10.1103/PhysRevB.36.4279>.
- [151] M.A. Berding, A. Sher, Electronic quasichemical formalism: Application to arsenic deactivation in silicon, *Phys. Rev. B* 58 (1998) 3853, <http://dx.doi.org/10.1103/PhysRevB.58.3853>.
- [152] V. Vedral, The role of relative entropy in quantum information theory, *Rev. Modern Phys.* 74 (2002) 197, <http://dx.doi.org/10.1103/RevModPhys.74.197>.
- [153] H. Qian, Relative entropy: Free energy associated with equilibrium fluctuations and nonequilibrium deviations, *Phys. Rev. E* 63 (2001) 042103, <http://dx.doi.org/10.1103/PhysRevE.63.042103>.
- [154] J. Kristensen, I. Bilionis, N. Zabarar, Relative entropy as model selection tool in cluster expansion, *Phys. Rev. B* 87 (2013) 174112, <http://dx.doi.org/10.1103/PhysRevB.87.174112>.
- [155] T.B. Massalski, H. Okamoto, P.R. Subramanian, L. Kacprzak (Eds.), *Binary Alloy Phase Diagrams*, ASM International, Materials Park, Ohio, USA, 1990.
- [156] J.-O. Andersson, T. Helander, L. Höglund, P. Shi, B. Sundman, Thermo-Calc and DICTRA, computational tools for materials science, *CALPHAD* 26 (2002) 273–312, [http://dx.doi.org/10.1016/S0364-5916\(02\)00037-8](http://dx.doi.org/10.1016/S0364-5916(02)00037-8).
- [157] M.C. Gao, D.E. Alman, Searching for next single-phase high-entropy alloy compositions, *Entropy* 15 (2013) 4504–4519, <http://dx.doi.org/10.3390/e15104504>.
- [158] Z. Wu, H. Bei, F. Otto, G.M. Pharr, E.P. George, Recovery, recrystallization, grain growth and phase stability of a family of FCC-structured multi-component equiatomic solid solution alloys, *Intermetallics* 46 (2014) 131–140, <http://dx.doi.org/10.1016/j.intermet.2013.10.024>.
- [159] M.S. Lucas, G.B. Wilks, L. Mauger, J.A. Muñoz, O.N. Senkov, E. Michel, J. Horwath, S.L. Semiatin, M.B. Stone, D.L. Abernathy, E. Karapetrova, Absence of long-range chemical ordering in equimolar FeCoCrNi, *Appl. Phys. Lett.* 100 (2012) 251907, <http://dx.doi.org/10.1063/1.4730327>.
- [160] N.D. Stepanov, D.G. Shaysultanov, G.A. Salishchev, M.A. Tikhonovsky, Structure and mechanical properties of a light-weight AlNbTiV high entropy alloy, *Mater. Lett.* 142 (2015) 153–155, <http://dx.doi.org/10.1016/j.matlet.2014.11.162>.
- [161] Y.D. Wu, Y.H. Cai, T. Wang, J.J. Si, J. Zhu, Y.D. Wang, X.D. Hui, A refractory $\text{Hf}_2\text{Nb}_{25}\text{Ti}_{25}\text{Zr}_{25}$ high-entropy alloy with excellent structural stability and tensile properties, *Mater. Lett.* 130 (2014) 277–280, <http://dx.doi.org/10.1016/j.matlet.2014.05.134>.
- [162] O.N. Senkov, G.B. Wilks, D.B. Miracle, C.P. Chuang, P.K. Liaw, Refractory high-entropy alloys, *Intermetallics* 18 (2010) 1758–1765, <http://dx.doi.org/10.1016/j.intermet.2010.05.014>.
