

# ***Black-box, accurate, and efficient prediction of band structures with Koopmans functionals***

**Edward Linscott**



**PSI**



**DPG conference**



**21 March 2024**



# Accurate band structures

Band structures are a pillar of electronic structure theory... but can we calculate them accurately and efficiently?

# Accurate band structures

Band structures are a pillar of electronic structure theory... but can we calculate them accurately and efficiently?

DFT qualitative failures

# Accurate band structures

Band structures are a pillar of electronic structure theory... but can we calculate them accurately and efficiently?

DFT qualitative failures

GW accurate but expensive,  
ill-behaved, and not a functional

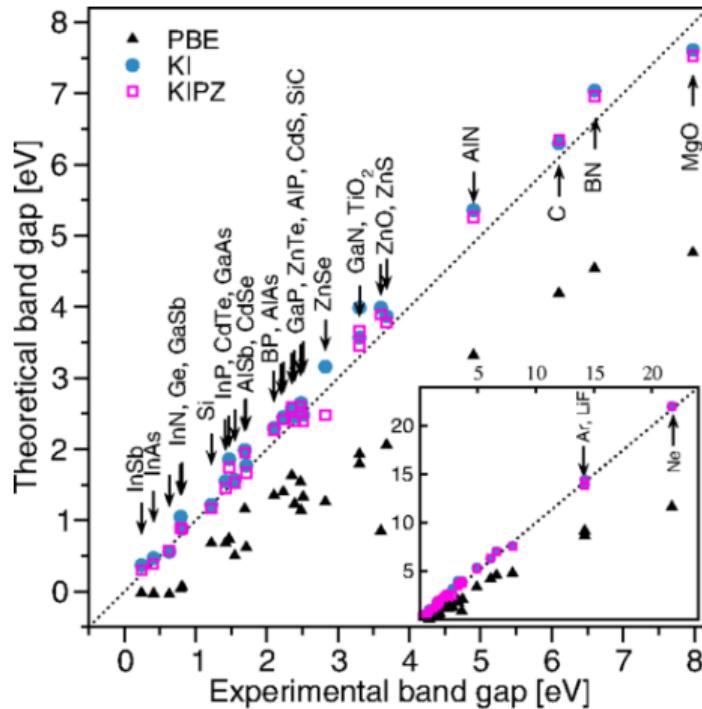
# Accurate band structures

Band structures are a pillar of electronic structure theory... but can we calculate them accurately and efficiently?

DFT qualitative failures

GW accurate but expensive,  
ill-behaved, and not a functional

Koopmans functionals accurate band  
structures within a functional  
theory



# Accurate band structures

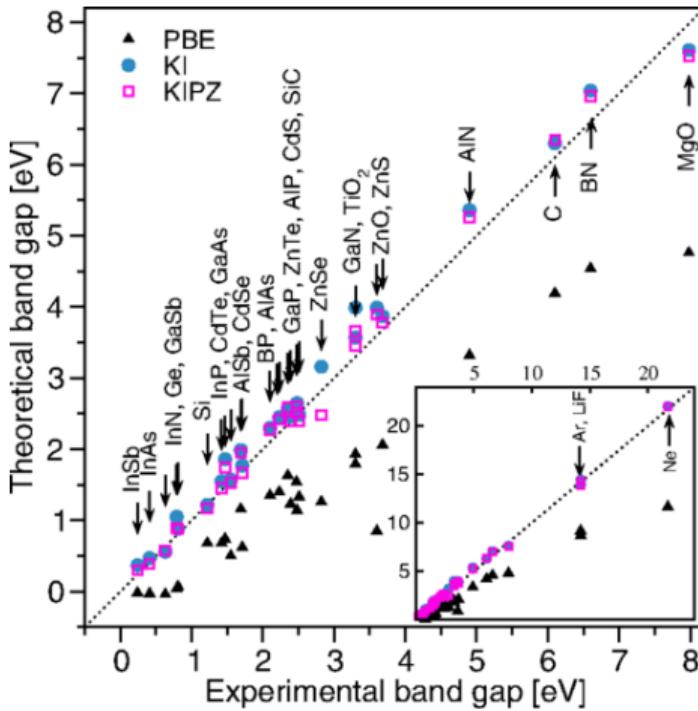
Band structures are a pillar of electronic structure theory... but can we calculate them accurately and efficiently?

DFT qualitative failures

GW accurate but expensive,  
ill-behaved, and not a functional

Koopmans functionals accurate band  
structures within a functional  
theory

|                  | PBE  | $G_0W_0$ | KI   | KIPZ | $QS\tilde{G}W$ |
|------------------|------|----------|------|------|----------------|
| $E_{\text{gap}}$ | 2.54 | 0.56     | 0.27 | 0.22 | 0.18           |
| IP               | 1.09 | 0.39     | 0.19 | 0.21 | 0.49           |



# Features of Koopmans functionals

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left( - \int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left( - \int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition (eigenvalues should match total energy differences)

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left( - \int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition (eigenvalues should match total energy differences)
- is orbital-density-dependent

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left( - \int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition (eigenvalues should match total energy differences)
- is orbital-density-dependent
- relies on localization

$$E_{\text{Koopmans}}[\rho, \{f_i\}, \{\alpha_i\}] = E_{DFT}[\rho] + \sum_i \alpha_i \left( - \int_0^{f_i} \varepsilon_i(f) df + f_i \eta_i \right)$$

General features:

- a correction to DFT that enforces a generalized piecewise linearity condition (eigenvalues should match total energy differences)
- is orbital-density-dependent
- relies on localization
- requires the ab initio calculation of screening parameters

# What should go in the box?

In order to evaluate this functional, one must...

# What should go in the box?

In order to evaluate this functional, one must...

- initialize a set of variational orbitals

# What should go in the box?

In order to evaluate this functional, one must...

- initialize a set of variational orbitals
- calculate the screening parameters  $\{\alpha_i\}$

# What should go in the box?

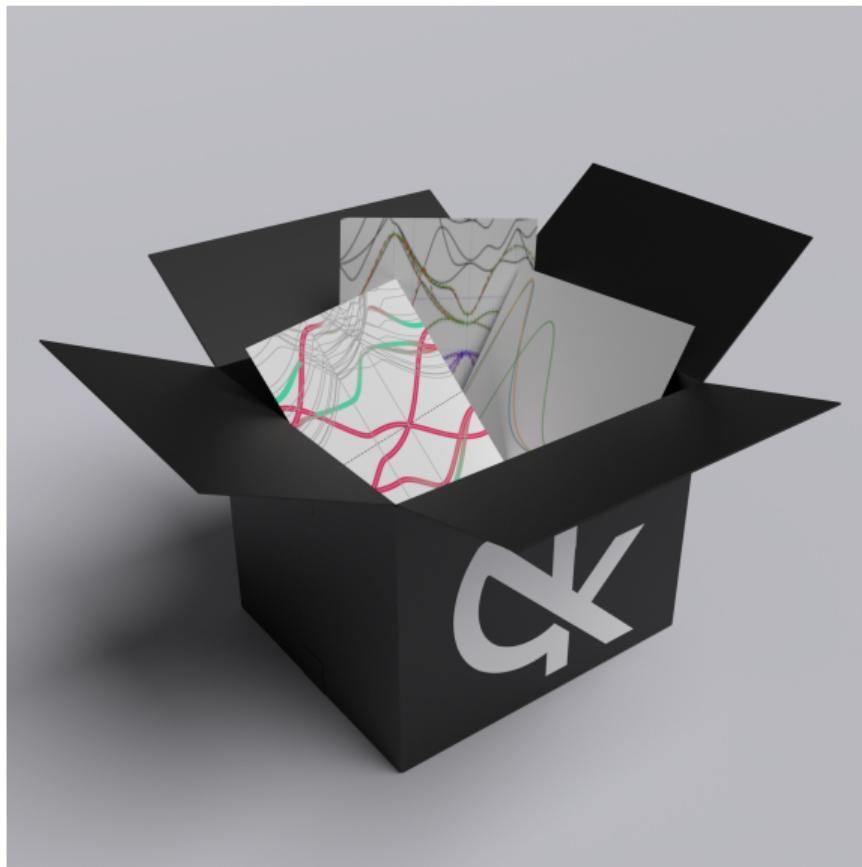
In order to evaluate this functional, one must...

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

# What should go in the box?

In order to evaluate this functional, one must...

