



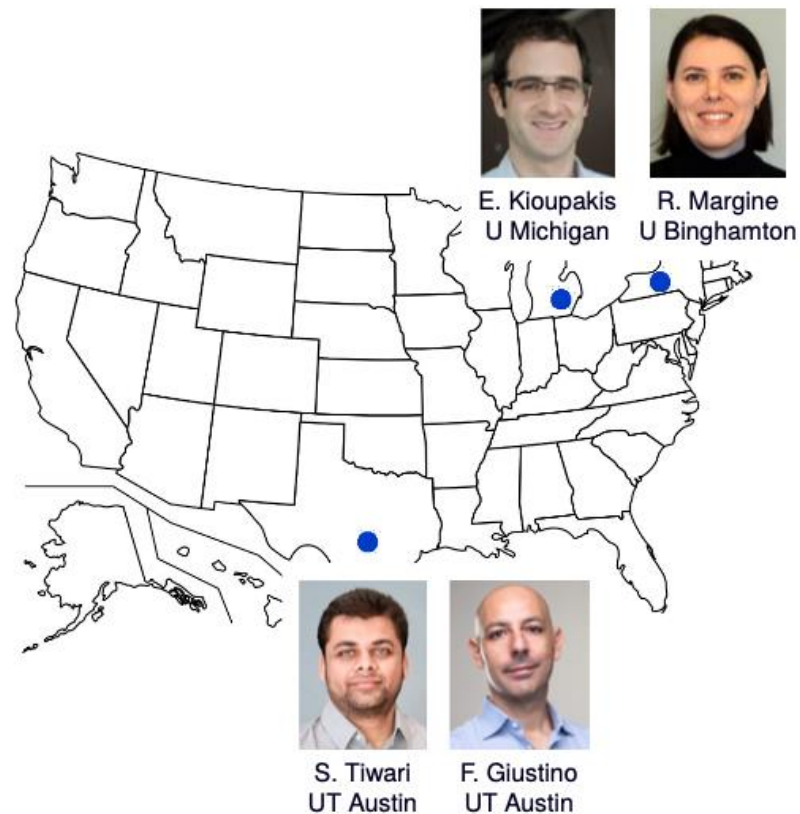
Introduction to EPWpy and MatCSSI

Sabyasachi Tiwari

Oden Institute for Computational Engineering and Sciences
UT Austin

- Electron-Phonon-Physics with EPW
- Programmable abstraction using EPWpy
- Integrated MATCSSI portal
- Conclusion

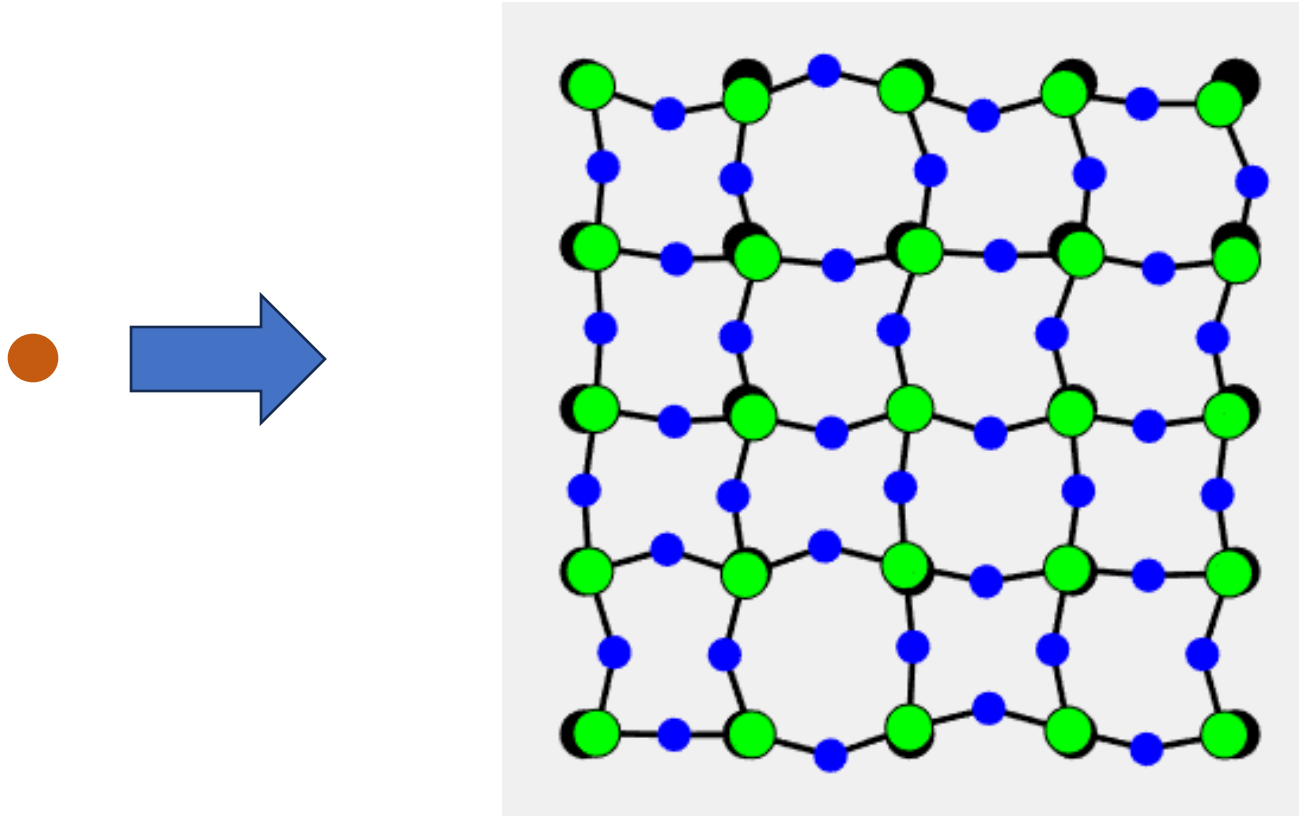
- Electron-Phonon-Physics with EPW
- Programmable abstraction using EPWpy
- Integrated MATCSSI portal
- Conclusion



