



# Koopmans functionals

accurately and efficiently predicting spectral properties  
with a functional formulation

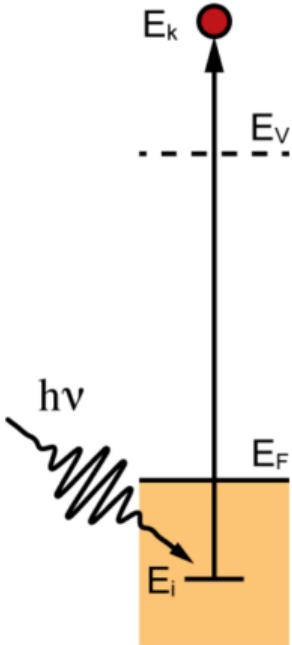
Koopmans functionals are a class of functionals that aim to reproduce spectral properties and total energies on the same footing

As a result they give band structures and orbital energies comparable to state-of-the-art GW

Koopmans can be used to replace GW when calculating neutral excitations with BSE

We have released `koopmans`, a package that contains everything necessary to run calculations using Koopmans functionals without expert knowledge

Goal: spectral properties (charged excitations) with a functional theory



# Koopmans functionals: theory

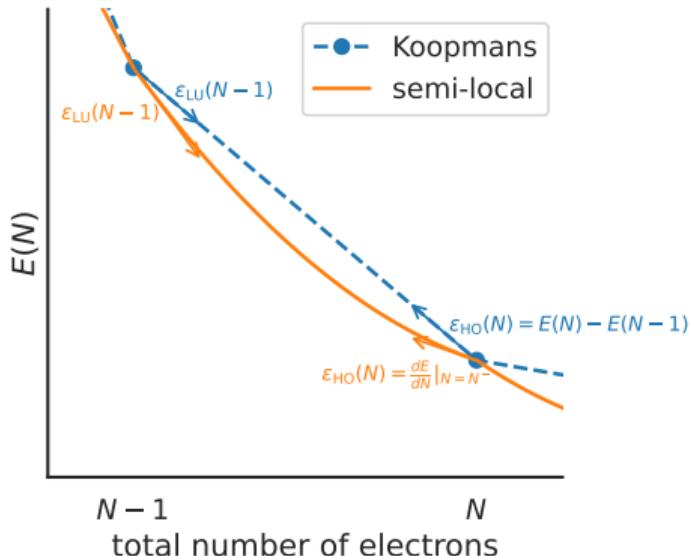
Goal: spectral properties (charged excitations) with a functional theory

Core idea: for every orbital  $i$  their energy

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

should be...

- independent of its own occupation  $f_i$
- equal to the corresponding total energy difference  $E(N-1) - E(N)$



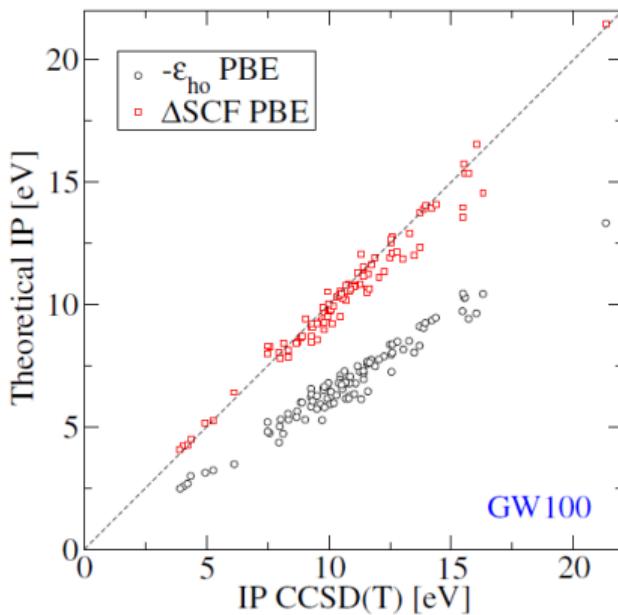
Goal: spectral properties (charged excitations) with a functional theory

Core idea: for every orbital  $i$  their energy

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

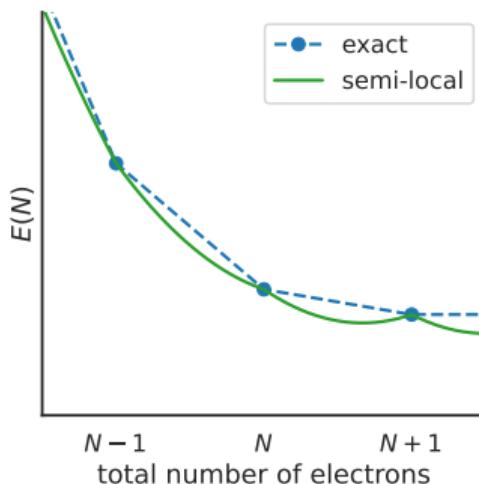
should be...

- independent of its own occupation  $f_i$
- equal to the corresponding total energy difference  $E_i(N - 1) - E(N)$