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

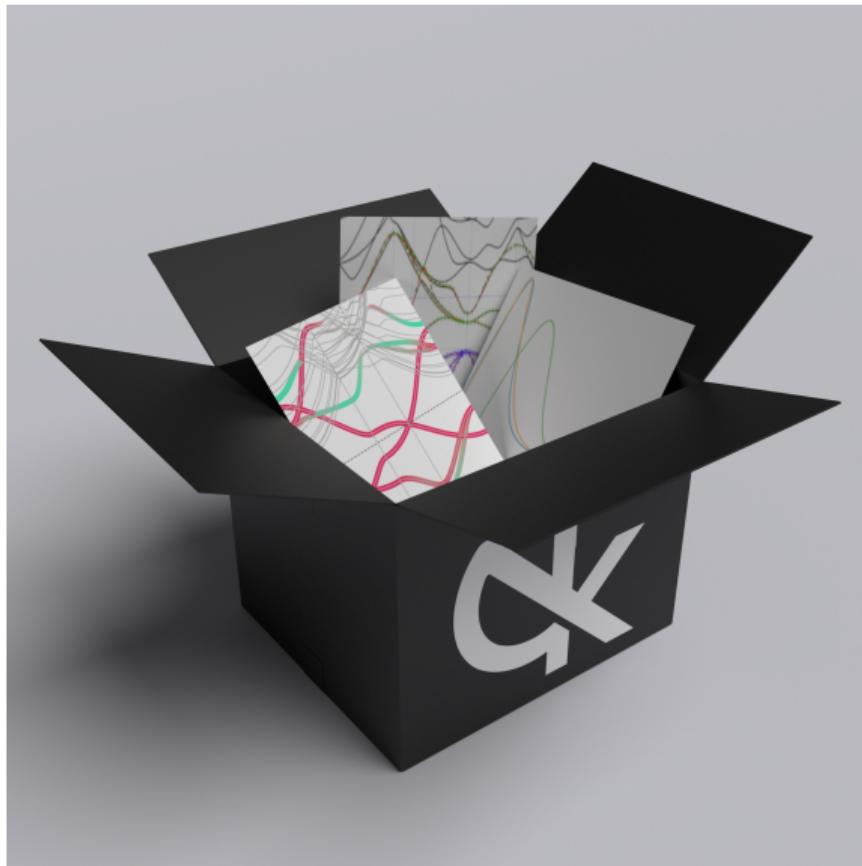


# What should go in the box?

In order to evaluate this functional, one must...

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

For a black box, we need...

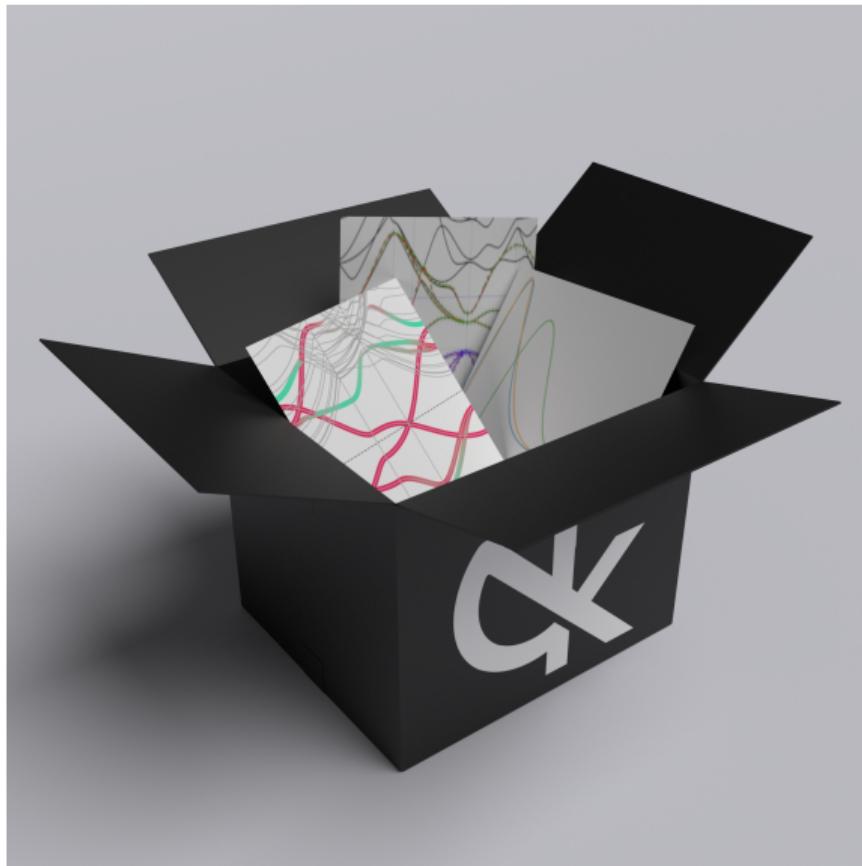


# What should go in the box?

In order to evaluate this functional, one must...

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

For a black box, we need...

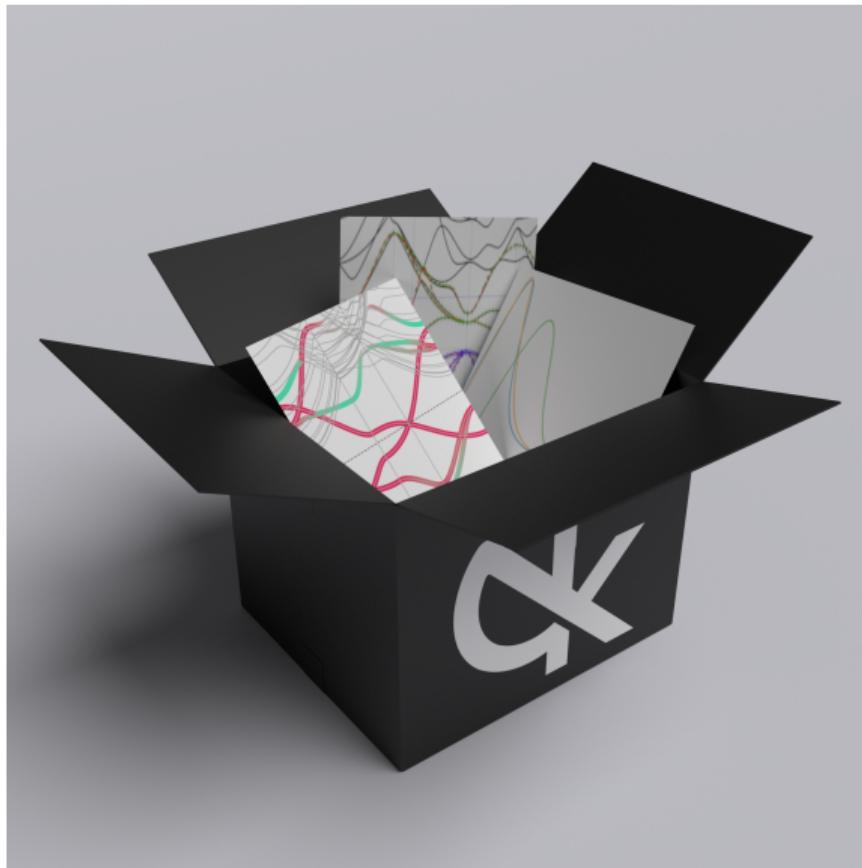


# What should go in the box?

In order to evaluate this functional, one must...

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

For a black box, we need...

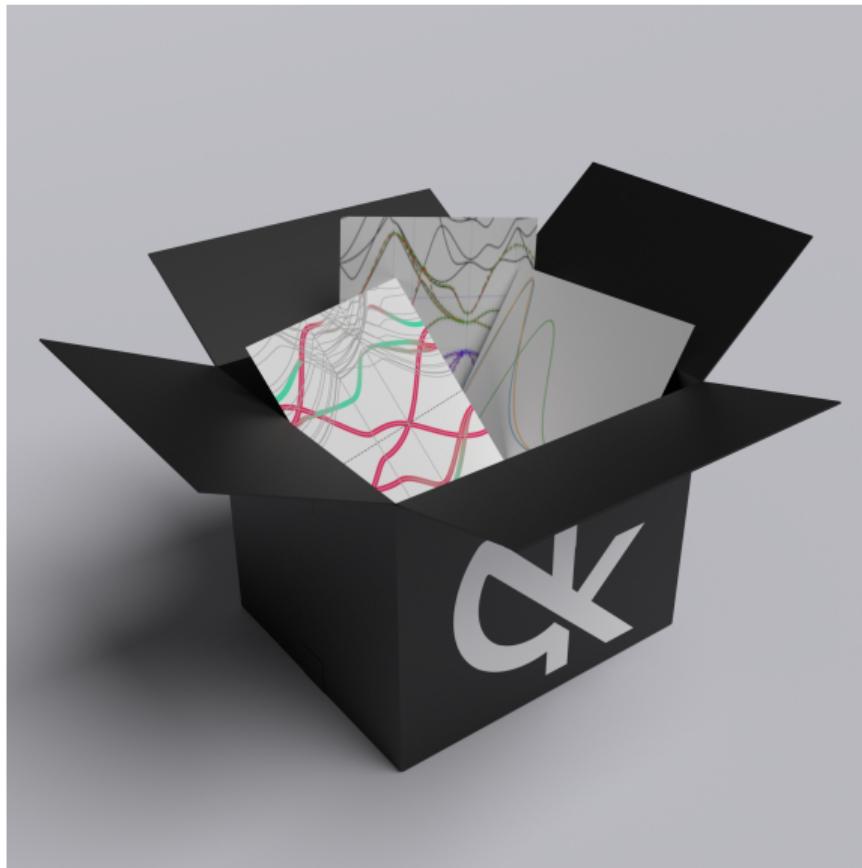


# What should go in the box?

In order to evaluate this functional, one must...

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

For a black box, we need...

