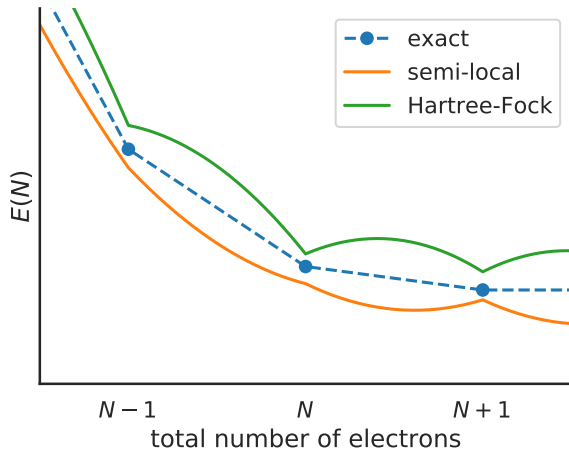
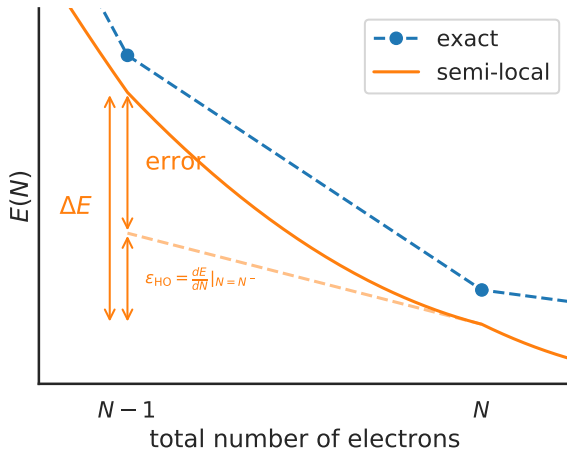


# koopmans

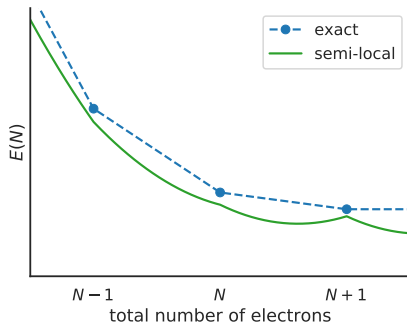
an open-source package for accurately predicting spectral properties



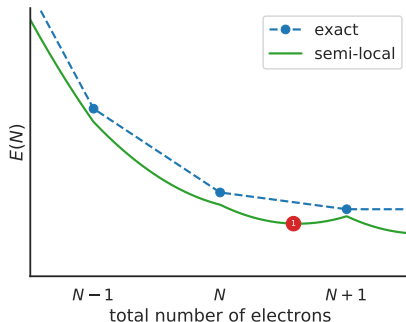
# Self-interaction error



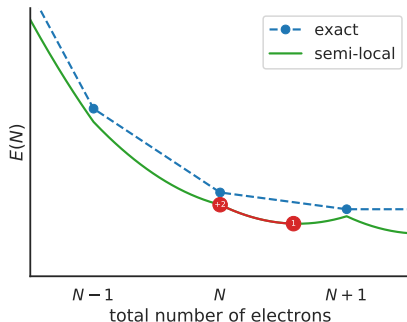
$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{-\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$



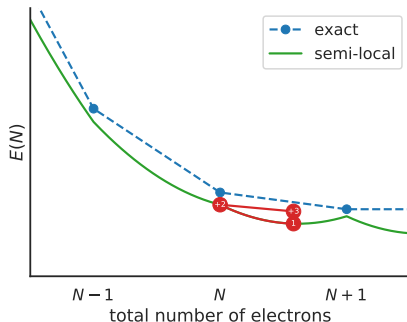
$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{-\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$



$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$



$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{-\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$



$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

Differences to semi-local functionals:



$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{-\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

Differences to semi-local functionals:

- orbital density dependence (can use MLWFs)

$$v_i^{\text{KI}}/\alpha_i = -E_{\text{H}}[n_i] + E_{\text{xc}}[\rho] - E_{\text{xc}}[\rho - n_i] - \int d\mathbf{r}' v_{\text{xc}}(\mathbf{r}', [\rho]) n_i(\mathbf{r}')$$