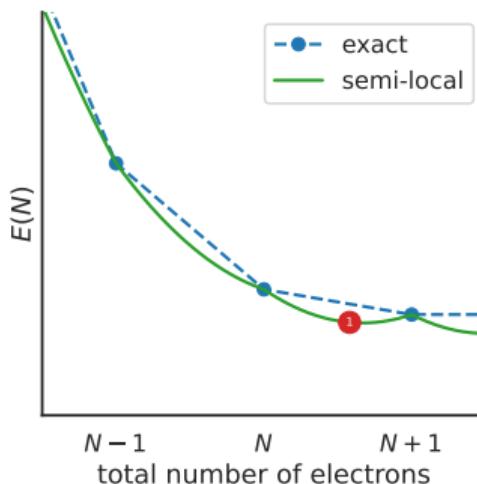
# Koopmans functionals: theory

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



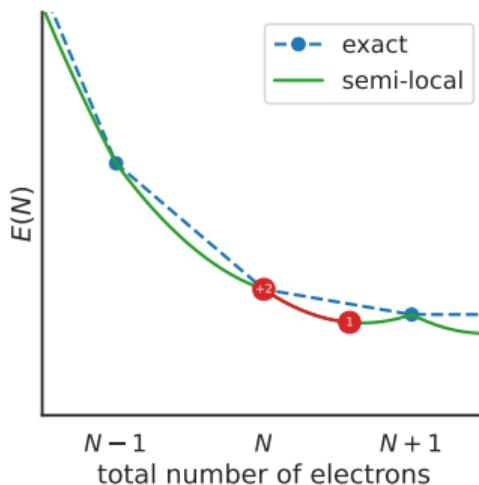
# Koopmans functionals: theory

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



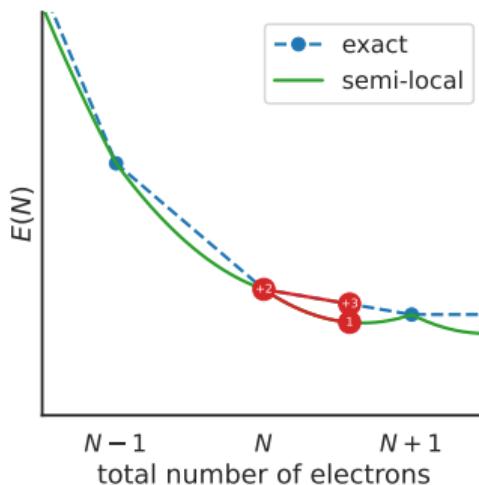
# Koopmans functionals: theory

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



# Koopmans functionals: theory

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



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

Differences to semi-local functionals:

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

Differences to semi-local functionals:

- screening

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

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

Differences to semi-local functionals:

- screening

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

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

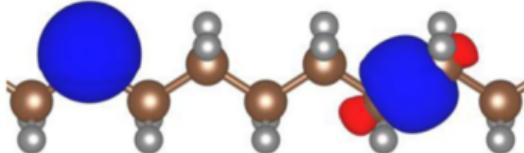
Differences to semi-local functionals:

- screening
- orbital-density dependence

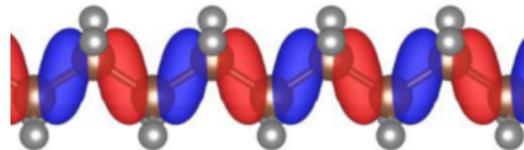
$$v_i^{\text{KI}}/\alpha_i = -E_H[\mathbf{n}_i] + E_{\text{xc}}[\rho] - E_{\text{xc}}[\rho - \mathbf{n}_i] - \int d\mathbf{r}' v_{\text{xc}}(\mathbf{r}', [\rho]) \mathbf{n}_i(\mathbf{r}')$$

Consequences of ODD:

- variational (localised, minimising) vs canonical (delocalised, diagonalising) orbitals



(a) variational



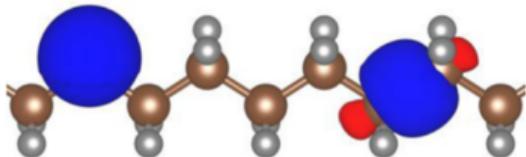
(b) canonical

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

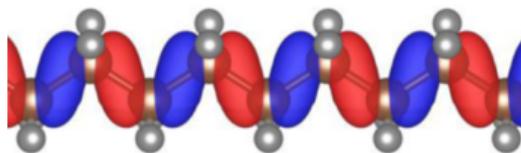
N. L. Nguyen et al. *Phys. Rev. X* 8.2 (2018), 021051

Consequences of ODD:

- variational (localised, minimising) vs canonical (delocalised, diagonalising) orbitals



(a) variational



(b) canonical

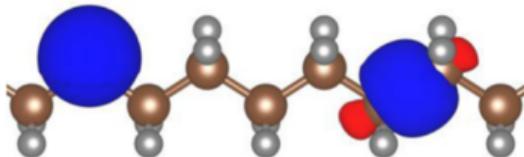
- ODD functional means that we know  $\hat{H}|\varphi_i\rangle$  for variational orbitals  $\{|\varphi_i\rangle\}$  but we don't know  $\hat{H}$  in general

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

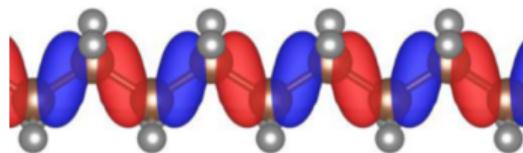
N. L. Nguyen et al. *Phys. Rev. X* 8.2 (2018), 021051

Consequences of ODD:

- variational (localised, minimising) vs canonical (delocalised, diagonalising) orbitals



(a) variational



(b) canonical

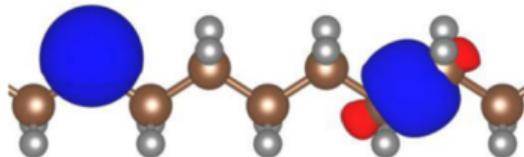
- ODD functional means that we know  $\hat{H}|\varphi_i\rangle$  for variational orbitals  $\{|\varphi_i\rangle\}$  but we don't know  $\hat{H}$  in general
- Practically we can often use MLWFs

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

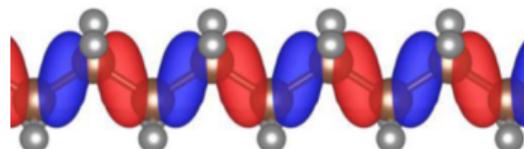
N. L. Nguyen et al. *Phys. Rev. X* 8.2 (2018), 021051

Consequences of ODD:

- 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
- Practically we can often use MLWFs
- a natural generalisation in the direction of spectral functional theory<sup>1</sup>