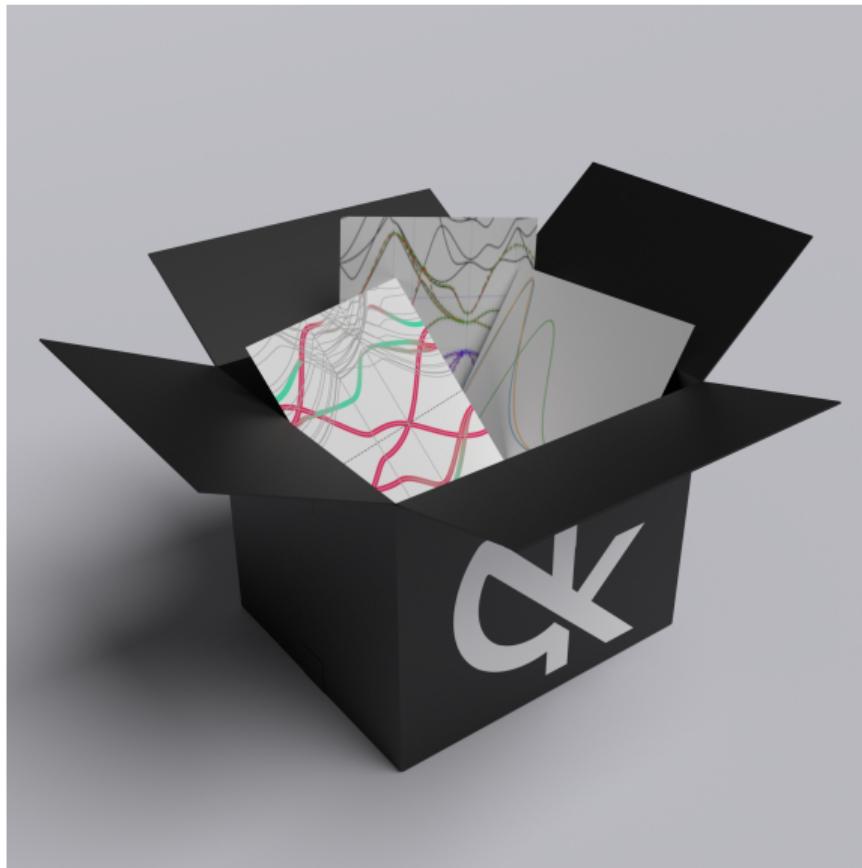


# What should go in the box?

In order to evaluate this functional, one must...

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

For a black box, we need...



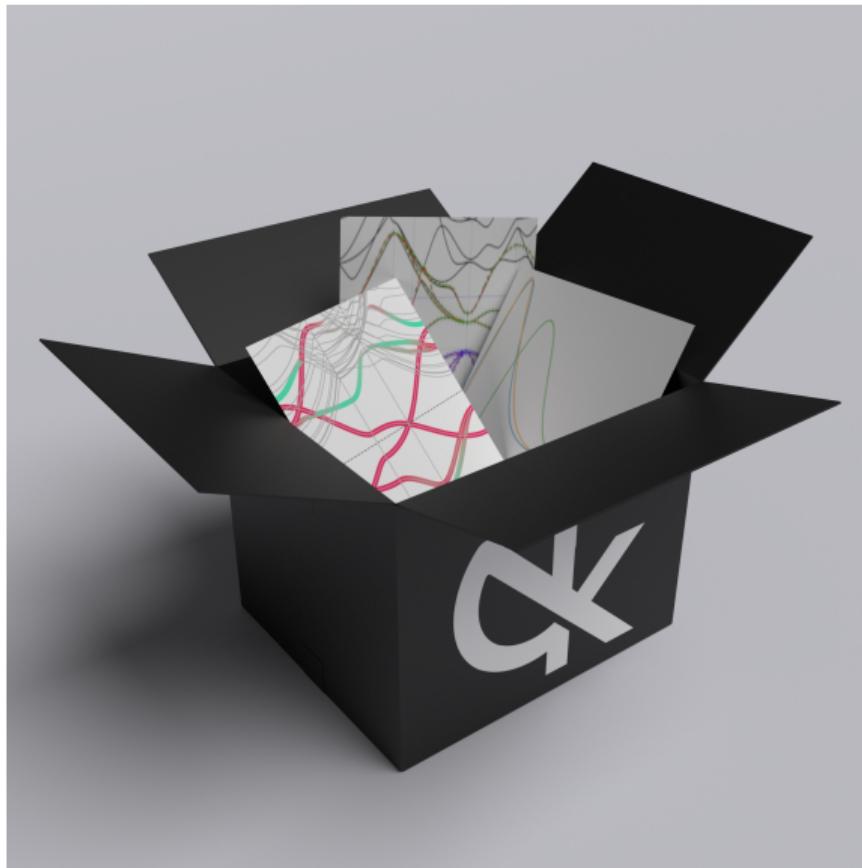
# What should go in the box?

In order to evaluate this functional, one must...

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

For a black box, we need...

- automated start-to-finish calculations



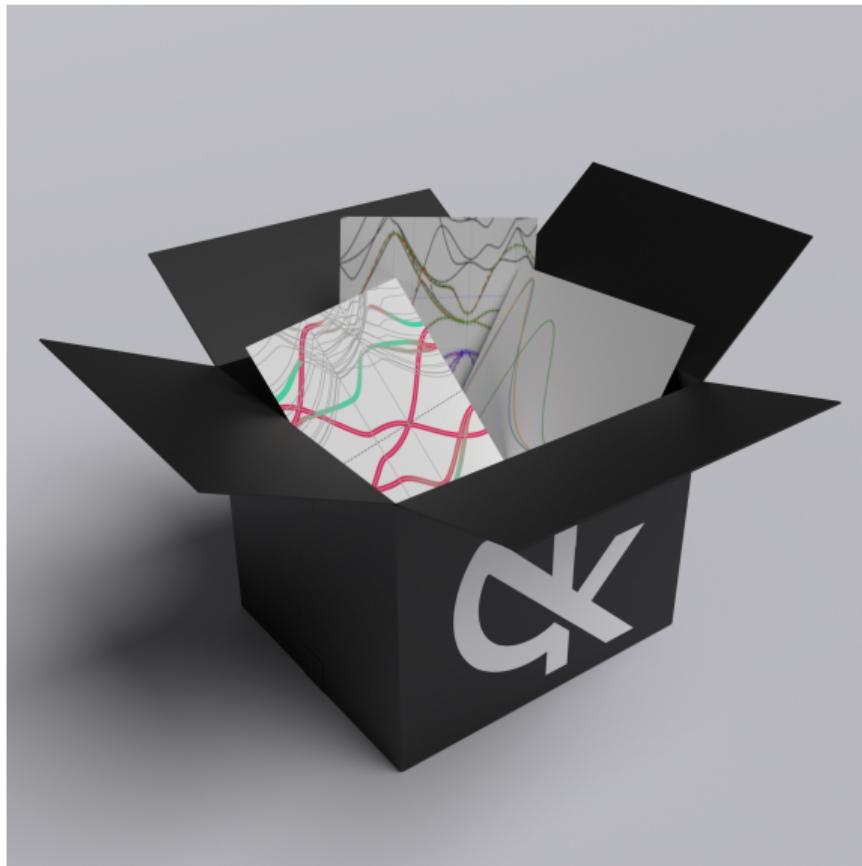
# What should go in the box?

In order to evaluate this functional, one must...

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

For a black box, we need...

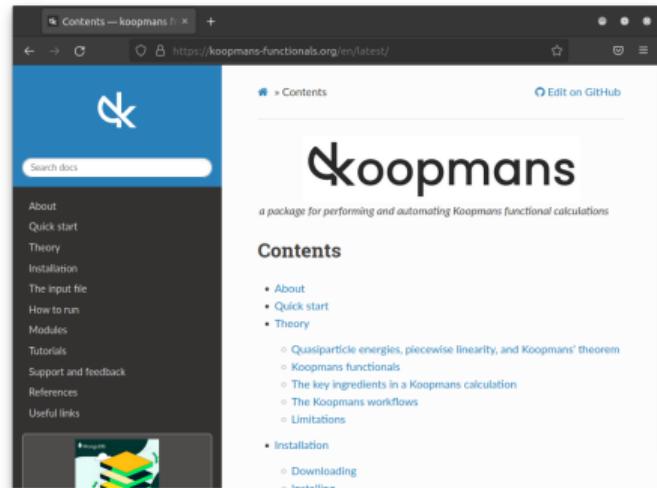
- automated start-to-finish calculations
- minimal human input



# koopmans

- v1.0 released last year<sup>1</sup>
- implementations of Koopmans functionals within Quantum ESPRESSO
- automated workflows
  - Koopmans calculations
  - Wannierization
  - dielectric tensor
  - ...
- built on top of ASE<sup>2</sup>
- does not require expert knowledge

koopmans-functionals.org



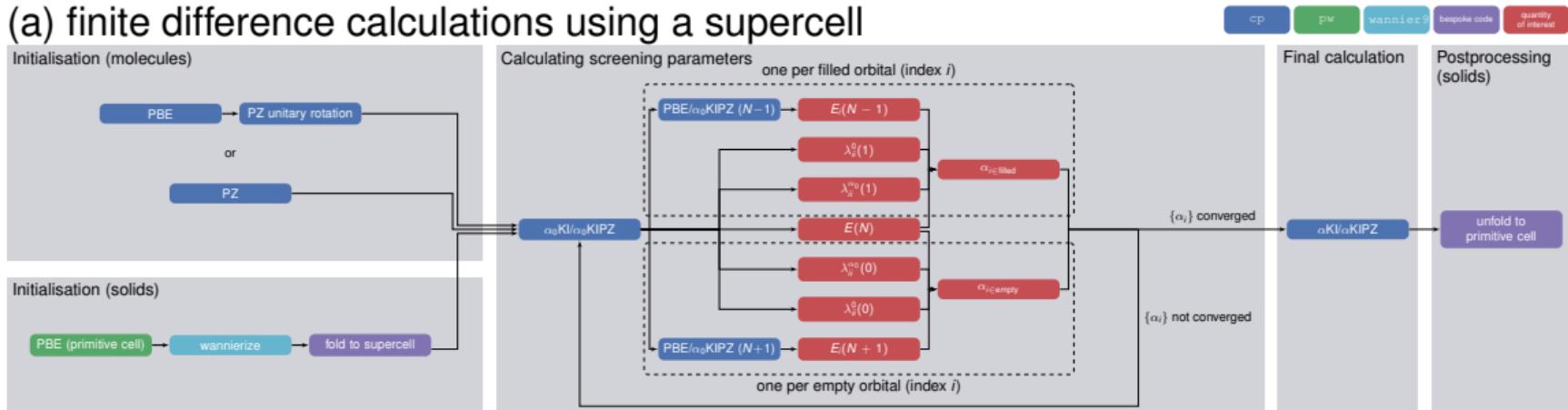
<sup>1</sup> E. B. Linscott et al. *J. Chem. Theory Comput.* 19.20 (2023), 7097