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{-\int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

Differences to semi-local functionals:

- orbital density dependence (can use MLWFs)
- screening

$$\frac{dE}{df_i} \approx \alpha_i \frac{\partial E}{\partial f_i}$$

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{\text{DFT}}[\rho] + \sum_i \alpha_i \left( \underbrace{- \int_0^{f_i} \varepsilon_i(f) df}_{\text{removes curvature}} + \underbrace{f_i \int_0^1 \varepsilon_i(f) df}_{\text{restores linearity}} \right)$$

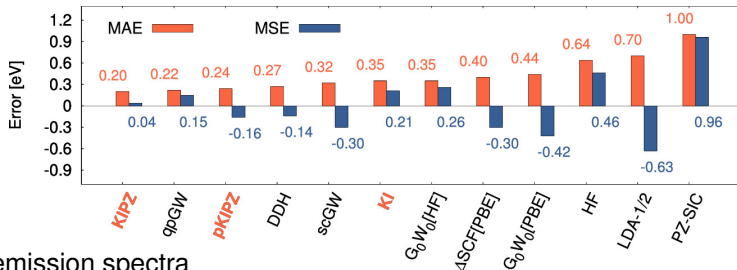
Differences to semi-local functionals:

- orbital density dependence (can use MLWFs)
- screening

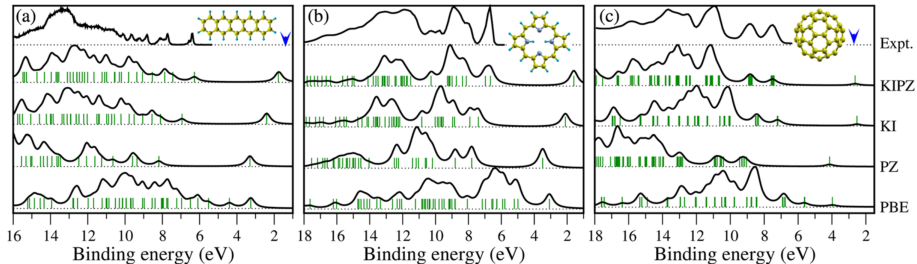
$$\frac{dE}{df_i} \approx \alpha_i \frac{\partial E}{\partial f_i} \Rightarrow \varepsilon_i^{\text{Koopmans}} = \frac{\partial E_{\text{Koopmans}}}{\partial f_i} \approx E_i(N-1) - E(N)$$

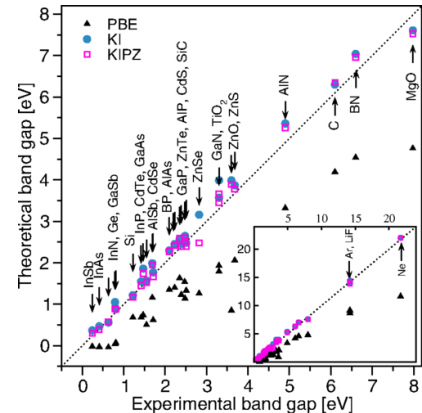
# Koopmans functionals: results for molecules

Ionisation potentials =  $E(N-1) - E(N) \stackrel{?}{=} -\varepsilon_{HO}$  of 100 molecules (the GW100 set) cf. CCSD(T)



Ultraviolet photoemission spectra

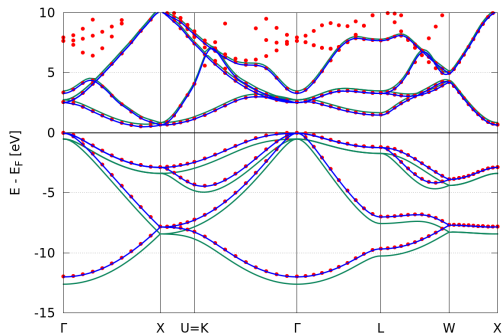




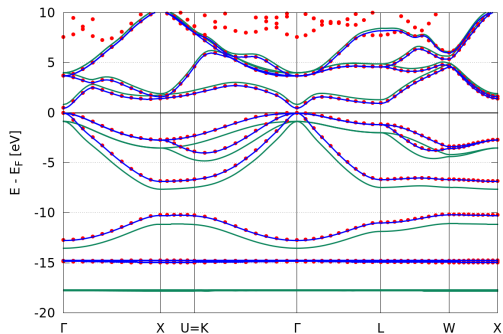
Mean absolute error (eV) across prototypical semiconductors and insulators