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

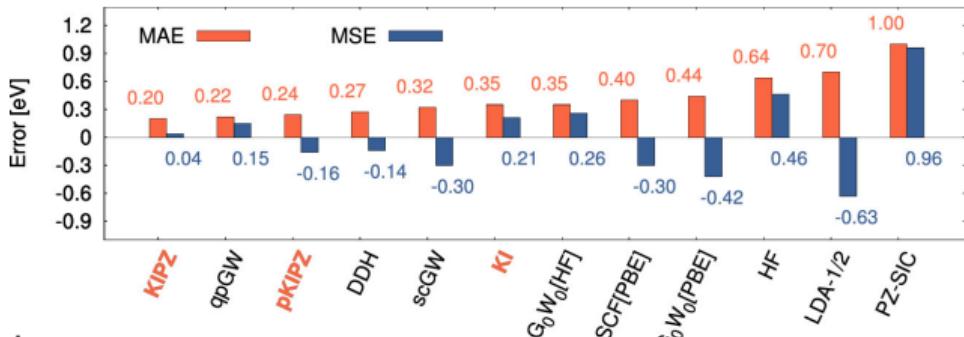
N. L. Nguyen et al. *Phys. Rev. X* 8.2 (2018), 021051

Resonance with other efforts:

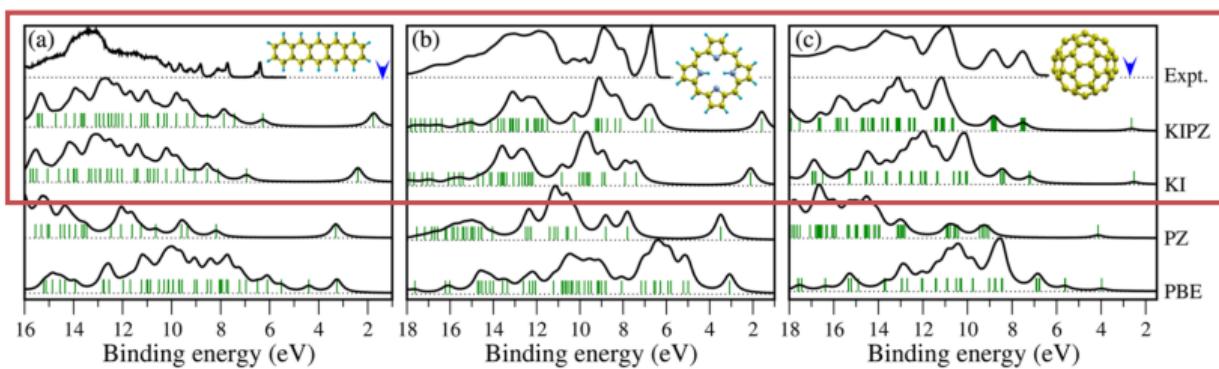
- Wannier transition-state method of Anisimov and Kozhevnikov V. I. Anisimov et al. *Phys. Rev. B* 72.7 (2005), 075125
- Optimally tuned hybrid functionals of Kronik, Pasquarello, and others L. Kronik et al. *J. Chem. Theory Comput.* 8.5 (2012), 1515; D. Wing et al. *Proc. Natl. Acad. Sci.* 118.34 (2021), e2104556118
- Ensemble DFT of Kronik and co-workers E. Kraisler et al. *Phys. Rev. Lett.* 110.12 (2013), 126403
- Koopmans-Wannier of Wang and co-workers J. Ma et al. *Sci. Rep.* 6.1 (2016), 24924
- Dielectric-dependent hybrid functionals of Galli and co-workers J. H. Skone et al. *Phys. Rev. B* 93.23 (2016), 235106
- LOSC functionals of Yang and co-workers C. Li et al. *Natl. Sci. Rev.* 5 (2018). <https://academic.oup.com/nsr/article/5/2/203/4104965>, 203

# Koopmans functionals: results for molecules

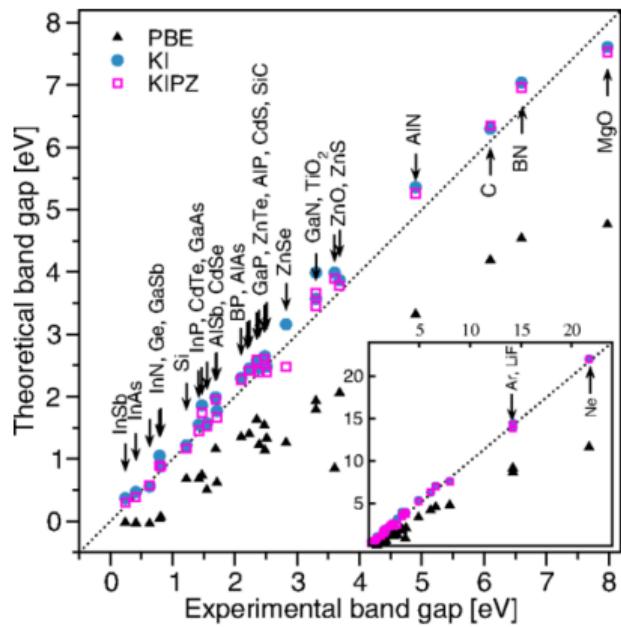
Ionisation potentials =  $E(N - 1) - E(N)$  ?  $= -\varepsilon_{HO}$  of 100 molecules (the GW100 set) cf. CCSD(T)