<sup>2</sup> A. H. Larsen et al. *Journal of Physics: Condensed Matter* 29.27 (2017), 273002



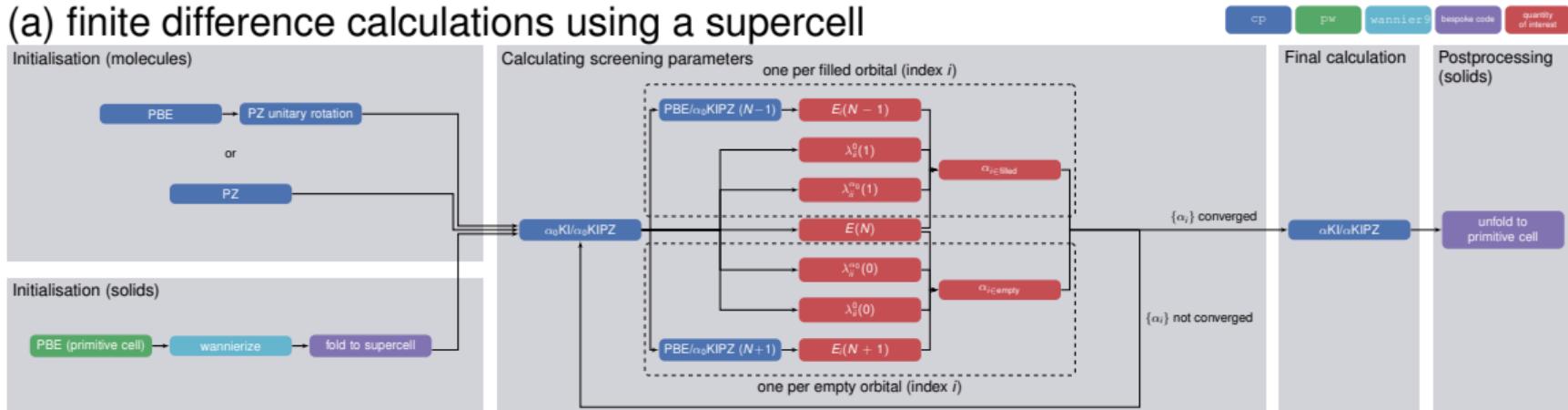
# Workflows

## (a) finite difference calculations using a supercell

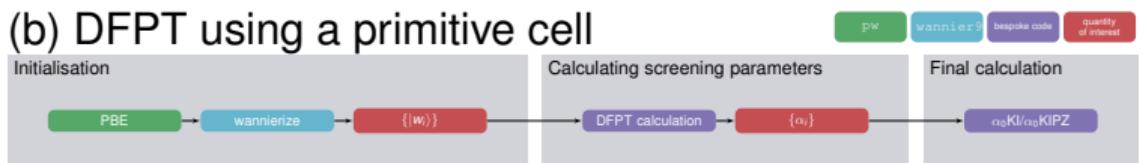


# Workflows

## (a) finite difference calculations using a supercell

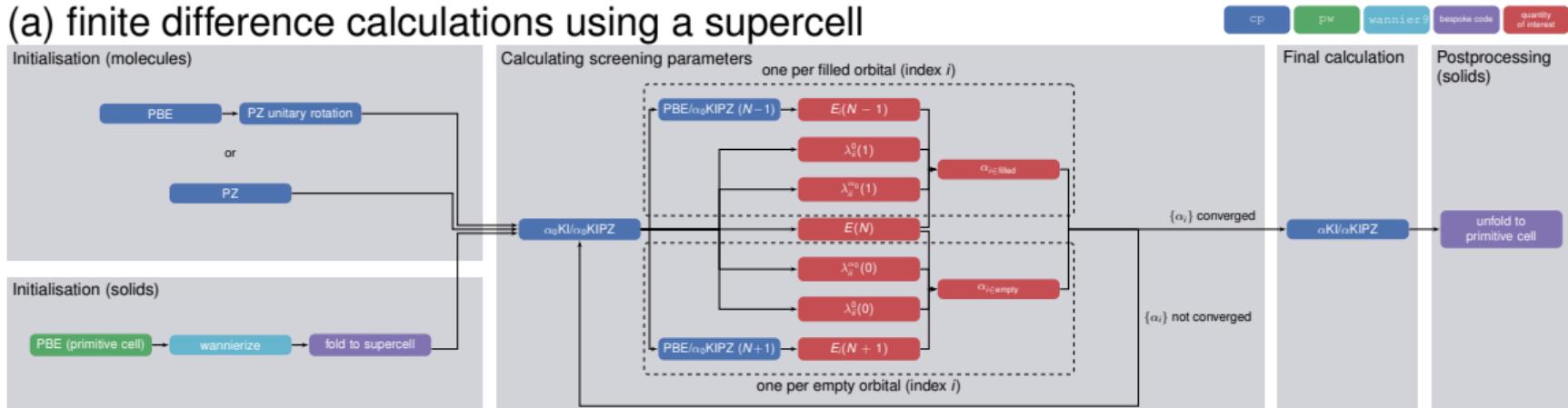


## (b) DFPT using a primitive cell

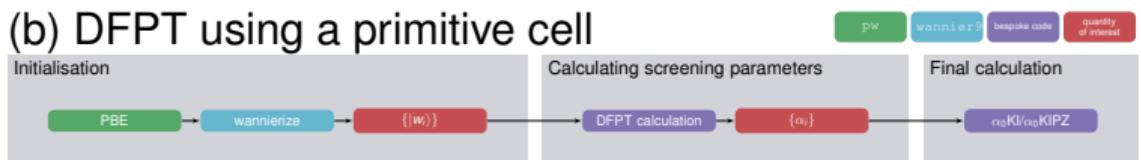


# Workflows

## (a) finite difference calculations using a supercell



## (b) DFPT using a primitive cell



All implemented in Koopmans

# Workflows

What still stands in our way? Take the example of silicon:

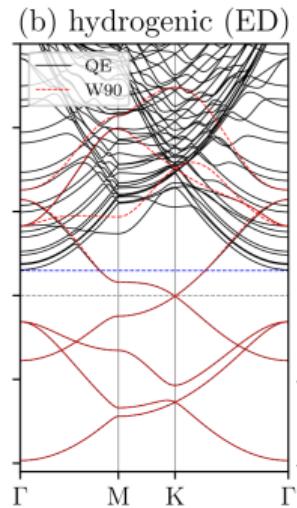
# Workflows

What still stands in our way? Take the example of silicon:

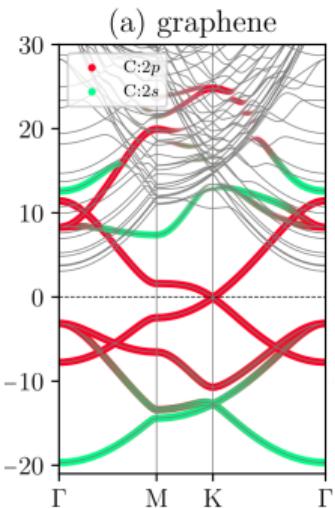
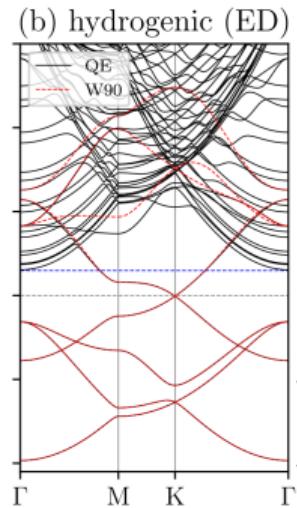
```
{  
  "workflow": {  
    "task": "singlepoint",  
    "functional": "ki",  
    "base_functional": "lda",  
    "method": "dfpt",  
    "pseudo_library": "pseudo_dojo_standard"},  
  "atoms": {  
    "cell_parameters": {"periodic": true, "ibrav": 2, "celldms": {"1": 10.2622}},  
    "atomic_positions": {  
      "units": "crystal",  
      "positions": [[{"Si": 0.00, 0.00, 0.00}, {"Si": 0.25, 0.25, 0.25}]]},  
  "kpoints": {"grid": [8, 8, 8]},  
  "calculator_parameters": {  
    "ecutwfc": 60.0,  
    "pw": {"nbnd": 20},  
    "w90": {  
      "projections": [[[{"fsite": [0.25, 0.25, 0.25], "ang_mtm": "sp3"}],  
                      [{"fsite": [0.25, 0.25, 0.25], "ang_mtm": "sp3"}]],  
      "dis_froz_max": 10.6,  
      "dis_win_max": 16.9}}}}
```

One very manual step: Wannierization. Can we automate this?

One very manual step: Wannierization. Can we automate this?