Current developers

Kyle Bushick
Fabio Caruso
Jie-Cheng Chen
Zhenbang Dai
Feliciano Giustino
Viet-Anh Ha
Emmanouil Kioupakis
Jon Lafuente-Bartolomé
Tae Yun Kim
Chao Lian
Jae-Mo Lihm
Zhe Liu
Roxana Margine
Hitoshi Mori
Yiming Pan
Samuel Poncé
Danylo Radevych
Young-Woo Son
Sabyasachi Tiwari
Aidan Thorn
Shashi Mishra
Amanda Wang
Wooil Yang
Marios Zacharias
Xiao Zhang

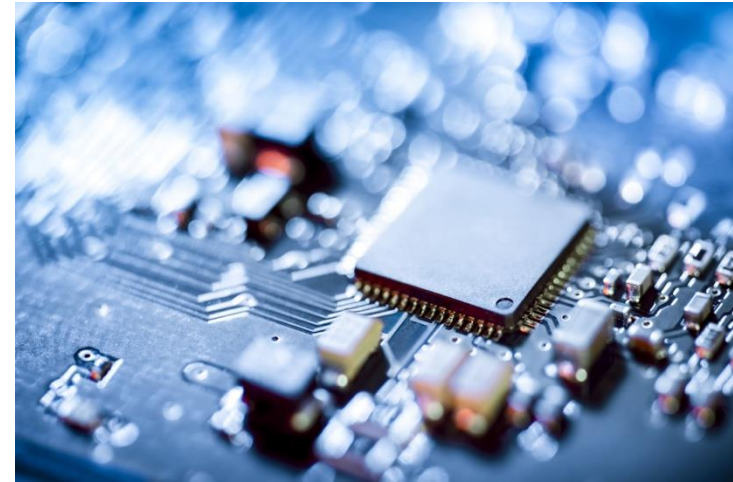
Why Electron-Phonon Physics?



- Lattice vibrations : Phonons
- Electrons/excited particles
 - Interact with lattice
 - E-P coupling

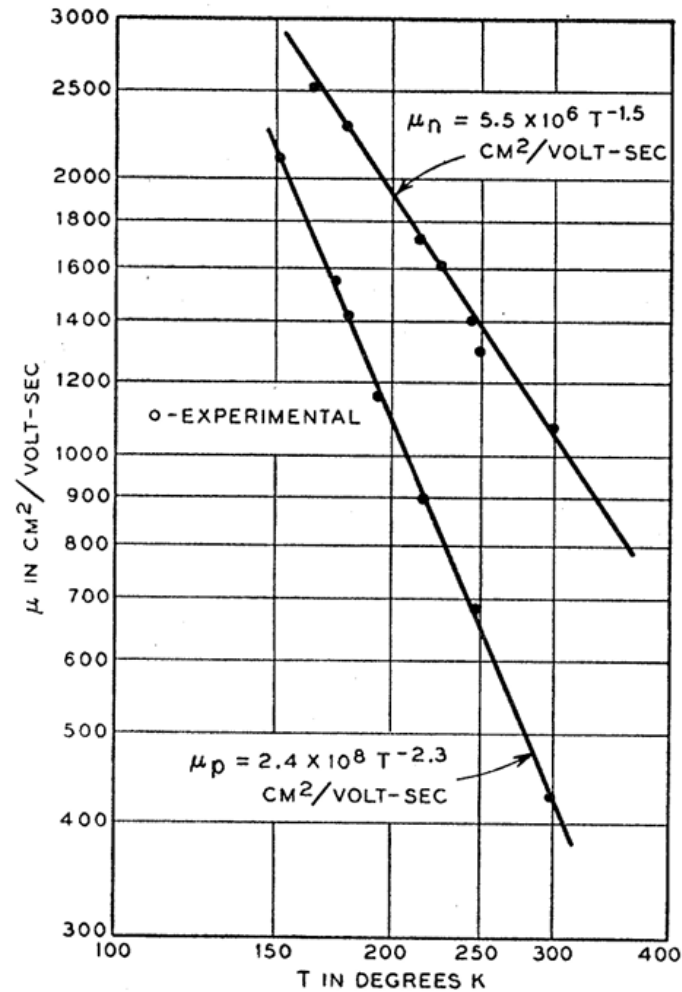
Wendt et.al., Sci. Adv. 10, 1126 (2019)

