



ZStar

Reproducible polarization, Born effective charge, phonon, infrared, Raman, and dielectric-response workflows.

ZStar v0.3.1

Python 3.9+

License GPL-3.0

[English](#) | [简体中文](#) | [English PDF](#) | [中文 PDF](#)

Unified Spectroscopy

One symmetry-adapted SCF ensemble now supports BEC/APT, Gamma phonons, IR and static nonresonant Raman. Raman reuses retained electronic matrices, adding PYATB postprocessing rather than new SCFs.

```
zstar bec pre --stru STRU --input INPUT --dim 3
zstar spectra pre
zstar spectra run
zstar spectra post
```

Set `--dim 2`, `1`, or `0` at BEC preparation for slabs, wires, or molecules. [Full tutorial](#) · [Four-dimensional cost and accuracy comparison](#)

Overview

ZStar automates efficient, symmetry-adapted finite-displacement response calculations. Its Unified ABACUS + PYATB framework reconstructs Born effective charges (BECs), Gamma force constants and Raman derivatives from one displacement ensemble. Polarization, force and electronic dielectric observations yield infrared (IR), Raman and dielectric responses with explicit accuracy checks. Additional VASP, CP2K and Quantum ESPRESSO adapters support their documented response routes; they are complementary to the Unified framework.

The toolkit keeps every stage visible: structures, solver inputs, band-gap gates, polarization values, charge-density data, tensor reconstruction reports, spectra, and progress records remain available for inspection and restart.

The numerical checks used for the current release are summarized in [docs/validation.md](#).

The Unified ABACUS + PYATB workflow shares Phonopy-generated displacements between BEC and Gamma phonons. See the [shared-response guide](#) and [matched examples](#) for its theory, actual-displacement convention, precision safeguards, and validation status. The released package and historical examples retain their recorded versions.

Version **0.3.1** includes the Unified framework, short output names, and completed Separate/Unified benchmarks. Use the matching release when reproducing the manuscript; historical examples retain their original provenance.

For mixed displacements, converge the PYATB Berry mesh as well as the SCF and displacement amplitude. The [direct validation report](#) includes SiC, tetragonal HfO₂, and newly relaxed alpha-In₂Se₃, with raw tensors, mesh/step diagnostics, and separately accounted CPU core-hours.

Main capabilities

- Unified BEC/APT, Gamma force constants and static nonresonant Raman derivatives, with independent Cartesian/mode-displacement controls.
- Symmetry reduction, full-cell tensor reconstruction, and acoustic-sum-rule correction.
- A serial, resumable **0.no-move -> displaced structures** execution model.
- Reuse of the converged **0.no-move** charge density for every displacement.
- One-time insulating-state check after the reference SCF.
- Shell, Slurm, and Torque driver generation.
- Automatic compatibility with legacy and direct-static-response PYATB versions.
- A serial, resumable CP2K backend for Berry-phase BEC tensors and native APT checks.
- Three-dimensional, hybrid two-dimensional, and hybrid one-dimensional polarization/BEC analysis.
- Phonon generation, post-processing, mode classification, IR spectra, Raman spectra, and dielectric response.
- Auxiliary electrostatic-potential analysis for slabs and polar materials.
- A packaged, standards-compliant Agent Skill with JSON workspace preflight.
- A calculator-neutral response schema and backend plugin registry.
- Native Quantum ESPRESSO DFPT collection for molecular and bulk BEC/IR data.

The calculator-independent interface, physical **dim=0/1/2/3** contract, QE workflow, density adapters, Phonopy interchange, polarized Raman, optical constants, and dimensional response normalization are documented in the [calculator-independent guide](#).

Physical Scope

Three-dimensional crystals

For a periodic 3D crystal, ZStar evaluates

$$Z^*(\kappa, \alpha, \beta) = \Omega/e * dP_{\alpha} / du_{(\kappa, \beta)}$$

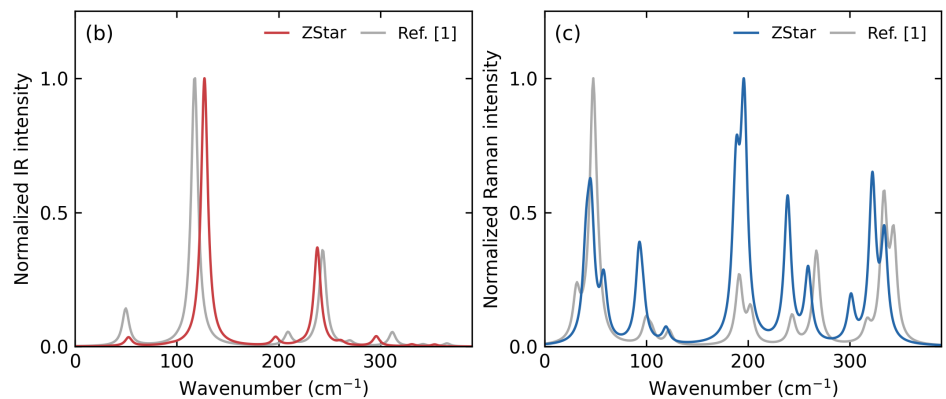
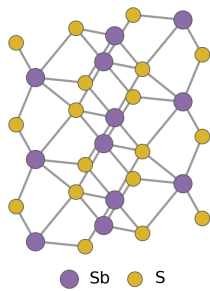
from Berry-phase polarization differences. Polarization branches are matched modulo the polarization quantum before the finite difference is taken.

One-dimensional wires and nanowires

For a z -periodic wire, ZStar combines PYATB Berry polarization along z with real-space ABACUS cube dipoles along the nonperiodic x/y directions. It reports vacuum-independent BEC tensors and $\text{\AA}^2$ line polarizabilities, and supports Gamma-point IR and Raman spectra. A bulk NAC is explicitly rejected because finite-wavevector polar phonons require a genuine 1D Coulomb cutoff. See the [one-dimensional workflow](#).

The [BN\(9,0\) nanotube](#) and [Sb₂S₃ chain](#) provide complete unified BEC/Gamma calculations, full tensors, IR/Raman results, input assets and resumable `run.sh` scripts. Their BEC and force constants use the same SCFs. The Sb₂S₃ comparison retains the original public reference curves and the observed Raman intensity differences; its reference is a computational dataset, not a verified associated journal article.