	PBE	$G_0W_0$	KI	KIPZ	QSGW
$E_{\text{gap}}$	2.54	0.56	0.27	0.22	0.18
IP	1.09	0.39	0.19	0.21	0.49

# Koopmans functionals: results for solids

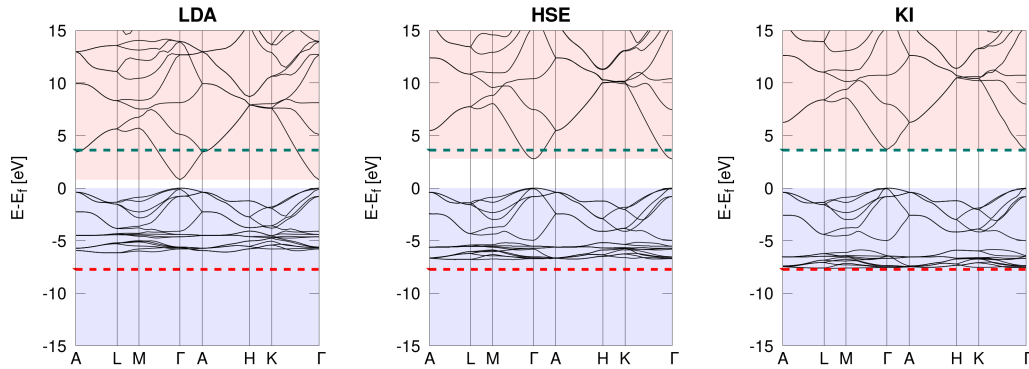


(a) Si, KIPZ



(b) GaAs, KI

		PBE	QSGW	KI	pKIPZ	KIPZ	exp
Si	$E_{\text{gap}}$	0.55	1.24	1.18	1.17	1.19	1.17
	$\langle \varepsilon_d \rangle$	14.9	17.6	16.9		17.7	18.9
GaAs	$E_{\text{gap}}$	0.50	1.61	1.53	1.49	1.50	1.52
	$\langle \varepsilon_d \rangle$	14.9	17.6	16.9		17.7	18.9



ZnO	LDA	HSE	$GW_0$	sc $\tilde{G}\tilde{W}$	KI	exp.
$E_{\text{gap}}$ (eV)	0.79	2.79	3.0	3.2	3.62	3.60
$\langle \epsilon_d \rangle$ (eV)	-5.1	-6.1	-6.4	-6.7	-6.9	-7.5/-8.0

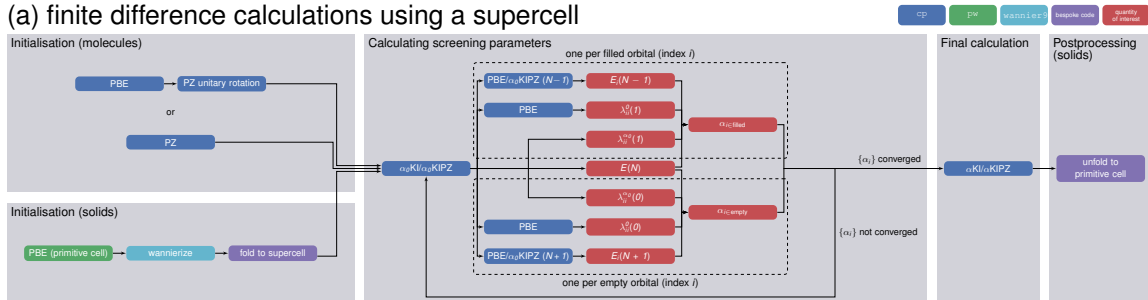
Screening coefficients  $\{\alpha_i\}$  must be determined first, via. . .



## Koopmans functionals: the workflow(s)

Screening coefficients  $\{\alpha_i\}$  must be determined first, via. . .