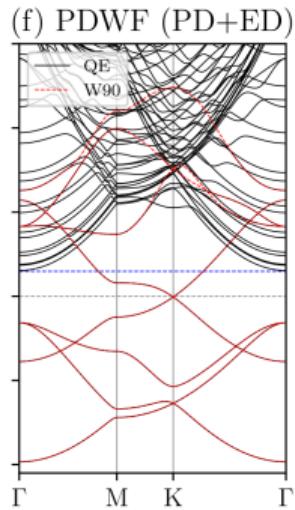
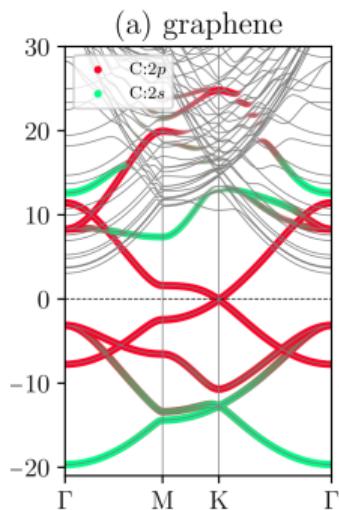
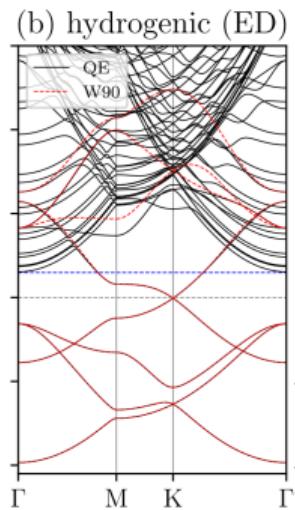


One very manual step: Wannierization. Can we automate this?

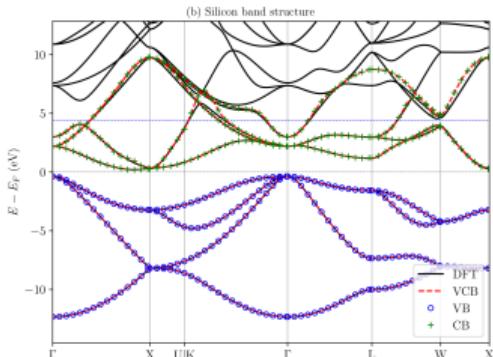


# Automating Wannierization

One very manual step: Wannierization. Can we automate this?



Aside: we must additionally use a parallel transport algorithm to separate the occupied and empty manifolds



Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability
- they serve as our initial guesses for the Wannier functions

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

| element | electronic configuration |
|---------|--------------------------|
| Li      | $1s^2 2s^1$              |

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

| element | electronic configuration | PAOs   |
|---------|--------------------------|--------|
| Li      | $1s^2 2s^1$              | 1s, 2s |

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

| element | electronic configuration | PAOs   | number of PAOs |
|---------|--------------------------|--------|----------------|
| Li      | $1s^2 2s^1$              | 1s, 2s | 2              |

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

| element | electronic configuration | PAOs     | number of PAOs |
|---------|--------------------------|----------|----------------|
| Li      | $1s^2 2s^1$              | $1s, 2s$ | 2              |
| F       | $[1s^2] 2s^2 2p^5$       |          |                |

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

| element | electronic configuration | PAOs                   | number of PAOs |
|---------|--------------------------|------------------------|----------------|
| Li      | $1s^2 2s^1$              | $1s, 2s$               | 2              |
| F       | $[1s^2] 2s^2 2p^5$       | $2s, 2p_x, 2p_y, 2p_z$ | 4              |

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

| element | electronic configuration | PAOs                   | number of PAOs |
|---------|--------------------------|------------------------|----------------|
| Li      | $1s^2 2s^1$              | $1s, 2s$               | 2              |
| F       | $[1s^2] 2s^2 2p^5$       | $2s, 2p_x, 2p_y, 2p_z$ | 4              |

6 Wannier functions for a system with 10 electrons

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

| element | electronic configuration | PAOs                   | number of PAOs |
|---------|--------------------------|------------------------|----------------|
| Li      | $1s^2 2s^1$              | $1s, 2s$               | 2              |
| F       | $[1s^2] 2s^2 2p^5$       | $2s, 2p_x, 2p_y, 2p_z$ | 4              |

6 Wannier functions for a system with 10 electrons = 5 occupied bands + only 1 unoccupied band!

Pseudoatomic orbitals play a dual purpose

- we use them to calculate projectability → we rely on the PAOs having high overlap with the atomic-like bands
- they serve as our initial guesses for the Wannier functions → the number of Wannier functions is limited to the number of pseudoatomic orbitals

e.g. LiF

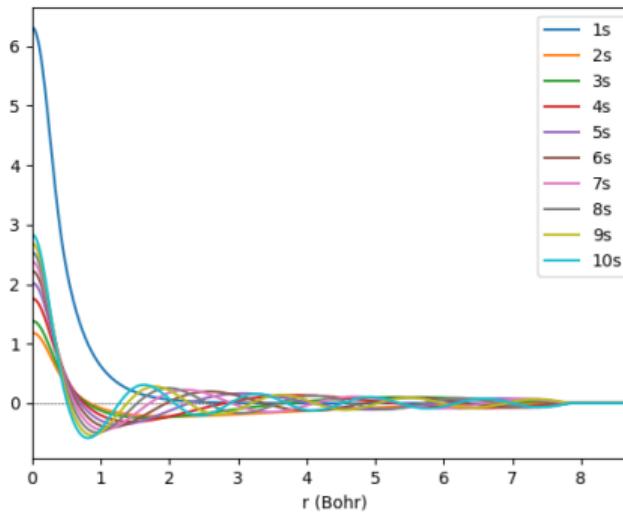
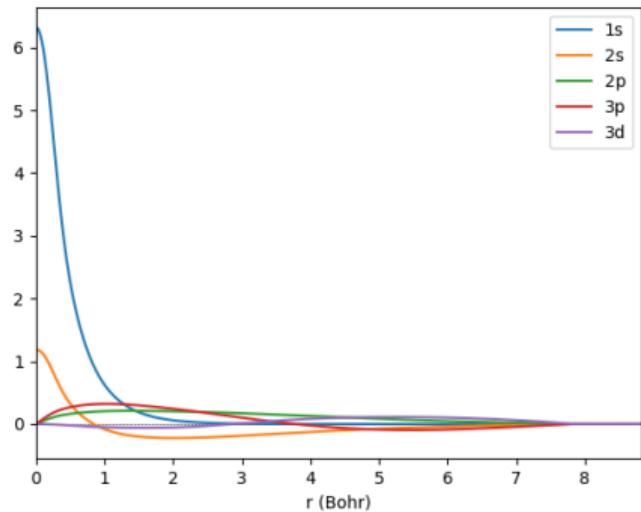
| element | electronic configuration | PAOs                   | number of PAOs |
|---------|--------------------------|------------------------|----------------|
| Li      | $1s^2 2s^1$              | $1s, 2s$               | 2              |
| F       | $[1s^2] 2s^2 2p^5$       | $2s, 2p_x, 2p_y, 2p_z$ | 4              |

6 Wannier functions for a system with 10 electrons = 5 occupied bands + only 1 unoccupied band!

If we want more Wannier functions, we're gonna need a bigger boat more PAOs...

# Automating Wannierization

Existing strategy: use the PAOs provided by OpenMX



The projectability is

$$p_{m\mathbf{k}} = \sum_n |\langle \varphi_n | \psi_{m\mathbf{k}} \rangle|^2$$

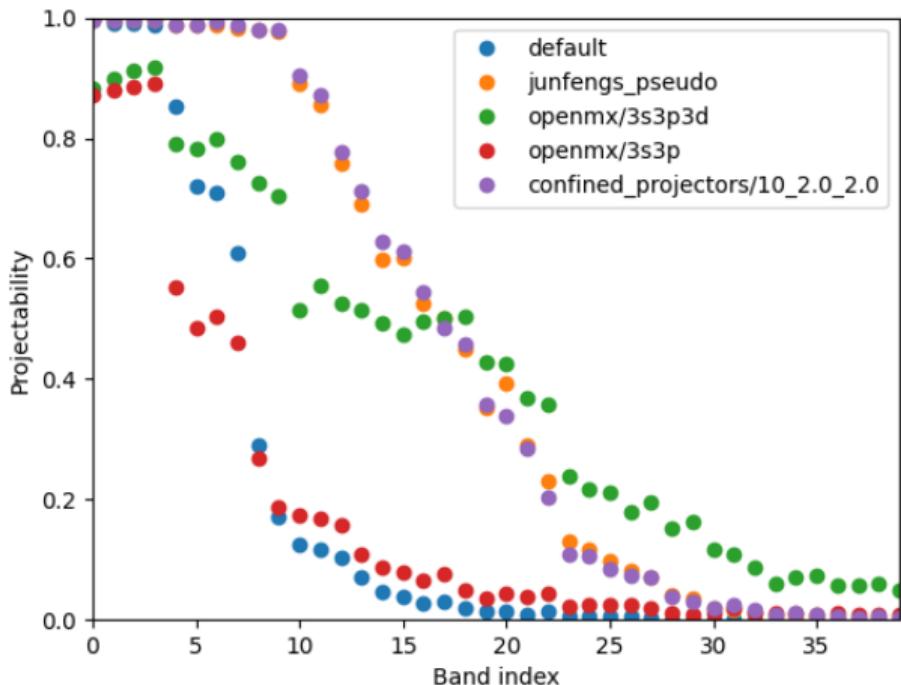
We want a set of projectors such that...