(a) Sb₂S₃
1D, Nanowire



Two-dimensional slabs

A slab requires separate treatment of in-plane and out-of-plane response:

- **In-plane polarization components:** Berry-phase polarization is used while the full supercell remains insulating.
- **Out-of-plane polarization component:** the total slab dipole is integrated from the ABACUS charge-density cube, including ionic and electronic contributions.
- **Normalization:** in-plane BECs are made independent of vacuum height through the usual volume factor; 2D dielectric spectra are reported as sheet polarizability unless an effective thickness is supplied.

Legacy indexed `Z-BORN-*.out` tables store displacement as rows and polarization as columns. Phonopy `BORN` and the unified reconstruction use polarization-first tensors, as in the equation above; structured records declare their axes. A complete 2D BEC calculation needs information spanning all three displacement

directions. The default unified workflow obtains this from Phonopy seeds and their site-symmetry images; the legacy Cartesian route explicitly generates **x**, **y**, and **z**. The current hybrid implementation requires the slab normal to align with Cartesian **z**; a tilted slab is rejected explicitly.

One reference/displaced cube pair can be audited independently:

```
zstar polar2d --reference-cube reference.cube \  
  --displaced-cube atom_zplus.cube \  
  --displacement 0.01 --outdir slab_dipole_check
```

This writes the planar charge redistribution, dipole finite difference, effective charge, diagnostics, and PNG/PDF/SVG plots.

Installation

ZStar requires Python 3.9 or newer.

The revised manuscript accompanies **0.3.1**. For exact reproduction, install its wheel with `pip install zstar==0.3.1` and obtain the examples from the matching GitHub tag. Examples are not included in the PyPI package. See [the reproducible benchmarks](#) and [revision validation](#).

Install the released package:

```
pip install -U zstar
```

Or install a local checkout:

```
git clone https://github.com/xdzhu/zstar.git  
cd zstar  
pip install .
```

External programs are required only for the corresponding workflows:

- ABACUS for SCF, charge density, force, and sparse-matrix calculations.
- PYATB for Berry-phase polarization, band checks, and electronic dielectric response.
- Phonopy for displacement generation and phonon post-processing.
- CP2K for the optional CP2K finite-displacement BEC backend.
- VASP for native bulk BEC and mode-displaced dielectric-response workflows.
- Quantum ESPRESSO for the optional native DFPT BEC, dielectric, and IR route.

The core package uses `spglib` for periodic symmetry and does not require `pymatgen`. Install the optional VASP extra when using `vasprun.xml`, `CHGCAR/POTCAR` conversion, or the legacy smodes/Wyckoff adapter:

```
pip install -U "zstar[vasp]"
```

POSCAR/CONTCAR structures, ABACUS **STRU** files, MD frame discovery, and the core symmetry-reduction paths are handled by ZStar's lightweight readers.

Verify the command:

```
zstar --version
zstar --help
```

Configure calculator executables once per project (or add **--user** for a user configuration), then let generated drivers select **mpirun** or **srun**:

```
zstar config init
zstar config set executables.abacus /opt/abacus/bin/abacus
zstar config set executables.pyatb /opt/pyatb/bin/pyatb
zstar config check
zstar backend list --check
```

Agent Skill

Install the bundled, standards-compliant **\$run-zstar-workflows** skill after installing ZStar:

```
zstar skill install
zstar skill preflight --root . --lane bec --dim bulk
```

Open a new agent session after installation. Use **--force** to refresh the skill after upgrading ZStar, or **--dest /path/to/skills** for a custom compatible skill directory. The skill encodes dimensional conventions, reference-first execution, restart behavior, scheduler authorization boundaries, and artifact-based completion checks. See docs/agent_skill.md.

The canonical CLI, compatibility aliases, configuration precedence, and every public utility family are summarized in the [command-line reference](#).

Born Charge Workflow

1. Generate the reference and displacement folders

Run this in a directory containing **STRU**:

```
zstar bec pre --stru STRU
```

bec pre defaults to ABACUS + PYATB and the Unified symmetry-adapted displacements with automatic sign selection. These defaults need not be repeated. Use **--calculator cp2k**, **--calculator vasp**, or **--calculator qe** only when selecting another backend.

For a 2D slab:

```
zstar bec pre --stru STRU --dim 2
```

For a **z**-periodic wire:

```
zstar bec pre --stru STRU --dim 1 --method central
```

The default ABACUS + PYATB tree starts with **0.no-move**, followed by Phonopy seed folders **disp-001**, **disp-002**, etc., with **cal_force 1** enabled. BEC and Gamma phonons share these SCFs. **--ensemble cartesian** retains atom/direction folders such as **1.Ti/x+**. No per-displacement scheduler script is required.

Useful generation options:

Option	Meaning
--method auto forward central	Automatic sign selection (shared default), one-sided, or explicit central sampling.
--ensemble phonopy cartesian	Shared BEC/Gamma displacements (default) or the legacy Cartesian layout.
--reduce / --all	Symmetry-reduced atoms (default) or every atom.
--move "x y z"	Specified directions, using the legacy Cartesian route.
--displacement 0.01	Specified step in Angstrom; shared default is 0.02 bohr, using the actual serialized displacement vector in the fit.
--dim 0 1 2 3	Molecular, one-dimensional, two-dimensional, or three-dimensional analysis.
--input-mode abacus pyatb hamgnn custom	Input preparation route.
--input_sets FILES	Extra files or directories copied into generated tasks.
--pp DIR	Directory used to resolve ABACUS pseudopotentials.
--orb DIR	Directory used to resolve ABACUS numerical orbitals.

```
zstar bec pre --stru STRU \  
  --pp /path/to/PSEUDO \  
  --orb /path/to/ORBITAL
```

ABACUS resource files can be kept outside the case directory. When **--pp** or **--orb** is supplied, ZStar resolves each file named in **STRU** before generating any displacement folder. If the exact filename is not found, a unique element-prefix match is accepted; multiple matches stop the command with the candidate list and instructions for disambiguation. The source **STRU** is never modified, and the resolved copy is written to **.zstar/STRU.resolved**. The selected files and SHA256 checksums are recorded in **.zstar/assets.json** and staged into every generated ABACUS task.

Frequently used directories can be configured once:

```
zstar config init --user
zstar config set abacus.pseudo_dir /opt/abacus/PSEUDO --user
zstar config set abacus.orbital_dir /opt/abacus/ORBITAL --user
zstar config check
```

The corresponding configuration is:

```
[abacus]
pseudo_dir = "/opt/abacus/PSEUDO"
orbital_dir = "/opt/abacus/ORBITAL"
```

Command-line directories override the global values. Relative paths in the original `STRU` are interpreted relative to the directory containing that `STRU`, never relative to an accidental shell working directory. If several versions match an element, put the exact filename in `STRU` or pass a narrower directory; ZStar never silently chooses the first match.