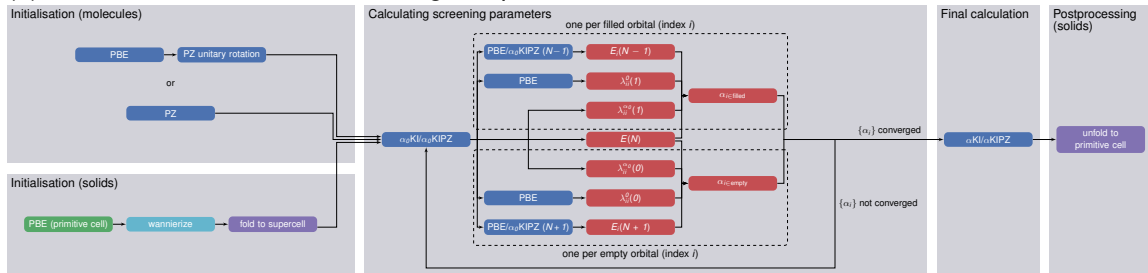
(a) finite difference calculations using a supercell



## Koopmans functionals: the workflow(s)

Screening coefficients  $\{\alpha_i\}$  must be determined first, via. . .

(a) finite difference calculations using a supercell



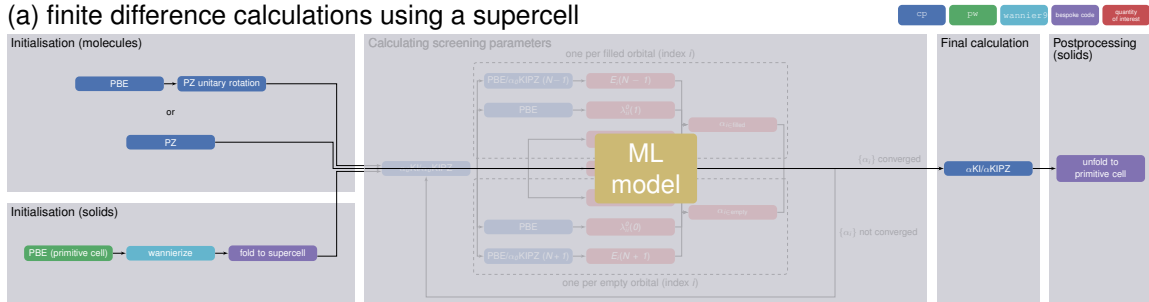
(b) DFPT using a primitive cell



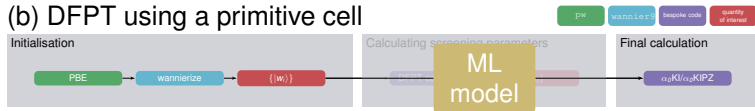
## Koopmans functionals: the workflow(s)

Screening coefficients  $\{\alpha_i\}$  must be determined first, via. . .

(a) finite difference calculations using a supercell



(b) DFPT using a primitive cell



(c) via machine learning (resulting band gaps accurate to  $\sim 0.02$  eV)

$k_{CW,x}$  (DFPT implementation) is distributed in Quantum ESPRESSO v7.1 onwards

But complicated workflows mean that...

- lots of different codes that need to handshake
- lots of scope for human error
- reproducibility becomes difficult

Our solution...

# koopmans

- beta version just released
- implementations of Koopmans functionals
- automated workflows
  - start-to-finish Koopmans calculations
  - Wannierisation
  - dielectric tensor
  - convergence tests
  - ...
- built on top of ASE<sup>1</sup>
- does not require expert knowledge

---

<sup>1</sup> A. H. Larsen et al. *J. Phys. Condens. Matter* 29.27 (12, 2017), 273002.

```
{
  "workflow": {
    "task": "singlepoint",
    "functional": "ki",
    "method": "dscf",
    "init_orbitals": "mlwfs",
    "alpha_guess": 0.1
  },
  "atoms": {
    "atomic_positions": {
      "units": "crystal",
      "positions": [{"Si", 0.00, 0.00, 0.00},
                    ["Si", 0.25, 0.25, 0.25]]
    },
    "cell_parameters": {
      "periodic": true,
      "ibrav": 2,
      "celldm(1)": 10.262
    }
  },
}
```

```
  "k_points": {
    "grid": [8, 8, 8],
    "path": "LGXKG"
  },
  "calculator_parameters": {
    "ecutwfc": 60.0,
    "w90": {
      "projections": [{"fsite": [0.125, 0.125, 0.125],
                        "ang_mtm": "sp3"}]
    },
    "emp": {
      "dis_froz_max": 11.5,
      "dis_win_max": 17.0
    }
  }
}
```