- [163] O.N. Senkov, G.B. Wilks, J.M. Scott, D.B. Miracle, Mechanical properties of $\text{Nb}_{25}\text{Mo}_{25}\text{Ta}_{25}\text{W}_{25}$ and $\text{V}_{20}\text{Nb}_{20}\text{Mo}_{20}\text{Ta}_{20}\text{W}_{20}$ refractory high entropy alloys, *Intermetallics* 19 (2011) 698–706, <http://dx.doi.org/10.1016/j.intermet.2011.01.004>.
- [164] X. Yang, Y. Zhang, P.K. Liaw, Microstructure and compressive properties of NbTiV-TaAl₃ high entropy alloys, *Procedia Eng.* 36 (2012) 292–298, <http://dx.doi.org/10.1016/j.proeng.2012.03.043>.
- [165] O.N. Senkov, S.V. Senkova, D.B. Miracle, C. Woodward, Mechanical properties of low-density, refractory multi-principal element alloys of the Cr-Nb-Ti-V-Zr system, *Mater. Sci. Eng. A* 565 (2013) 51–62, <http://dx.doi.org/10.1016/j.msea.2012.12.018>.
- [166] Y. Zhang, X. Yang, P.K. Liaw, Alloy design and properties optimization of high-entropy alloys, *JOM* 64 (2012) 830–838, <http://dx.doi.org/10.1007/s11837-012-0366-5>.
- [167] B. Gorr, M. Azim, H.-J. Christ, T. Mueller, D. Schliephake, M. Heilmair, Phase equilibria, microstructure, and high temperature oxidation resistance of novel refractory high-entropy alloys, *J. Alloys Compd.* 624 (2015) 270–278, <http://dx.doi.org/10.1016/j.jallcom.2014.11.012>.
- [168] O.N. Senkov, J.M. Scott, S.V. Senkova, D.B. Miracle, C.F. Woodward, Microstructure and room temperature properties of a high-entropy TaNbHfZrTi alloy, *J. Alloys Compd.* 509 (2011) 6043–6048, <http://dx.doi.org/10.1016/j.jallcom.2011.02.171>.
- [169] H. Bei, Multi-component solid solution alloys having high mixing entropy, *USA Patent US A1* (2013) 0108502.
- [170] M.C. Gao, B. Zhang, S. Yang, S.M. Guo, Senary refractory high-entropy alloy HfNbTaTiVZr, *Metall. Mater. Trans. A* 47 (2016) 3333–3345, <http://dx.doi.org/10.1007/s11661-015-3105-z>.
- [171] C. An-Ben, S. Arden, *Semiconductor Alloys: Physics and Materials Engineering*, Springer Science & Business Media, 1995.
- [172] J.S. Rutherford, The enumeration and symmetry-significant properties of derivative lattices, *Acta Crystallogr. Sect. A* 48 (1992) 500–508, <http://dx.doi.org/10.1107/S0108767392000898>.
- [173] G.L.W. Hart, R.W. Forcade, Algorithm for generating derivative structures, *Phys. Rev. B* 77 (2008) 224115, <http://dx.doi.org/10.1103/PhysRevB.77.224115>.
- [174] S. Kullback, R.A. Leibler, On information and sufficiency, *Ann. Math. Stat.* 22 (1951) 79–86, <http://dx.doi.org/10.1214/aoms/1177296994>.

- [175] J.W.D. Connolly, A.R. Williams, Density-functional theory applied to phase transformations in transition-metal alloys, *Phys. Rev. B* 27 (1983) 5169–5172, <http://dx.doi.org/10.1103/PhysRevB.27.5169>.
- [176] A. van de Walle, G. Ceder, Automating first-principles phase diagram calculations, *J. Phase Equilib.* 23 (2002) 348–359, <http://dx.doi.org/10.1361/105497102770331596>.
- [177] A. van de Walle, Multicomponent multisublattice alloys, nonconfigurational entropy and other additions to the Alloy Theoretic Automated Toolkit, *CALPHAD* 33 (2009) 266–278, <http://dx.doi.org/10.1016/j.calphad.2008.12.005>.