2. Run the serial, resumable calculation

Local shell execution:

```
zstar bec run --root . \
  --abacus-command "mpirun -np 1 abacus" \
  --pyatb-command "mpirun -np 1 pyatb" \
  --omp-threads 28
```

For a wire or slab, use `--dim 1` or `--dim 2`, respectively. For an isolated molecule, use `--dim 0`; the workflow uses a Gamma-only supercell and reports atomic polar tensors (APT), not periodic-crystal BEC. For low-dimensional polarization/BEC runs, ZStar writes `out_chg 1 10` so real-space transverse dipole differences are not limited by ABACUS cube rounding.

The default execution order is:

1. Run `0.no-move` SCF and save charge density and sparse matrices.
2. Generate a normal PYATB high-symmetry band path with `pyatb_input --band`.
3. Stop before any displacement if the reference is metallic.
4. Calculate reference polarization and electronic dielectric response.
5. Copy the reference charge cube/restart into each target `OUT.<suffix>/.`
6. Run every displacement serially and calculate its polarization.
7. Record stage state under `.zstar/` so an interrupted run can resume.

The default gate checks the band gap along a regular high-symmetry path. A denser MP-grid check can be requested explicitly:

```
zstar bec run --root . --gap-mode mp --mp-density 0.08
```

Inspect progress at any time:

```
zstar bec stat --root .
```

3. Generate scheduler-specific drivers

Shell:

```
zstar bec job --root . --system shell
```

Slurm:

```
zstar bec job --system slurm
```

Torque/PBS:

```
zstar bec job --system torque
```

Queue, account, resources, time limits and module/environment commands belong in one header: **Specified** `--header FILE` > **Current** `./header.sh` in the workflow root > **Global** `~/zstar/header.sh`. Headers are not merged. Without a header, ZStar generates an editable default with compact guidance. Executable paths and MPI/OMP remain in `zstar config`:

```
zstar config set execution.mpi 1
zstar config set execution.omp 40
zstar bec job --system slurm --header /path/to/header.sh
```

The header allocation must match MPI/OMP. The selected content is embedded and hashed for reproducibility. The same rule applies to `zstar phonon job` and `zstar spectra job`. Complete Slurm, Torque and local examples are in the [job-header tutorial](#). Legacy resource flags and `--env-script` remain supported.

Backend-aware defaults use `mpirun -np N` for shell/Torque and `srun --ntasks=N` for Slurm. Add `--dry-run` to validate the generated script, environment, ordering, and resume records without launching a calculation. The three backends and their scheduler acceptance checks are recorded in [docs/validation.md](#).

Add `--submit` only when the generated script has been reviewed and the active environment provides the required executables.

4. Collect polarization and construct BEC tensors

Molecular APT from ABACUS + PYATB:

```

zstar bec pre --stru STRU --dim 0 --method central --displacement 0.01
zstar bec run --root . \
  --abacus-command abacus --pyatb-command pyatb --omp-threads 20
zstar bec post --root .

```

The Unified collector reconstructs molecular APTs and force constants together, retaining raw and projected tensors in `response_fit.json` and the normalized `response.json`. Its PYATB output adapter retains full-precision polarization values without modifying the installed PYATB kernel. The legacy Cartesian molecular collector writes `apt.json`; for old rounded outputs it can also recover small signals from separately printed phases.

Three-dimensional:

```

zstar bec post --root .

```

Two-dimensional hybrid treatment:

```

zstar bec post --root .

```

One-dimensional hybrid treatment for a `z`-periodic wire:

```

zstar bec post --root .

```

The manifest carries the selected dimensionality and difference method from `pre` into later actions. The old `gen/workflow/deal` commands remain accepted for compatibility.

Key outputs:

File	Meaning
<code>BEC.rep.raw.dat</code>	Raw tensors for explicitly calculated symmetry representatives.
<code>BEC.raw.dat</code>	Raw full-cell tensors, before charge-neutrality projection.
<code>BEC.dat</code>	Full-cell tensors reconstructed by symmetry and corrected by the acoustic sum rule.
<code>BEC.rep.dat</code>	Reduced tensors after reconstruction and neutrality correction.
<code>BORN</code>	Electronic dielectric tensor plus Phonopy-order BEC tensors.
<code>force_fit.json</code>	Force reconstruction diagnostics, separate from the response exchange record.
<code>response_fit.json</code>	Unified raw/projected BEC or APT, force constants, units and diagnostics.
<code>response.json</code>	Standardized response record, including dimensionality and provenance.
<code>BEC_symmetry.json</code>	Legacy Cartesian reconstruction and residual report.
<code>zstar_2d_bec.json</code>	Legacy Cartesian hybrid 2D diagnostics.

File	Meaning
<code>zstar_1d_bec.json</code>	Legacy Cartesian hybrid 1D diagnostics.
<code>apt.json</code>	Legacy Cartesian/cube molecular APT and translational-sum diagnostics.

New outputs use these short names. Existing archives keep their original names (such as `Z-BORN-symm.out`, `zstar_response.json`, and `molecular_apt.json`). Readers fall back to the corresponding old name only when the requested standard file is absent; an explicitly existing file always takes precedence. The rename does not change tensor axes or units. `BORN` is written once, without the redundant `BORN-for-phonopy.out` copy.

CP2K BEC Backend

For a molecular (`--dim 0`) or three-dimensional insulating Gamma-point CP2K input, ZStar can build APT or BEC tensors directly from dipoles:

```
zstar bec pre --calculator cp2k --input input.inp --root cp2k_bec --dim 0 \
  --method central --displacement 0.005
zstar bec run --root cp2k_bec --cp2k-command cp2k.smp \
  --omp-threads 20 --data-dir /path/to/cp2k/data
zstar bec post --root cp2k_bec
```

The reference wavefunction is reused by every displacement, and interrupted runs resume from `.zstar/cp2k_bec_state.json`. CP2K 2025.2+ native `APT_FD` results can be generated and compared through `cp2k-bec native` and `cp2k-bec compare`. See the [complete CP2K guide](#) for tensor conventions, input restrictions, convergence checks, and direct-node validation.