## Ultraviolet photoemission spectra



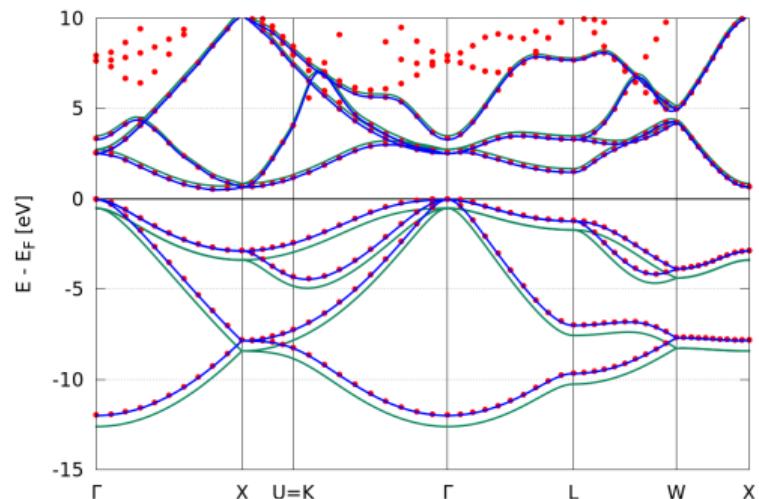
# Koopmans functionals: results for solids



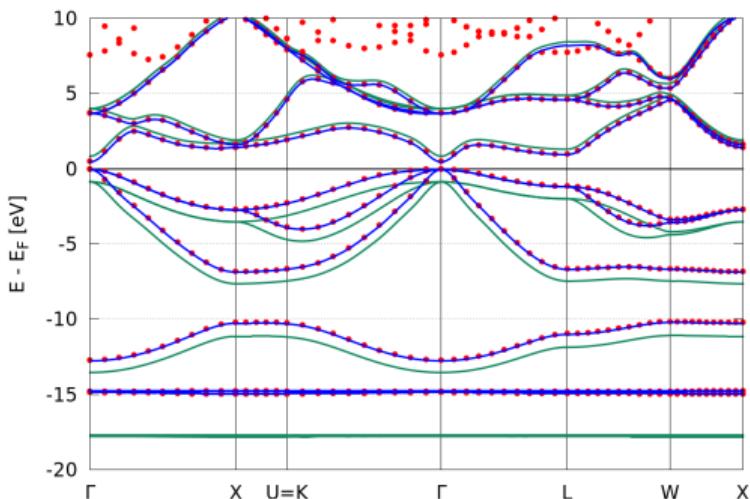
Mean absolute error (eV) across prototypical semiconductors and insulators

|                  | PBE  | G <sub>0</sub> W <sub>0</sub> | KI   | KIPZ | QSGW̃ |
|------------------|------|-------------------------------|------|------|-------|
| $E_{\text{gap}}$ | 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



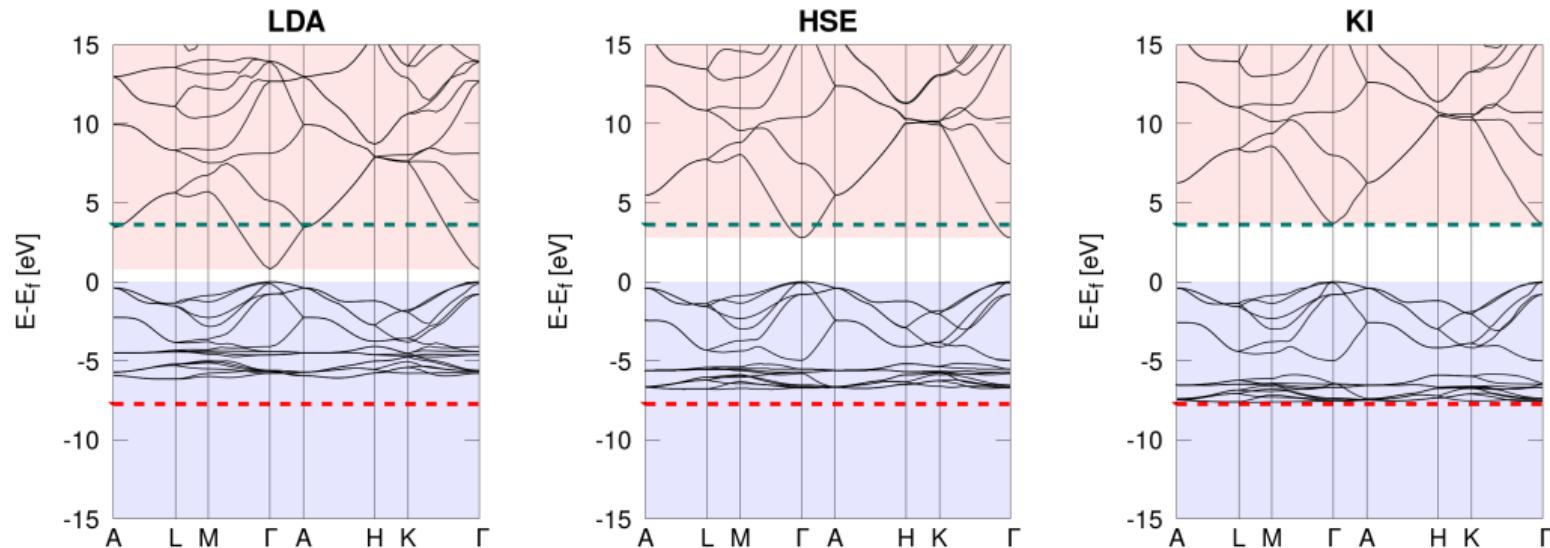
(a) Si, KIPZ



(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 |
| 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 |      |