- [178] K. Charles, K. Herbert, *Thermal Physics*, W.H. Freeman, 1980.
- [179] J.W. Cahn, F. Larché, A simple model for coherent equilibrium, *Acta Metall.* 32 (1984) 1915–1923, [http://dx.doi.org/10.1016/0001-6160\(84\)90173-1](http://dx.doi.org/10.1016/0001-6160(84)90173-1).
- [180] E.A. Guggenheim, *Mixtures: The Theory of the Equilibrium Properties of Some Simple Classes of Mixtures, Solutions and Alloys*, Clarendon Press, 1952.
- [181] G.B. Taggart, Odd-order correlations in disordered alloys: A quasi-binary approximation, *Phys. Lett. A* 45 (1973) 221–222, [http://dx.doi.org/10.1016/0375-9601\(73\)90175-8](http://dx.doi.org/10.1016/0375-9601(73)90175-8).
- [182] A. van de Walle, M.D. Asta, Self-driven lattice-model Monte Carlo simulations of alloy thermodynamic properties and phase diagrams, *Model. Simul. Mater. Sci. Eng.* 10 (2002) 521, <http://dx.doi.org/10.1088/0965-0393/10/5/304>.
- [183] J.J. Kruzic, Bulk metallic glasses as structural materials: A review, *Adv. Eng. Mater.* 18 (2016) 1308–1331, <http://dx.doi.org/10.1002/adem.201600066>.
- [184] J. Schroers, Bulk metallic glasses, *Phys. Today* 66 (2013) 32–37, <http://dx.doi.org/10.1063/PT.3.1885>.
- [185] Y. Li, S. Zhao, Y. Liu, P. Gong, J. Schroers, How many bulk metallic glasses are there? *ACS Comb. Sci.* 19 (2017) 687–693, <http://dx.doi.org/10.1021/acscmbsci.7b00048>.
- [186] A.L. Greer, Confusion by design, *Nature* 366 (1993) 303–304, <http://dx.doi.org/10.1038/366303a0>.
- [187] G.O. Brunner, D. Schwarzenbach, Zur abgrenzung der koordinations-sphere und ermittlung der koordinationszahl in kristallstrukturen, *Z. Kristallogr.* 133 (1971) 127–133, <http://dx.doi.org/10.1524/zkri.1971.133.16.127>.
- [188] J.L.C. Daams, J.H.N. van Vucht, P. Villars, Atomic-environment classification of the cubic intermetallic structure types, *J. Alloys Compd.* 182 (1992) 1–33, [http://dx.doi.org/10.1016/0925-8388\(92\)90570-Y](http://dx.doi.org/10.1016/0925-8388(92)90570-Y).
- [189] J.L.C. Daams, P. Villars, Atomic environments in relation to compound prediction, *Eng. Appl. Artif. Intel.* 13 (2000) 507, [http://dx.doi.org/10.1016/S0952-1976\(00\)00029-4](http://dx.doi.org/10.1016/S0952-1976(00)00029-4).
- [190] Windows subsystem for linux documentation, 2022, <https://docs.microsoft.com/windows/wsl/>. Accessed April 11, 2022.
- [191] The WSL is available for Windows 10 and newer — users of older Windows versions can use Cygwin instead.
- [192] Both the WSL and Cygwin have only few tools to compile C++ code pre-installed.
- [193] The Cygwin Project, 2022, <https://www.cygwin.com/>. Accessed April 11, 2022.
- [194] The website contains further instructions on how to prepare systems for the execution of the script, including on how to set up and install the WSL and Cygwin.
- [195] P.E. Blöchl, Projector augmented-wave method, *Phys. Rev. B* 50 (1994) 17953–17979, <http://dx.doi.org/10.1103/PhysRevB.50.17953>.
- [196] R.L. Rivest, *The MD5 Message-Digest Algorithm*, RFC 1321, RFC Editor, 1992.