VASP BEC Backend

ZStar can also drive VASP's native BEC implementations. Use `dfpt` for `LEPSILON` linear response or `finite-field` for `LCALCEPS`/PEAD:

```
zstar bec pre --calculator vasp --input-dir vasp_input --root vasp_bec --method dfpt
zstar bec run --root vasp_bec --vasp-command "mpirun -np 20 vasp_std"
zstar bec post --root vasp_bec
```

The reference SCF is checked for a finite band gap before the response stage, and completed stages are resumable. The collector writes normalized JSON, ZStar tensors, and a Phonopy-compatible `BORN` file. The [complete VASP guide](#) documents finite-field safeguards, cluster scripts, tensor conventions, and the VASP 6.3.2 SiC validation.

Phonons and Dielectric Response

For the shared BEC workflow, `zstar bec post` already writes Gamma force constants, `qpoints.yaml`, `irreps.yaml`, and `BORN`. Continue directly with `zstar phonon irrep` and `zstar dielectric static`; no second Gamma SCF set is needed. The separate-directory workflow below is for supercell phonons or legacy archives. Keep a finite-q supercell calculation outside a shared Gamma ensemble, whose real-space interactions cannot resolve a dispersion.

1. Generate phonon calculations

In a phonon working directory containing `STRU`, `KPT`, and an `INPUT` with `cal_force 1`:

```
zstar phonon pre --root . --calculator abacus \  
  --stru STRU --dim "2 2 2" --symprec 1e-3  
zstar phonon run --root .
```

Run every generated `disp-*` force calculation with the local execution system. `zstar ph` does not require or duplicate an `abacus_x.sh`; if one is present it is copied only as an optional convenience.

2. Post-process forces and classify Gamma modes

```
zstar phonon stat --root .  
zstar phonon post --root .  
zstar phonon irrep --root . --file irreps.yaml --mode db
```

For non-analytical corrections, copy the BEC workflow's `BORN` into the phonon directory before post-processing:

```
cp ../polar/BORN .  
zstar phonon post --root . --nac
```

3. Static and frequency-dependent dielectric response

Copy the full tensors as well:

```
cp ../polar/BORN .  
cp ../polar/BEC.dat .
```

Static response:

```
zstar dielectric static --qpoints qpoints.yaml --born BEC.dat \  
  --dielectric BORN --dim 3
```

Frequency-dependent response:

```
zstar dielectric freq --qpoints qpoints.yaml --born BEC.dat \  
--dielectric BORN --dim 3
```

The command writes the zero-frequency tensor, real and imaginary response data, and PNG/PDF/SVG plots by default. Use `--no-plot` for data-only processing. Modes below 5 cm⁻¹ are excluded by default; change this with `--acoustic-cutoff`.

For 2D, omit `--thickness` to obtain a vacuum-independent sheet polarizability in angstroms:

```
zstar dielectric static --qpoints qpoints.yaml --born BEC.dat \  
--dielectric BORN --dim 2
```

`--thickness` `ANGSTROM` produces a thickness-normalized source-field response, not automatically an intrinsic perpendicular slab permittivity. For compatible, screened macroscopic supercell data, `--slab-boundary macroscopic` applies the direct in-plane and inverse out-of-plane conversion. This does not add local-field screening to PYATB. Electric-field conventions and the molecular/1D/2D/bulk Raman units are specified in [response definitions and units](#). The complete convention, bulk and two-dimensional examples, and output contract are documented in the [dielectric-response guide](#).

Infrared Spectrum

Calculate mode effective charges, oscillator strengths, broadened IR intensity, and dielectric/sheet response:

```
zstar spectra pre --kind ir  
zstar spectra post
```

The retained low-level expert command exposes mode selection and plotting details when needed:

```
zstar ir --modes "4,5,8-10" --outdir ir_selected
```

Typical outputs are `ir_modes.csv`, `ir_spectrum.dat`, `ir_response_real.dat`, `ir_response_imag.dat`, `ir_spectrum.png`, `ir_spectrum.pdf`, `ir_spectrum.svg`, and `ir_summary.json`.

Raman Spectrum

ZStar obtains non-resonant Raman tensors by central finite differences of the electronic dielectric response along Gamma-point normal coordinates.

1. Prepare mode displacements

```
zstar spectra pre --stru STRU --kind raman \  
--modes "4-12" --amplitude 0.02
```

The amplitude is a normal-coordinate displacement in `angstrom * sqrt(amu)`.

2. Run, collect, and plot

```
zstar spectra run --omp-threads 28
zstar spectra stat
zstar spectra post
```

The reference insulating gate is reused once; it is not repeated for every mode displacement. Each `plus/minus` stage reuses the reference charge density and records resumable state.

The low-level `zstar raman collect` and `zstar raman spectrum` commands remain available for expert reprocessing of an existing mode tree.

For 2D, `--dim 2` converts the vacuum-dependent dielectric derivative to a sheet-susceptibility derivative using the cell height stored in the phonon data.

Molecular IR and Raman

For the complete physical convention, workflow, outputs, and benchmark, see [Molecular IR and Raman spectroscopy](#).

An isolated molecule is represented in a sufficiently large periodic cell. Prepare its positive-frequency vibrational modes with the same central normal-coordinate displacements used for Raman calculations:

```
zstar spectra pre --stru STRU --dim 0 \
  --acoustic-cutoff 100 --amplitude 0.02
```

The complete resumable calculation produces both spectra:

```
zstar spectra run
zstar spectra post
```

Each displaced SCF is evaluated with two lightweight PYATB stages. The static dielectric response is converted to a molecular polarizability derivative, $d\alpha/dQ = V/(4\pi) * d(\epsilon_r)/dQ$. Branch-wrapped Berry polarization is converted to a molecular dipole derivative, $d\mu/dQ = V * dP/dQ$. The normal-coordinate step Q is in `angstrom * sqrt(amu)`.

Existing polarization results can still be reprocessed with the low-level expert commands:

```
zstar ir --dim 0 --qpoints qpoints.yaml \
  --displacements raman --outdir ir_spectrum
zstar raman spectrum --dim 0 --qpoints qpoints.yaml \
  --raman-dir raman --outdir raman_spectrum
```

Molecular spectra are normalized for mode assignment and workflow validation. They are not reported as gas-phase integrated cross sections. Use a converged vacuum size, basis, displacement amplitude, and electronic-response grid for quantitative intensity work.