# Koopmans functionals: results for solids



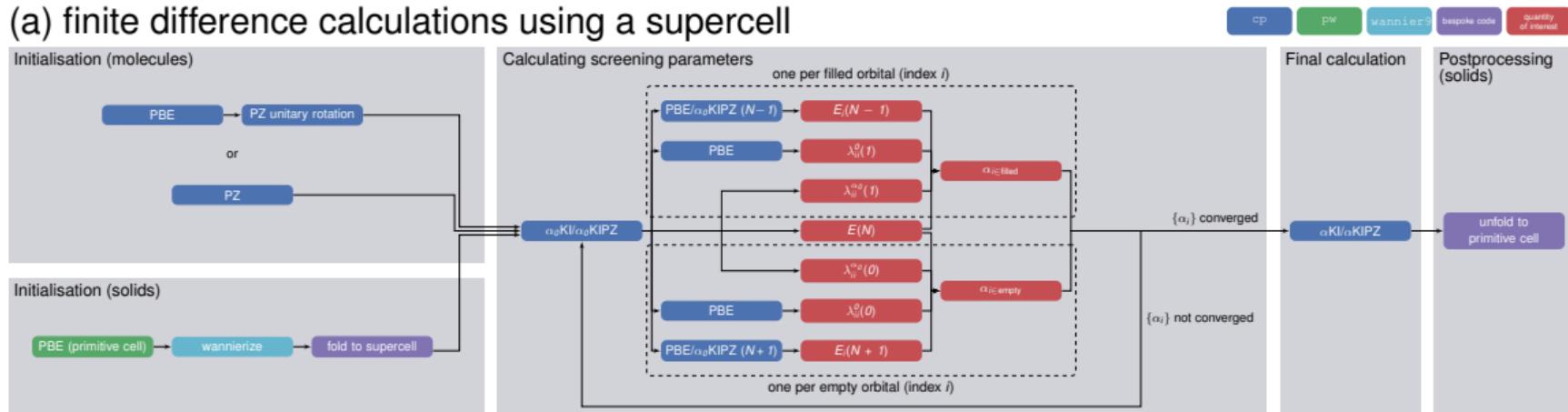
| ZnO                                  | LDA  | HSE  | $GW_0$ | $scG\tilde{W}$ | KI   | exp       |
|--------------------------------------|------|------|--------|----------------|------|-----------|
| $E_{gap}$ (eV)                       | 0.79 | 2.79 | 3.0    | 3.2            | 3.62 | 3.60      |
| $\langle \varepsilon_d \rangle$ (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 workflows

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

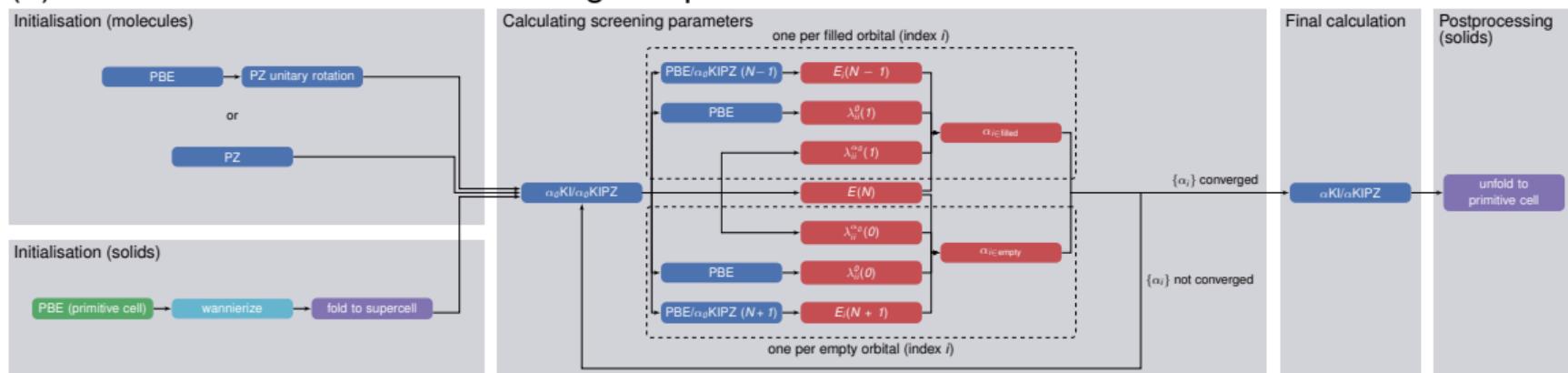
## (a) finite difference calculations using a supercell