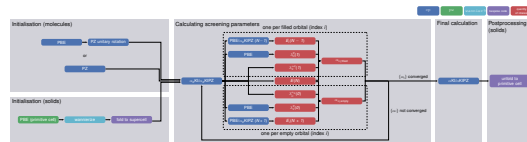
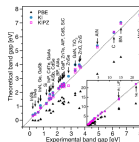
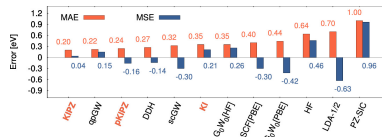
```
from ase.build import bulk
from koopmans.kpoints import Kpoints
from koopmans.projections import ProjectionBlocks
from koopmans.workflows import SinglepointWorkflow

# Use ASE to create bulk silicon
atoms = bulk('Si')

# Define the projections for the Wannierization (same for filled and empty manifold)
si_proj = [{'fsite': [0.25, 0.25, 0.25], 'ang_mtm': 'sp3'}]
si_projs = ProjectionBlocks([{'filled': True, **si_proj},
                              {'filled': False, **si_proj}],
                             atoms=atoms)

# Create the workflow
workflow = SinglepointWorkflow(atoms = atoms,
                               projections = si_projs,
                               ecutwfc = 40.0,
                               kpoints = Kpoints(grid=[8, 8, 8], path='LGXKG', cell=atoms.cell),
                               calculator_parameters = {'pw': {'nbnd': 10},
                                                         'w90_emp': {'dis_froz_max': 10.6, 'dis_win_max': 16.9}})

# Run the workflow
workflow.run()
```



- Koopmans functionals are a class of functionals that treat spectral properties on the same footing as total energy differences (via GPWL)
- they can give orbital energies and band structures with comparable accuracy to state-of-the-art GW
- the release of `koopmans` means you don't need expert knowledge to run Koopmans functional calculations

Want to find out more? Go to [koopmans-functionals.org](https://koopmans-functionals.org)

Free online school Nov 9-11 2022 *Advanced Quantum ESPRESSO tutorial: Hubbard and Koopmans functionals from linear response*. Register at <https://sites.google.com/view/hubbard-koopmans/home>

Follow [@ed\\_linscott](https://twitter.com/ed_linscott) for updates | Slides available at [elinscott.github.io](https://elinscott.github.io)



SPARE SLIDES

## Recap from earlier

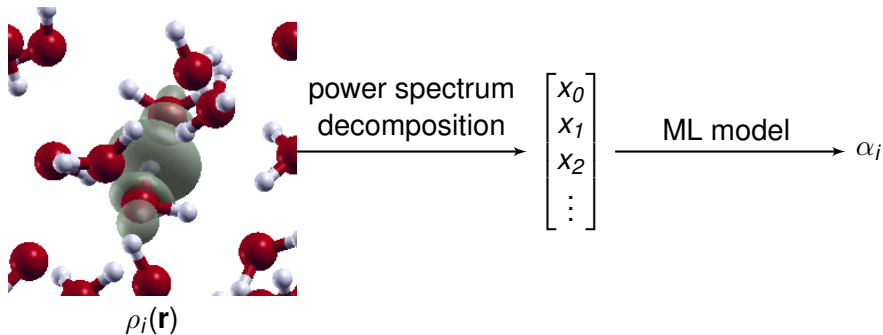
Key idea: construct a functional such that the *variational* orbital energies

$$\varepsilon_i^{\text{Koopmans}} = \langle \varphi_i | H | \varphi_i \rangle = \partial E_{\text{Koopmans}} / \partial f_i$$

possess two key properties:

- they are independent of the corresponding occupancies  $f_i$
- they are equal to the corresponding total energy difference  $E_i(N-1) - E(N)$

zero band gap  $\rightarrow$  occupancy matrix for variational orbitals is off-diagonal