VASP and CP2K calculators

The calculator-neutral spectroscopy layer also supports VASP and CP2K:

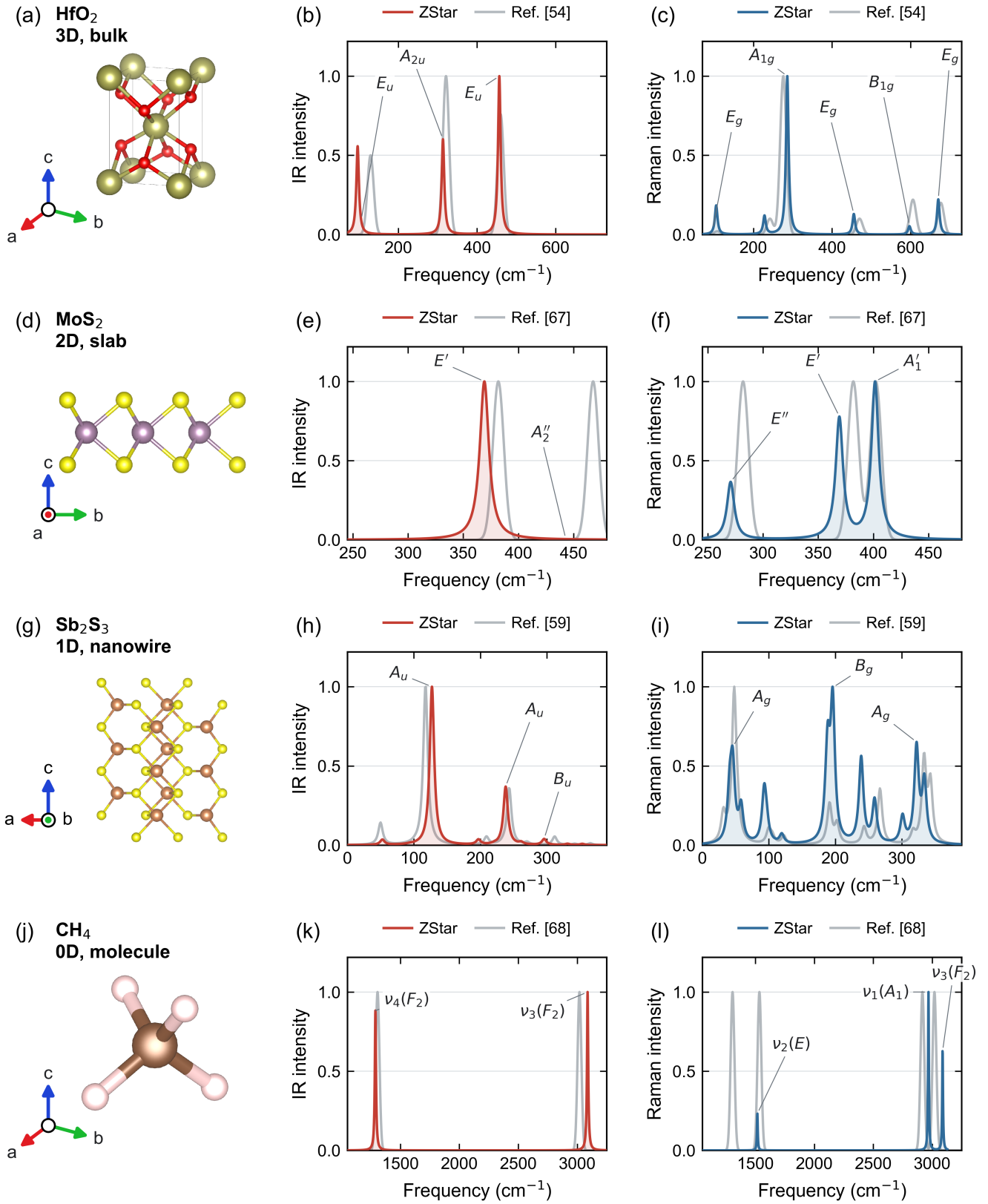
```
zstar spectra pre --calculator vasp --input-dir vasp_input \  
  --modes-xml phonon/vasprun.xml --root vasp_spectra --dim 3  
zstar spectra run --root vasp_spectra --command "mpirun -np 20 vasp_std"  
zstar spectra post --root vasp_spectra  
  
zstar spectra pre --calculator cp2k --input h2o.inp \  
  --root cp2k_spectra --dim 0
```

VASP uses central differences of native dielectric responses; CP2K uses native vibrational dipole and [LINRES/POLAR](#) intensities. See the [calculator spectroscopy guide](#). A complete SiC/HfO2 ABACUS-VASP comparison, including CPU core-hours, is available in the [backend benchmark](#).