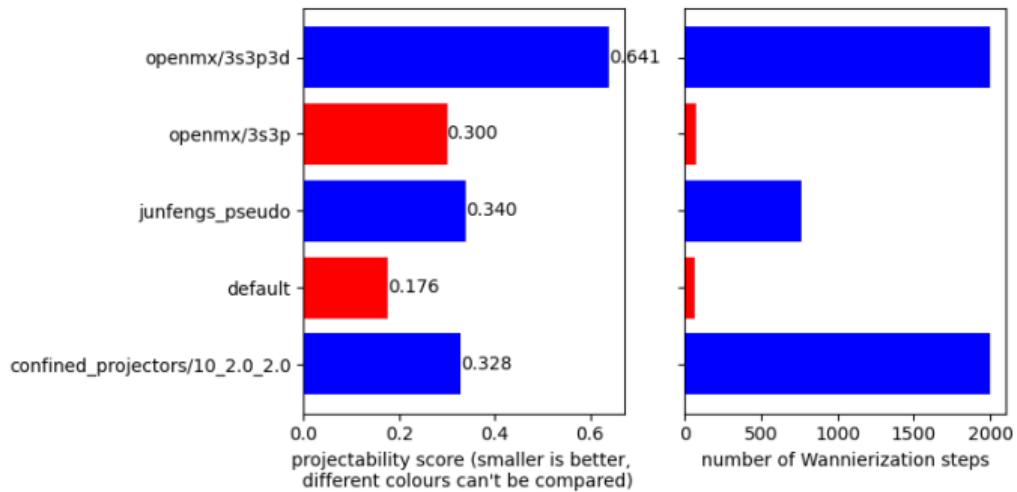
- they should have a large overlap with some Kohn-Sham states and a small overlap with all the others
- they should lie within the span of the Kohn-Sham states

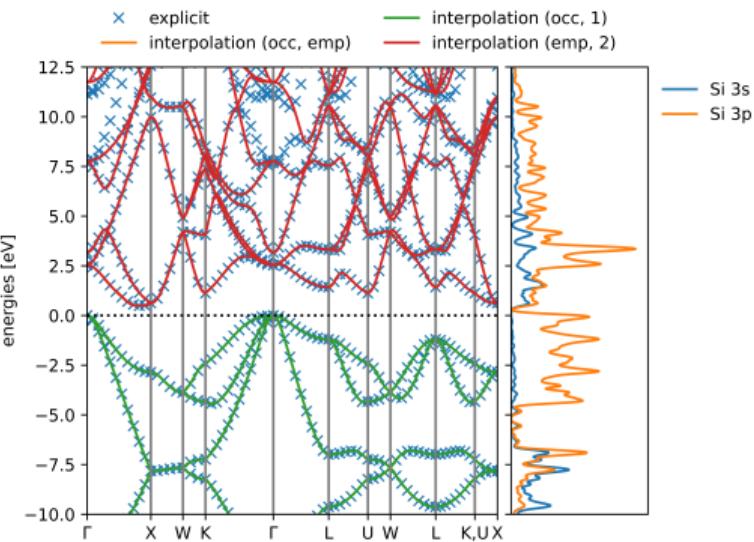
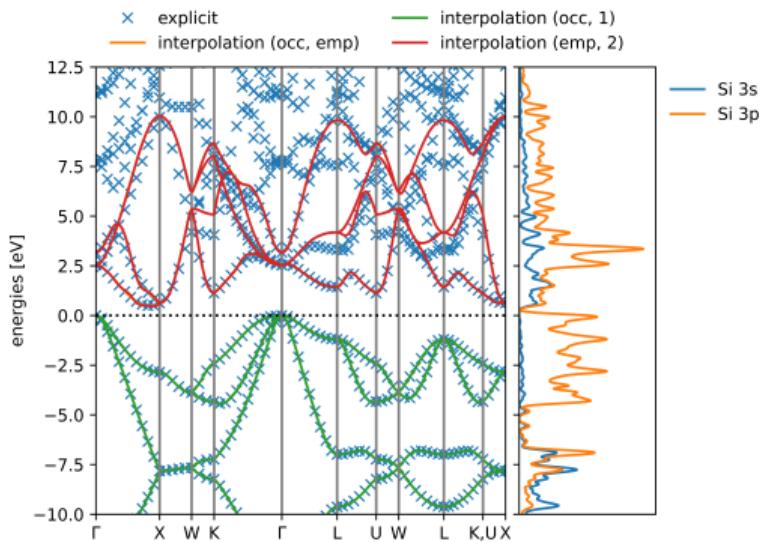
We can meet these criteria if we maximise the functional

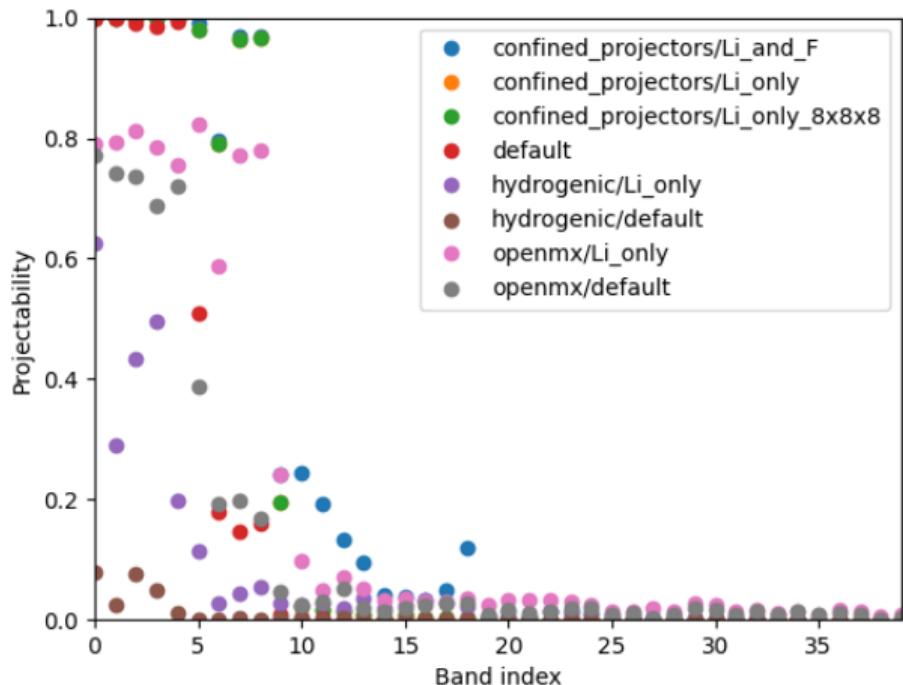
$$F[\{\varphi_n\}] = \frac{1}{N_{\mathbf{k}} N_w} \sum_{\mathbf{k}} \sum_{m \in S_{\mathbf{k}}} p_{m\mathbf{k}}$$

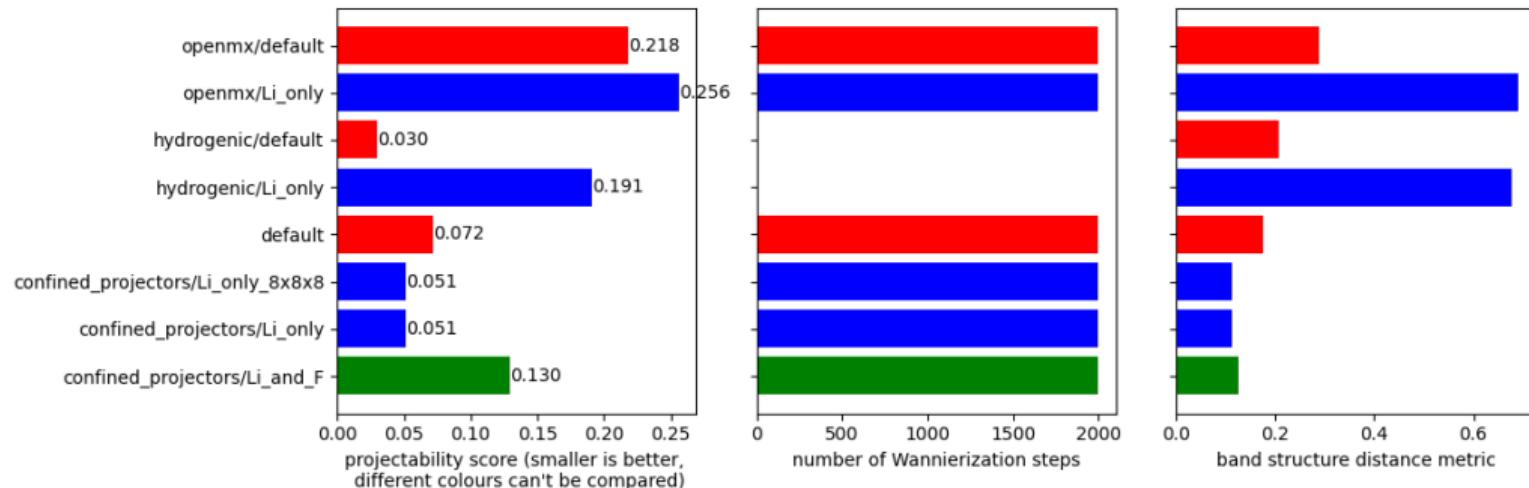
where  $S_{\mathbf{k}}$  corresponds to the  $N_w$ -largest values of  $p_{m\mathbf{k}}$ .