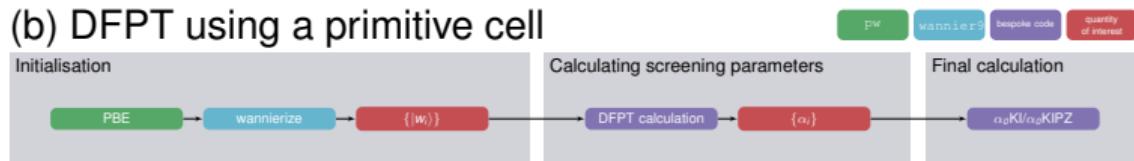
# Koopmans functionals: the workflows

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

## (a) finite difference calculations using a supercell



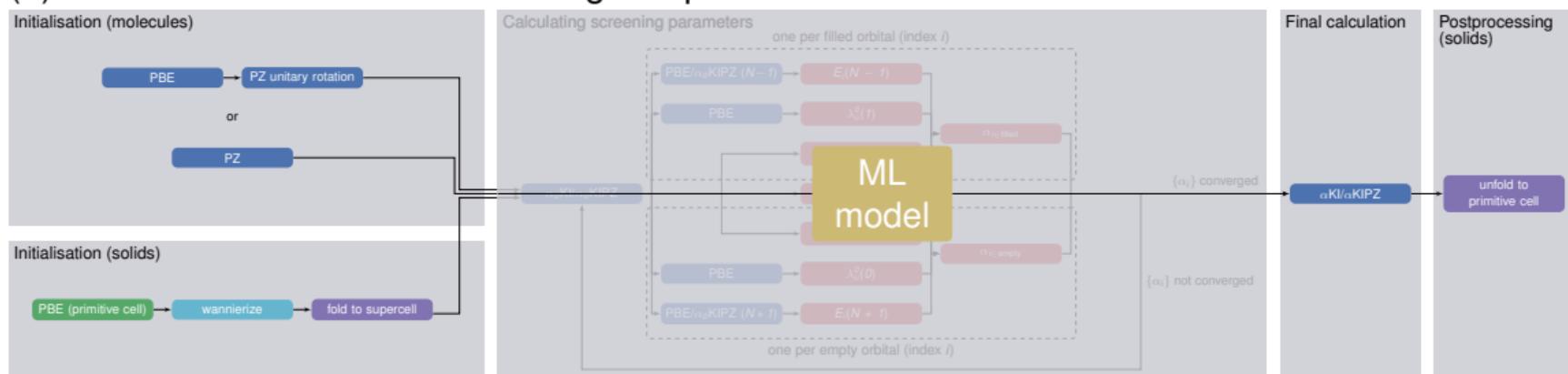
## (b) DFPT using a primitive cell



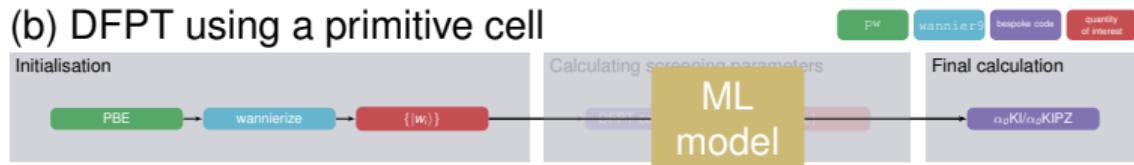
# Koopmans functionals: the workflows

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

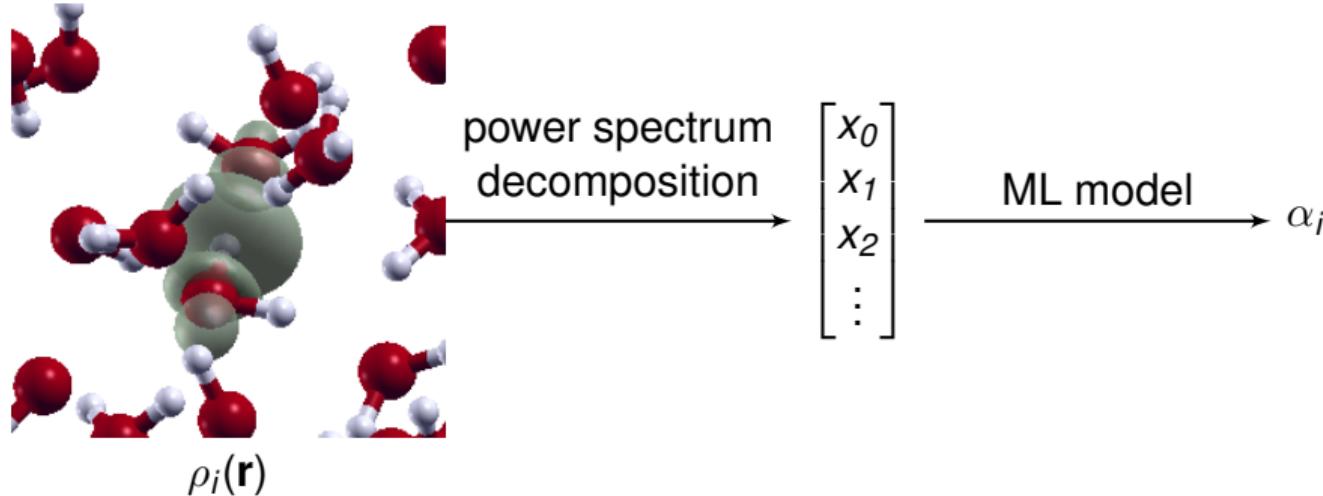
The general workflow:

- define/initialize a set of variational orbitals
- calculate the screening parameters  $\{\alpha_i\}$
- construct and diagonalize the Hamiltonian

Recent advances make some of these steps a lot easier...

J. Qiao et al. *Projectability Disentanglement for Accurate and Automated Electronic-Structure Hamiltonians*. <http://arxiv.org/abs/2303.07877>. 2023  
J. Qiao et al. *Automated Mixing of Maximally Localized Wannier Functions into Target Manifolds*. <http://arxiv.org/abs/2306.00678>. 2023

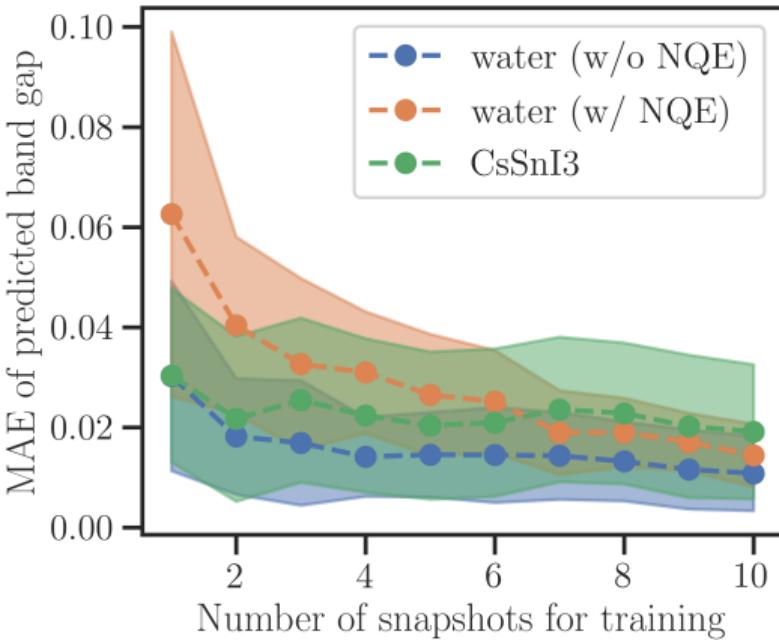
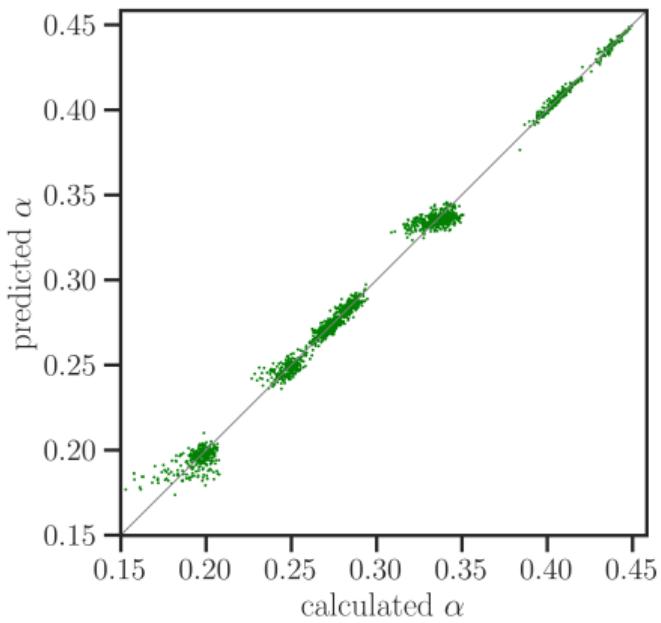
# Learning the screening parameters