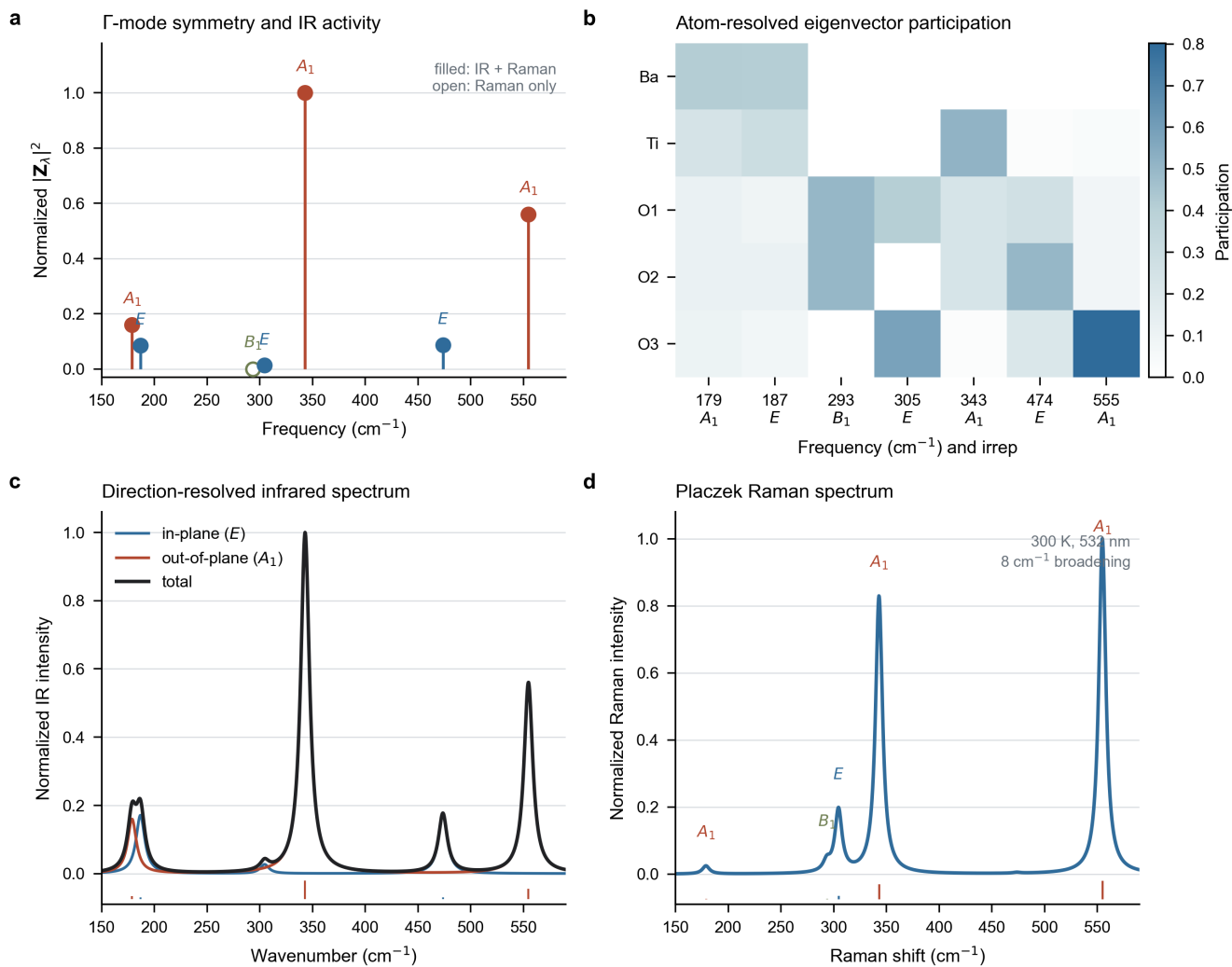
The molecular APT examples also include compact HSE reference records in [examples/0D_Molecules/{H2O,CH4}/results/hse_apt_summary.json](#). The associated solver scratch directories and cube files are intentionally excluded; the JSON records retain the functional, convergence threshold, displacement, tensor convention, and symmetry-corrected values needed to identify the benchmark. Both HSE examples use ABACUS charge-density cube dipoles, without PYATB.

Representative Validation Figures

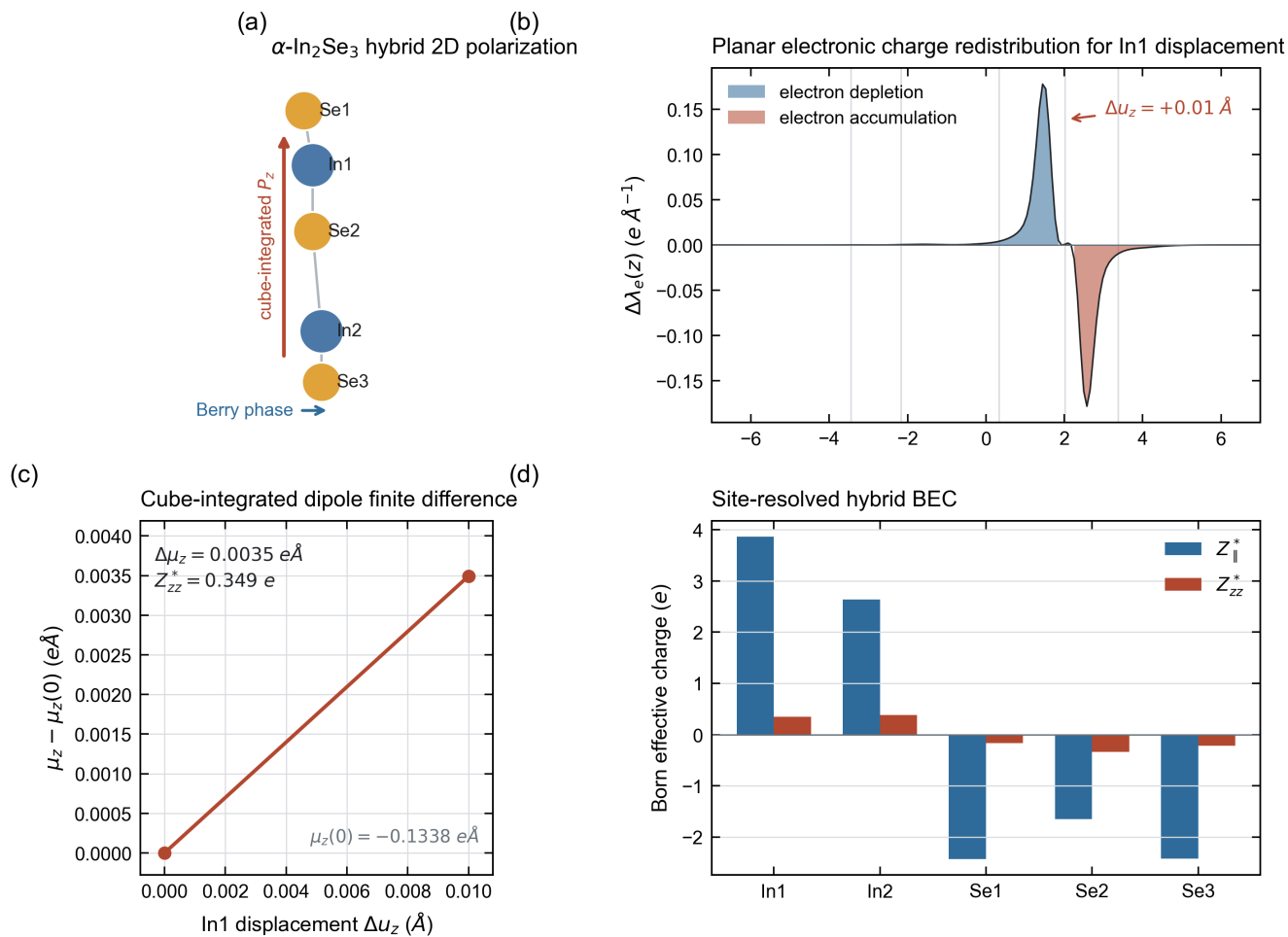
The compact source data, plotting script, vector files, and integrity manifest are archived in [docs/paper_figures](#).