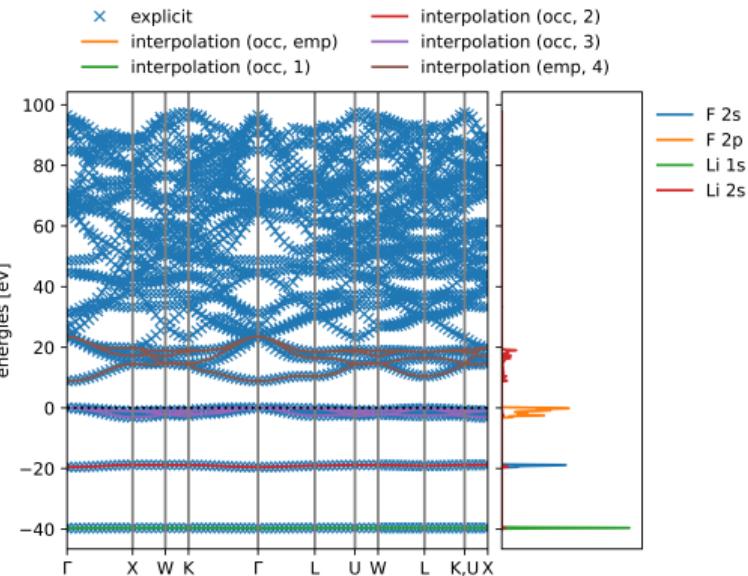
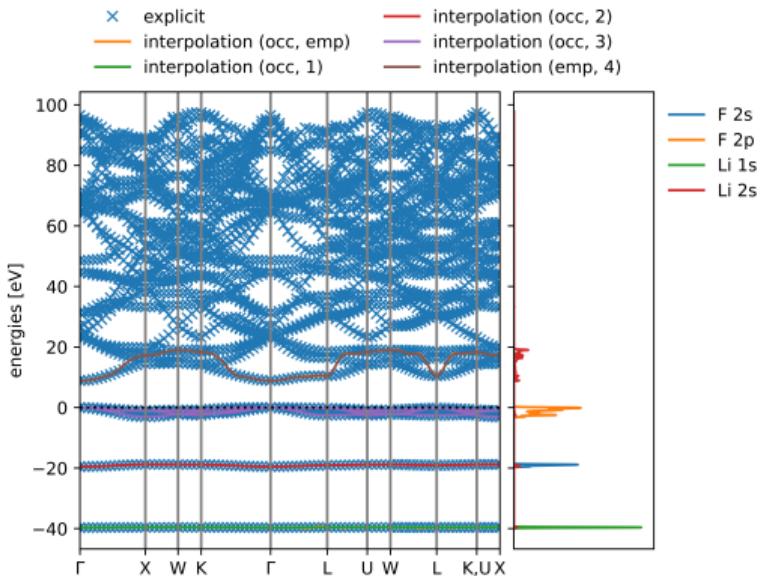












# Next steps...

- decide on a criteria for choosing the confining potential parameters (depth, width, and smoothness)

- decide on a criteria for choosing the confining potential parameters (depth, width, and smoothness)
- if the parameters are system-independent, publish a periodic table of PAOs

## Next steps...

- decide on a criteria for choosing the confining potential parameters (depth, width, and smoothness)
- if the parameters are system-independent, publish a periodic table of PAOs
- tidy and publish modified `oncvpsp.x`

## Next steps...

- decide on a criteria for choosing the confining potential parameters (depth, width, and smoothness)
- if the parameters are system-independent, publish a periodic table of PAOs
- tidy and publish modified `oncvpsp.x`
- finally do some Koopmans calculations! (Would need a good criteria for which elements to extend)

## Next steps...

- decide on a criteria for choosing the confining potential parameters (depth, width, and smoothness)
- if the parameters are system-independent, publish a periodic table of PAOs
- tidy and publish modified `oncvpsp.x`
- finally do some Koopmans calculations! (Would need a good criteria for which elements to extend)

Nice to have but not essential:

## Next steps...

- decide on a criteria for choosing the confining potential parameters (depth, width, and smoothness)
- if the parameters are system-independent, publish a periodic table of PAOs
- tidy and publish modified `oncvpsp.x`
- finally do some Koopmans calculations! (Would need a good criteria for which elements to extend)

Nice to have but not essential:

- investigate/resolve bad Wannier90 convergence

## Next steps...

- decide on a criteria for choosing the confining potential parameters (depth, width, and smoothness)
- if the parameters are system-independent, publish a periodic table of PAOs
- tidy and publish modified `oncvpsp.x`
- finally do some Koopmans calculations! (Would need a good criteria for which elements to extend)

Nice to have but not essential:

- investigate/resolve bad Wannier90 convergence
- the option to use external projectors in `projwfc.x`

The screenshot shows a browser window with two tabs open. The left tab is for the `upf-tools` repository on GitHub, and the right tab is for the `oncvpsp-tools` repository. Both tabs show the README.md file. The `upf-tools` README includes a badge for codecov at 80%, while the `oncvpsp-tools` README includes a badge for codecov at 82%. Both repositories have a "Getting Started" section with code snippets.

`upf-tools / README.md`

`elinscott` Fixing README (#4) 17 hours ago

172 lines (123 loc) · 6.54 KB

`Preview` `Code` `Blame`

## upf-tools

Tests passing · pypi v0.1.3 · python 3.8 | 3.9 | 3.10 | 3.11 · license MIT · docs passing  
codecov 80% · Cookiecutter · snekpack · code style black · Contributor Covenant 2.1

Tools for handling `.upf` (Unified Pseudopotential Format) files

### Getting Started

```
from upf_tools import UPFDict
psp = UPFDict.from_upf('/path/to/file.upf')
```

The screenshot shows a browser window with two tabs open. The left tab is for the `upf-tools` repository, and the right tab is for the `oncvpsp-tools` repository. Both tabs show the README.md file. The `oncvpsp-tools` README includes a badge for codecov at 82%, while the `upf-tools` README includes a badge for codecov at 80%. Both repositories have a "Getting Started" section with code snippets.

`oncvpsp-tools / README.md`

`elinscott` Fixing README badges (#2) yesterday

195 lines (142 loc) · 7.45 KB

`Preview` `Code` `Blame`

## oncvpsp-tools

Tests passing · pypi v0.0.2 · python 3.8 | 3.9 | 3.10 | 3.11 · license MIT · docs passing  
codecov 82% · Cookiecutter · snekpack · code style black · Contributor Covenant 2.1

Tools for handling input and output files of `oncvpsp.x`

### Getting Started

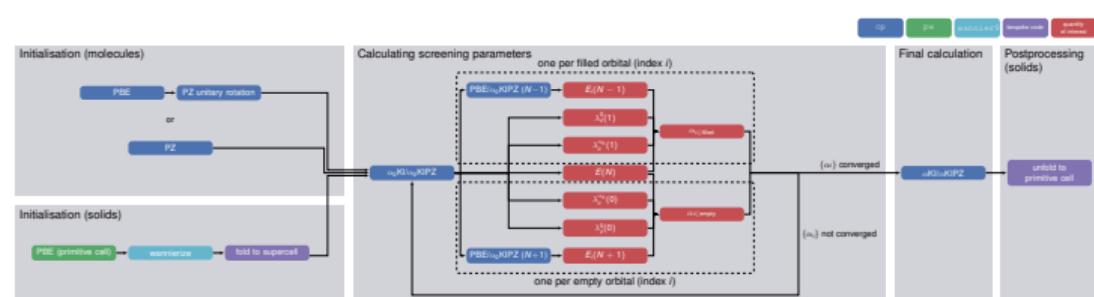
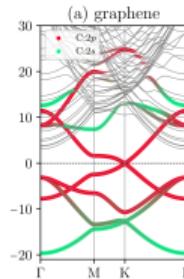
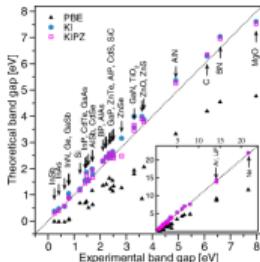
```
oncvpsp-tools allows you to inspect input files
```

```
from upf_tools import ONCVPSPInput
```

## Publish your code!

- It's easy: most of the infrastructure was from the `cookiecutter` I used (see my July 2022 GM)
- It avoids duplication: `upf-tools` was discovered by Marnik and is now used in AiIDA

# Take home messages



- Koopmans functionals yield band structures with comparable accuracy to state-of-the-art GW
- the release of koopmans means non-experts can now use Koopmans functionals in their own research
- work is ongoing to automate the Wannierization bottleneck

# Acknowledgements



Nicola Marzari



Nicola Colonna



Riccardo De Gennaro

Junfeng Qiao



**Swiss National  
Science Foundation**

**MARVEL**  
The logo for MARVEL consists of four red hexagons arranged in a horizontal row, with the first two being solid red and the last two being semi-transparent red.