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

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

# Learning the screening parameters



loss of accuracy of the band gap of  $\sim 0.02$  eV  
(cf. when calculating screening parameters *ab initio*)  
speedup of 70 $\times$

`kcw.x` (DFPT implementation) is distributed in Quantum ESPRESSO v7.1 onwards

But complex workflows mean that...

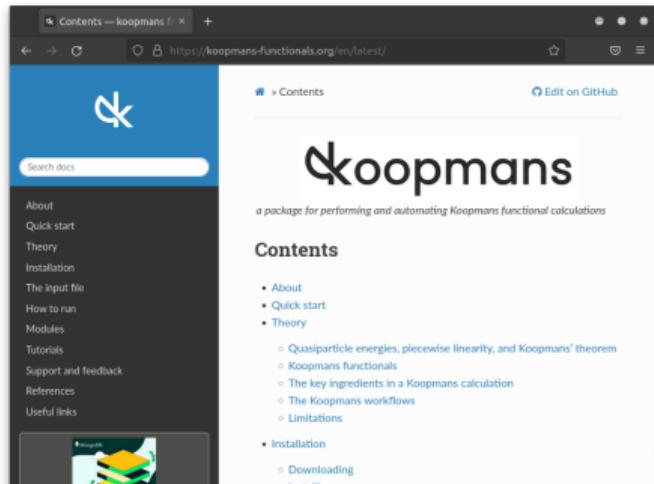
- lots of different codes that need to handshake
- lots of scope for human error
- reproducibility becomes difficult
- expert knowledge required

Our solution...

# koopmans

- beta version released earlier this year<sup>1</sup>
- implementations of Koopmans functionals
- automated workflows
  - start-to-finish Koopmans calculations
  - Wannierisation
  - dielectric tensor
  - convergence tests
  - ...
- built on top of ASE<sup>2</sup>
- does not require expert knowledge

koopmans-functionals.org



<sup>1</sup>Linscott et al., in prep

<sup>2</sup>A. H. Larsen et al. *J. Phys. Condens. Matter* 29.27 (2017), 273002

# koopmans: the input file

```
{  
    "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"}],  
            [{"fsite": [0.125, 0.125, 0.125],  
             "ang_mtm": "sp3"}]  
        ],  
        "dis_froz_max": 11.5,  
        "dis_win_max": 17.0  
    }  
}
```

# koopmans is scriptable

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

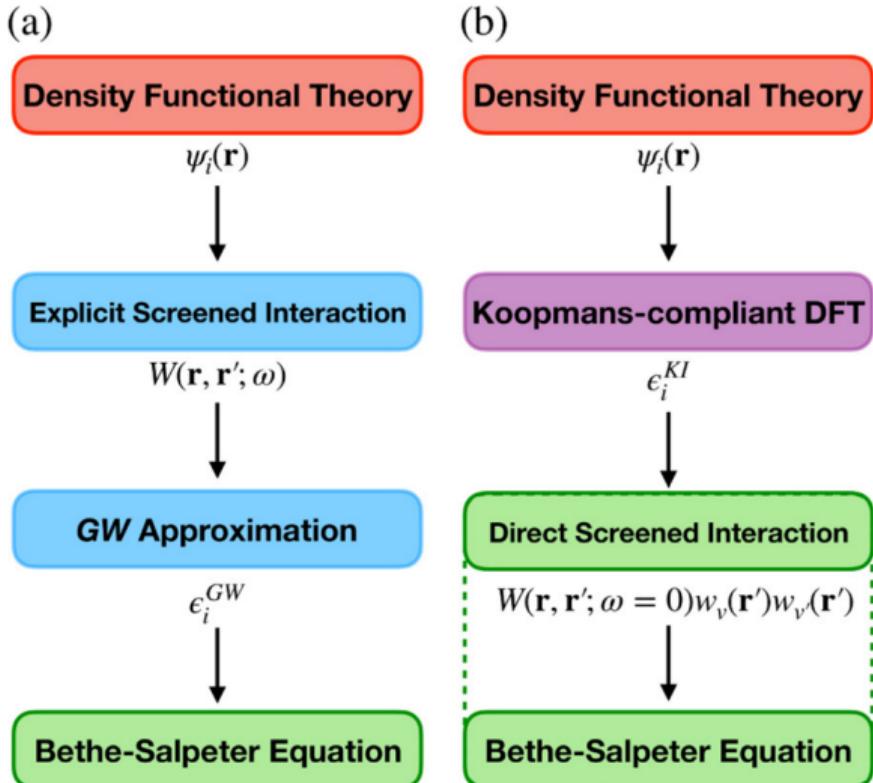
# Use ASE to create bulk silicon
atoms = bulk('Si')