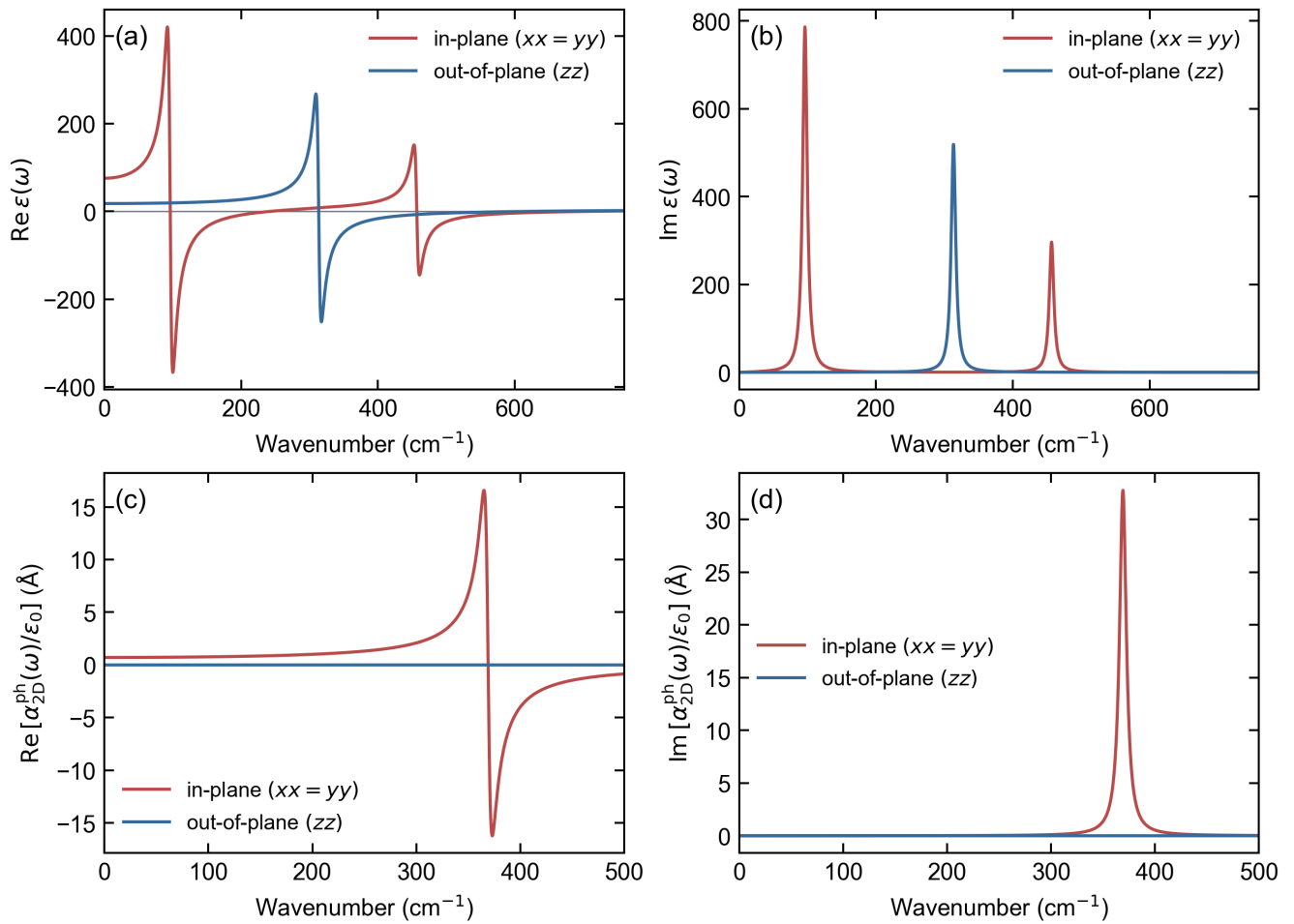
The four-row comparison follows the manuscript order: tetragonal HfO₂ (3D, bulk), monolayer MoS₂ (2D, slab), Sb₂S₃ (1D, nanowire), and CH₄ (0D, molecule). The PBEsol HfO₂ row contains all 15 stable optical modes and 30 completed Raman response stages. The refreshed ABACUS/PBE-D3(BJ) MoS₂ row combines all six optical modes with production BEC-derived IR intensities and 12 completed central-difference Raman response stages.



The BTO validation includes all ten positive-frequency optical modes and 20 completed Raman finite-difference response tasks. The **B1** mode at 293.38 cm^{-1} is Raman active and IR silent.



The In₂Se₃ validation displays the Berry-phase/cube-integral split and the actual planar charge redistribution for an out-of-plane In displacement.



The HfO2 panels show the total bulk response including the electronic background: the PBEsol/TZDP 9-au closure gives `epsilon(0) = diag(75.761034, 75.761034, 18.045191)`. The MoS2 panels show the intrinsic lattice sheet polarizability; they are not a vacuum-dependent supercell dielectric constant.

PYATB Compatibility

ZStar probes the PYATB executable selected for the workflow:

- New builds with direct static response use `static_dielectric_only`.
- Older builds use a compact 0-30 eV optical grid at 0.1 eV spacing. This range was checked against the new direct-static intercept; the coarse spacing avoids an unnecessarily dense full optical spectrum.
- Both `static_dielectric_function.dat` and legacy `dielectric_function_real_part.dat` are accepted.