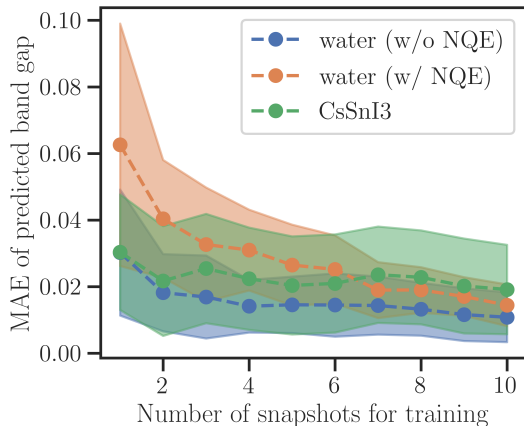
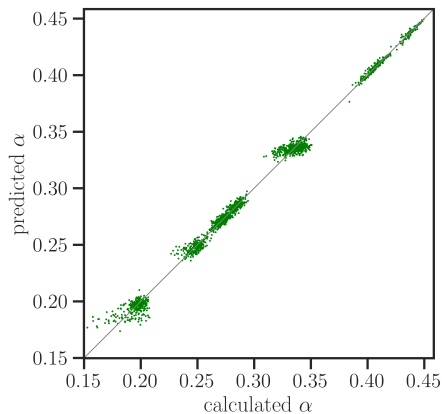


$$c_{nlm,k=\text{orbital}}^i = \int d\mathbf{r} g_{nl}(r) Y_{lm}(\theta, \varphi) \rho^i(\mathbf{r} - \mathbf{R}^i)$$

$g_{nl}$  = orthonormalised radial Gaussian basis functions

$Y_{lm}$  = spherical harmonics

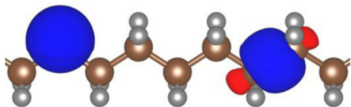
$$p_{n_1 n_2 l, k_1 k_2}^i = \pi \sqrt{\frac{8}{2l+1}} \sum_m c_{n_1 l m, k_1}^{i*} c_{n_2 l m, k_2}^i$$



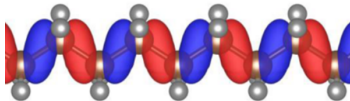
loss of accuracy of the band gap of  $\sim 0.02$  eV  
(cf. when calculating screening parameters *ab initio*)

# Consequences of ODD

- a natural generalisation in the direction of spectral functional theory<sup>1</sup>
- variational (localised, minimising) vs canonical (delocalised, diagonalising) orbitals



(a) variational



(b) canonical

- ODD functional means that we know  $\hat{H}|\varphi_i\rangle$  for variational orbitals  $\{|\varphi_i\rangle\}$  but we don't know  $\hat{H}$  in general
- Difficulties when it comes to calculating transport properties/spectra
- Perhaps a DFT+ $U$ -projector approach is more convenient?

<sup>1</sup> A. Ferretti et al. *Phys. Rev. B* 89.19 (27, 2014), 195134.

N. L. Nguyen et al. *Phys. Rev. X* 8.2 (23, 2018), 021051

# References

- Dabo, I. et al. *Phys. Rev. B* 82.11 (23, 2010).
- Borghi, G. et al. *Phys. Rev. B* 90.7 (20, 2014).
- Nguyen, N. L. et al. *Phys. Rev. X* 8.2 (23, 2018).
- Colonna, N. et al. *J. Chem. Theory Comput.* 14.5 (8, 2018).
- .*J. Chem. Theory Comput.* 15.3 (12, 2019).
- De Gennaro, R. et al. *Phys. Rev. B* 106.3 (5, 2022).
- Colonna, N. et al. *J. Chem. Theory Comput.* (4, 2022).
- Cohen, A. J. et al. *Phys. Rev. B - Condens. Matter Mater. Phys.* 77.11 (18, 2008).
- Li, C. et al. *J. Chem. Phys.* 146.7 (21, 2017).
- Nguyen, N. L. et al. *Phys. Rev. Lett.* 114.16 (24, 2015).
- Van Setten, M. J. et al. *J. Chem. Theory Comput.* 11.12 (22, 2015).
- Schubert, Y. et al. 2022.
- Larsen, A. H. et al. *J. Phys. Condens. Matter* 29.27 (12, 2017).
- Ferretti, A. et al. *Phys. Rev. B* 89.19 (27, 2014).