# 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.from_list([si_proj, si_proj], atoms=atoms)

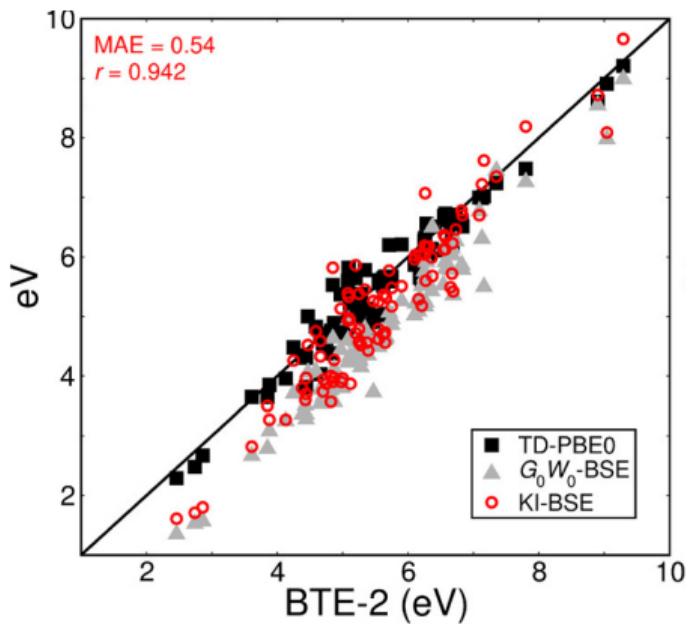
# Create the workflow
workflow = SinglepointWorkflow(atoms = atoms,
                                projections = si_projs,
                                ecutwfc = 40.0,
                                kpoints = Kpoints(grid=[8, 8, 8], path='LGXKG', cell=atoms.cell),
                                calculator_parameters = {'pw': {'nbnd': 10},
                                                        'w90': {'dis_froz_max': 10.6, 'dis_win_max': 16.9}})

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

# Neutral excitations with Koopmans

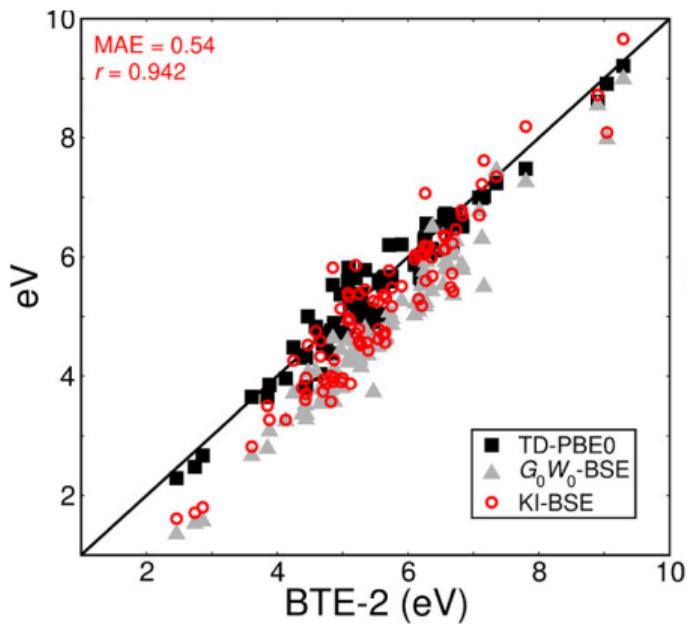


# Neutral excitations with Koopmans



| Thiel's set | KI-BSE<br>PBE | $G_0W_0$ -BSE<br>PBE | TDDFT<br>PBE | $G_0W_0$ -BSE<br>B3LYP | TDDFT<br>B3LYP |
|-------------|---------------|----------------------|--------------|------------------------|----------------|
| MAE (eV)    | 0.54          | 0.83                 | 0.46         | 0.55                   | 0.27           |

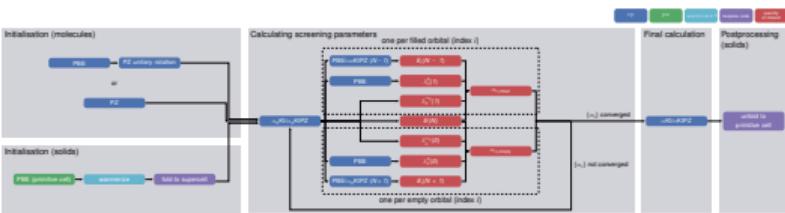
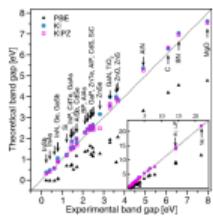
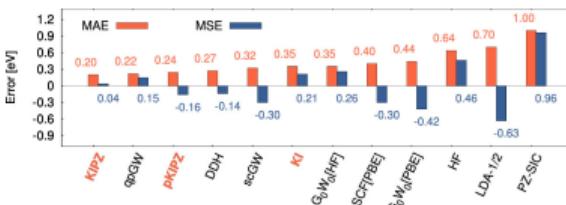
# Neutral excitations with Koopmans



| Thiel's set | KI-BSE<br>PBE | $G_0W_0$ -BSE<br>PBE | $G_0W_0$ -BSE<br>B3LYP | TDDFT<br>PBE | TDDFT<br>B3LYP |
|-------------|---------------|----------------------|------------------------|--------------|----------------|
| MAE (eV)    | 0.54          | 0.83                 | 0.46                   | 0.55         | 0.27           |

Based on RPA → room for improvement with finite-field approaches

# Take home messages



- 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
- Koopmans can replace GW when calculating neutral excitations with BSE

# Acknowledgements



Nicola Marzari



Nicola Colonna



Riccardo De Gennaro



Yannick Schubert



**Swiss National  
Science Foundation**

**MARVEL**



NATIONAL CENTRE OF COMPETENCE IN RESEARCH

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

Free online school Nov 9-11 2022 *Advanced Quantum ESPRESSO tutorial: Hubbard and Koopmans functionals from linear response*. Too late to register but it will be recorded!

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

# SPARE SLIDES

## Recap from earlier

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

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

are...

- independent of the corresponding occupancies  $f_i$
- 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