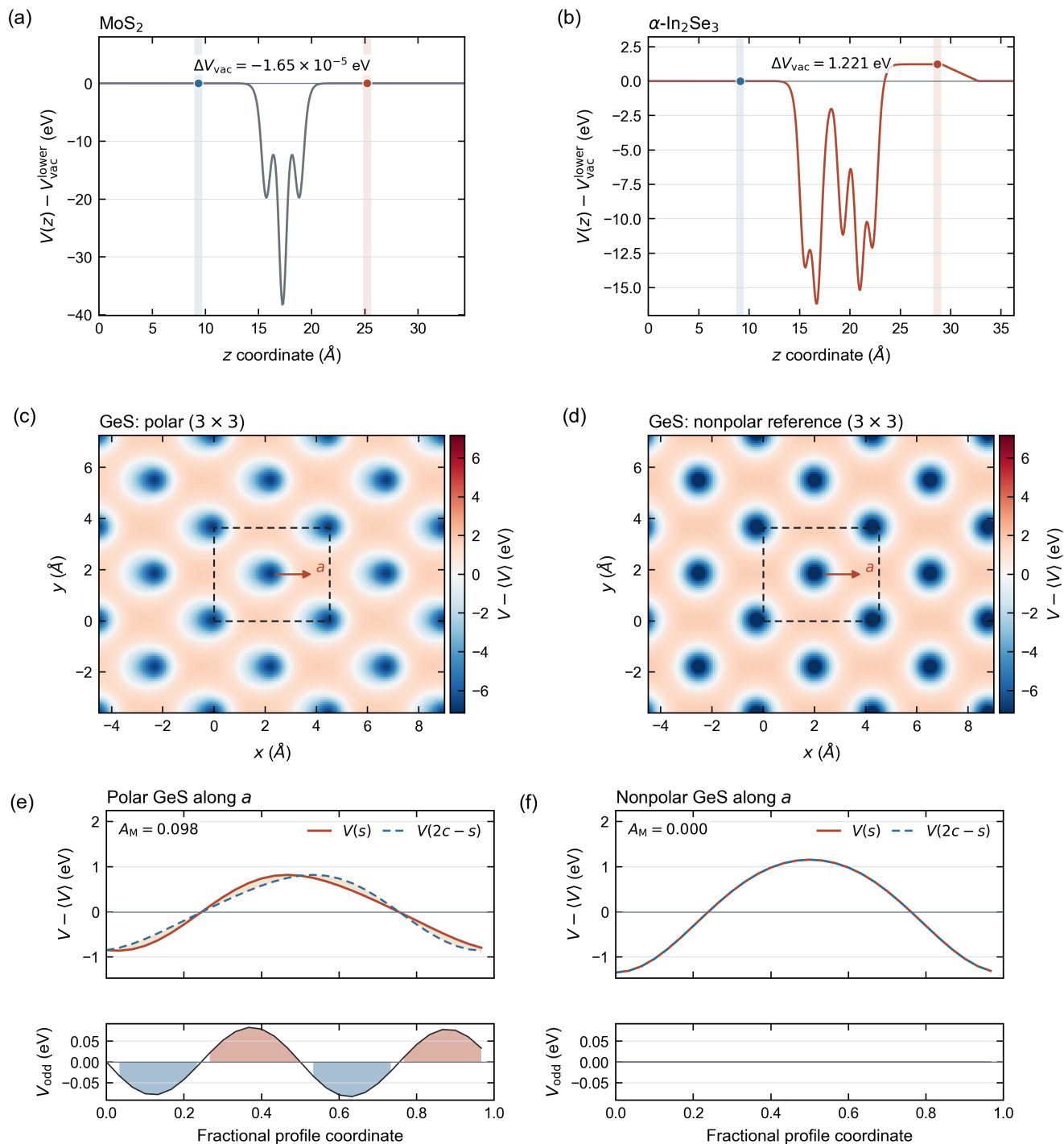
The selected mode and detected version are saved in `zstar_pyatb_compat.json`.

Electrostatic Potential

`zstar_pot` analyzes ABACUS `ElecStaticPot.cube` files:

```
zstar pot --cube OUT.ABACUS/ElecStaticPot.cube \  
  --axes z --plane xy --plane-average --tile 5 5 \  
  --vacuum-level --vacuum-sides --vacuum-window 0.75 \  
  --direction a+b --mirror-test \  
  --polar-arrow auto \  
  --outdir potential
```

It can generate axis profiles, tiled planar maps, directional averages, one- or two-sided vacuum-level diagnostics, and an optimized one-period mirror asymmetry metric. Local side windows avoid mixing a dipole-correction reset into a polar-slab surface plateau. In the representative tests, MoS2 has **Delta V_vac = -1.65e-5 eV**, while alpha-In2Se3 has **Delta V_vac = 1.220812 eV**.



Commands, interpretation limits, and the SnS/SnSe/SnTe directional examples are collected in [docs/potential_examples.md](#).

Command Map

Command	Purpose
<code>zstar bec pre/job/run/stat/post</code>	Polarization, APT/BEC, BORN , resume state, and scheduler drivers.
<code>zstar phonon pre/job/run/stat/post/irrep</code>	Displacements, serial forces, force constants, frequencies, and irreps.

Command	Purpose
<code>zstar spectra pre/job/run/stat/post</code>	Calculator-aware IR and Raman workflows.
<code>zstar dielectric static/freq/optics</code>	Static, vibrational, and electronic dielectric response.
<code>zstar backend list</code>	List capabilities and optionally check executables/plugins.
<code>zstar config init/show/set/check</code>	Configure calculator executable paths.
<code>zstar response</code>	Validate and normalize calculator-neutral response data.
<code>zstar density</code>	Prepare density-export adapters and provenance sidecars.
<code>zstar stru convert/wyckoff</code>	Convert structures or inspect Wyckoff positions.
<code>zstar data db/qnep</code>	Manage a traceable BEC/High-K database or export qNEP data.
<code>zstar skill install/path/preflight</code>	Install the Agent Skill or inspect a workspace.
<code>zstar pot</code>	Plot potential profiles/maps, vacuum steps, and mirror asymmetry.

See the [complete CLI reference](#) for aliases, leaf actions, and executable-resolution rules.

Repository and Release Policy

- `examples/` contains curated, directly runnable case inputs, compact reference results, and backend-specific examples. Large solver scratch outputs remain outside the repository.
- `dist/` and `build/` are local build products and are not committed.
- Scheduler drivers are generated by `zstar bec job`, `zstar phonon job`, and `zstar spectra job`; site-specific resources and environment commands use [Specified, Current or Global headers](#).
- [docs/how_to_update_pypi.md](#) records the release procedure.

The relative logo works for authenticated viewers of a private GitHub repository. PyPI requires a public HTTPS image URL, so `README_PYPI.md` deliberately omits private images.

Citation and License

If ZStar supports published work, please cite the ZStar software article or repository release together with the underlying electronic-structure and lattice-dynamics programs used in the calculation.

Machine-readable citation metadata for this release are provided in [CITATION.cff](#).

ZStar is distributed under the GNU General Public License v3.0.

Copyright (c) Xudong Zhu.

Continuous builds and releases

Every push to `main` and every pull request targeting `main` builds the wheel and source distribution in GitHub Actions and stores them as workflow artifacts. A version tag such as `v0.2.1` additionally creates a GitHub Release and attaches the built files. PyPI publication remains an explicit, separate release step.