NATIONAL CENTRE OF COMPETENCE IN RESEARCH

slides available at  [github/elinscott-talks](https://github.com/elinscott-talks) and on the THEOS wiki

# SPARE SLIDES

# Accurate band structures

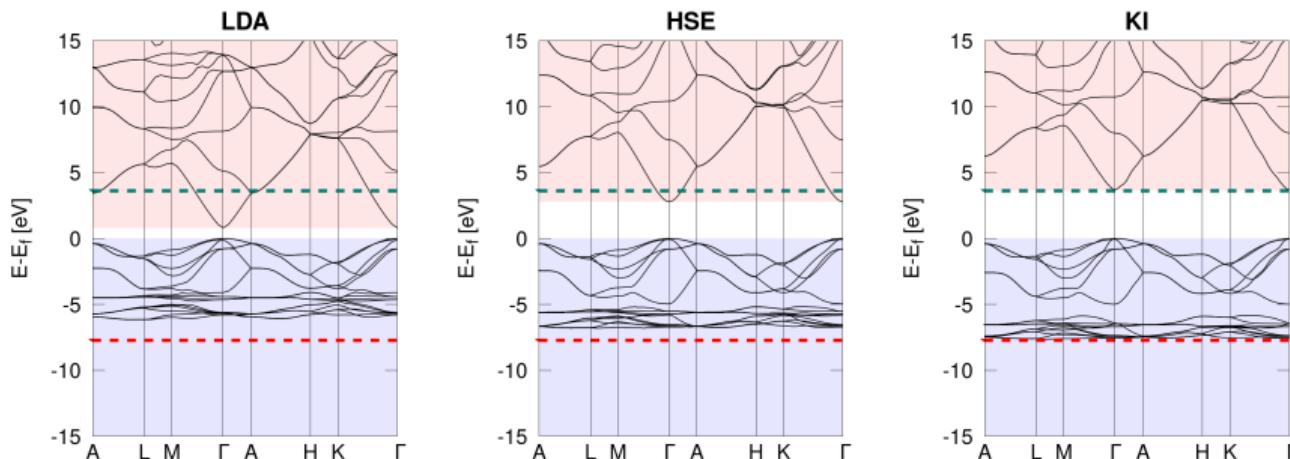
|  | PBE   | $G_0W_0^1$ | scGW <sup>2</sup> | KI@[PBE,MLWFs] | KIPZ@PBE | exp <sup>3</sup> |
|--|-------|------------|-------------------|----------------|----------|------------------|
| $E_g$                                    | 0.49  | 1.06       | 1.14              | 1.16           | 1.15     | 1.17             |
| $\Gamma_{1v} \rightarrow \Gamma_{25'v}$  | 11.97 | 12.04      |                   | 11.97          | 12.09    | $12.5 \pm 0.6$   |
| $X_{1v} \rightarrow \Gamma_{25'v}$       | 7.82  |            |                   | 7.82           |          | 7.75             |
| $X_{4v} \rightarrow \Gamma_{25'v}$       | 2.85  | 2.99       |                   | 2.85           | 2.86     | 2.90             |
| $L_{2'v} \rightarrow \Gamma_{25'v}$      | 9.63  | 9.79       |                   | 9.63           | 9.74     | $9.3 \pm 0.4$    |
| $L_{1v} \rightarrow \Gamma_{25'v}$       | 6.98  | 7.18       |                   | 6.98           | 7.04     | $6.8 \pm 0.2$    |
| $L_{3'v} \rightarrow \Gamma_{25'v}$      | 1.19  | 1.27       |                   | 1.19           |          | $1.2 \pm 0.2$    |
| $\Gamma_{25'v} \rightarrow \Gamma_{15c}$ | 2.48  | 3.29       |                   | 3.17           | 3.20     | $3.35 \pm 0.01$  |
| $\Gamma_{25'v} \rightarrow \Gamma_{2'c}$ | 3.28  | 4.02       |                   | 3.95           | 3.95     | $4.15 \pm 0.05$  |
| $\Gamma_{25'v} \rightarrow X_{1c}$       | 0.62  | 1.38       |                   | 1.28           | 1.31     | 1.13             |
| $\Gamma_{25'v} \rightarrow L_{1c}$       | 1.45  | 2.21       |                   | 2.12           | 2.13     | $2.04 \pm 0.06$  |
| $\Gamma_{25'v} \rightarrow L_{3c}$       | 3.24  | 4.18       |                   | 3.91           | 3.94     | $3.9 \pm 0.1$    |
| MSE                                      | 0.35  | 0.02       |                   | 0.01           | 0.03     |                  |
| MAE                                      | 0.44  | 0.21       |                   | 0.14           | 0.17     |                  |

<sup>1</sup> M. Shishkin et al. *Phys. Rev. Lett.* 99.24 (2007), 246403 for  $E_g$  and M. S. Hybertsen et al. *Phys. Rev. B* 34.8 (1986), 5390 for the transitions;

<sup>2</sup> M. Shishkin et al. *Phys. Rev. B* 75.23 (2007), 235102.

<sup>3</sup> O. Madelung. *Semiconductors*. 3rd ed. Berlin: Springer-Verlag, 2004.

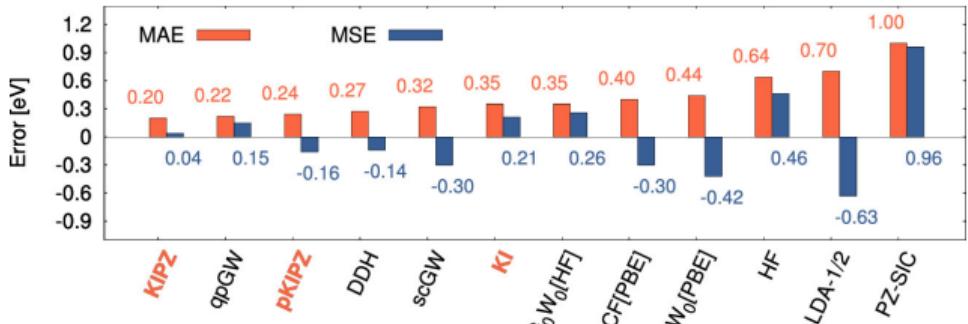
# Koopmans functionals give accurate band structures



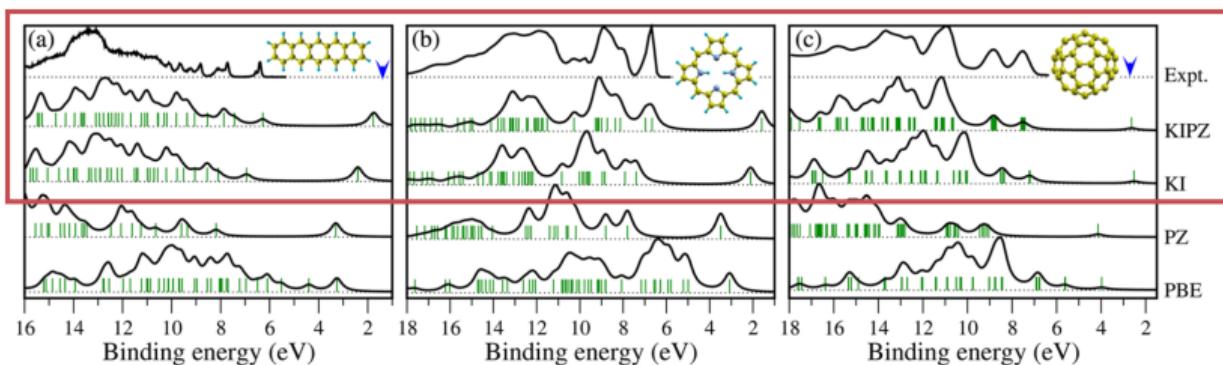
| ZnO                                  | LDA  | HSE  | $GW_0$ | $scG\tilde{W}$ | KI   | exp       |
|--------------------------------------|------|------|--------|----------------|------|-----------|
| $E_{gap}$ (eV)                       | 0.79 | 2.79 | 3.0    | 3.2            | 3.62 | 3.60      |
|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

# 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



# Koopmans functionals: results for molecules

Electron affinities =  $E(N) - E(N + 1) \stackrel{?}{=} -\varepsilon_{LU}$  of molecules cf. CCSD(T)/exp

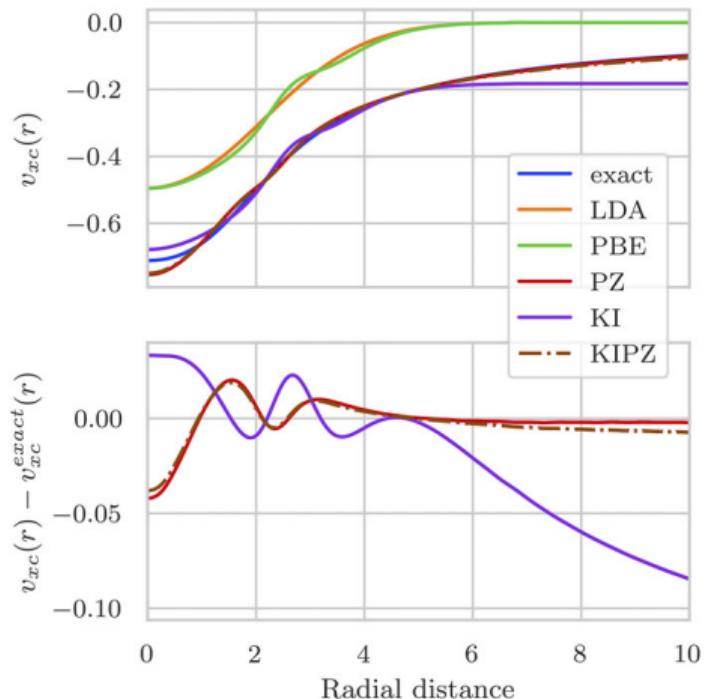
For 15 of the GW100 molecules with bound LUMOs



Linscott et al. (in prep)

# Koopmans functionals: results for toy systems

For Hooke's atom (two electrons in a harmonic confining potential with Coulombic repulsion)



# Koopmans functionals: results for toy systems

For Hooke's atom (two electrons in a harmonic confining potential with Coulombic repulsion)