Why Electron-Phonon Physics?



With more usage (higher heat) they become slow

Why Electron-Phonon Physics?



- Transport
 - With temperature mobility goes down

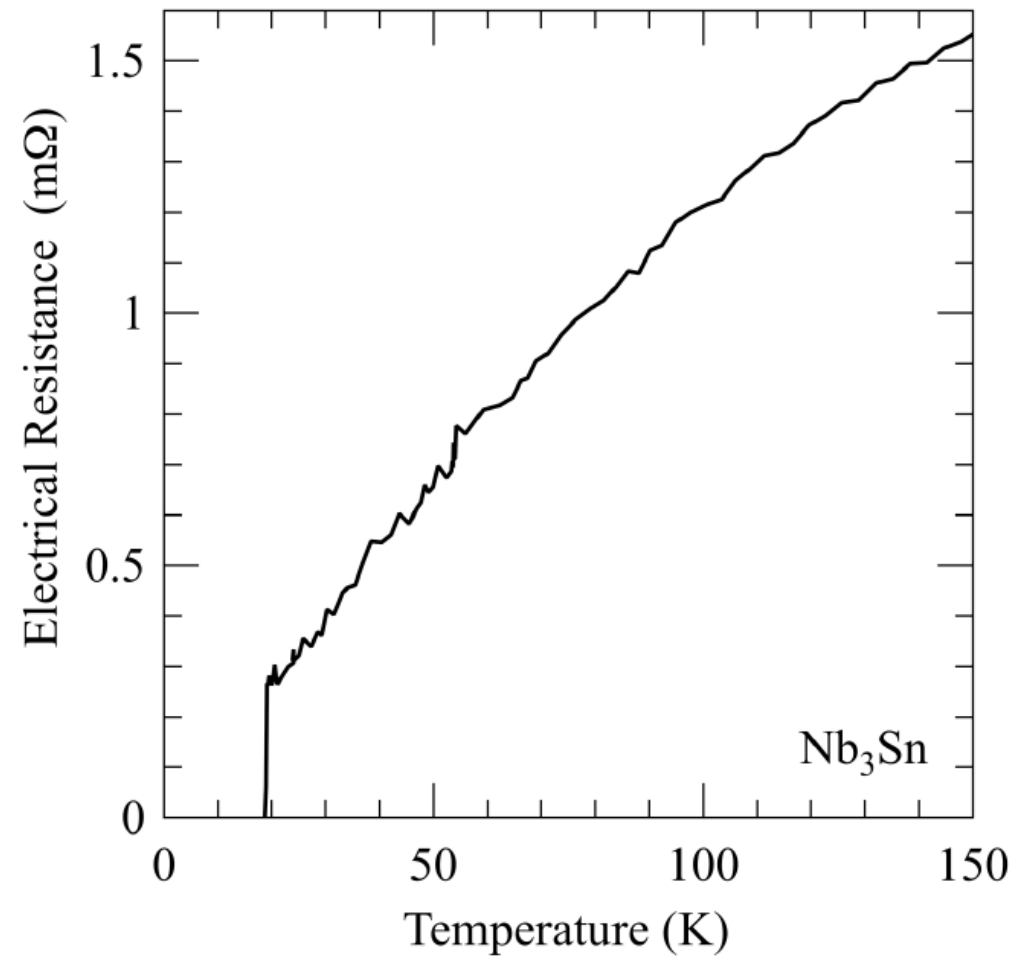
Prince et al., Phys. Rev. B. 93.1204 (1954)

Why Electron-Phonon Physics ($T \approx 0\text{ K}$)?



MRI machines use superconductors to generate magnetic field

Why Electron-Phonon Physics ($T \approx 0$ K)?



- Superconductivity

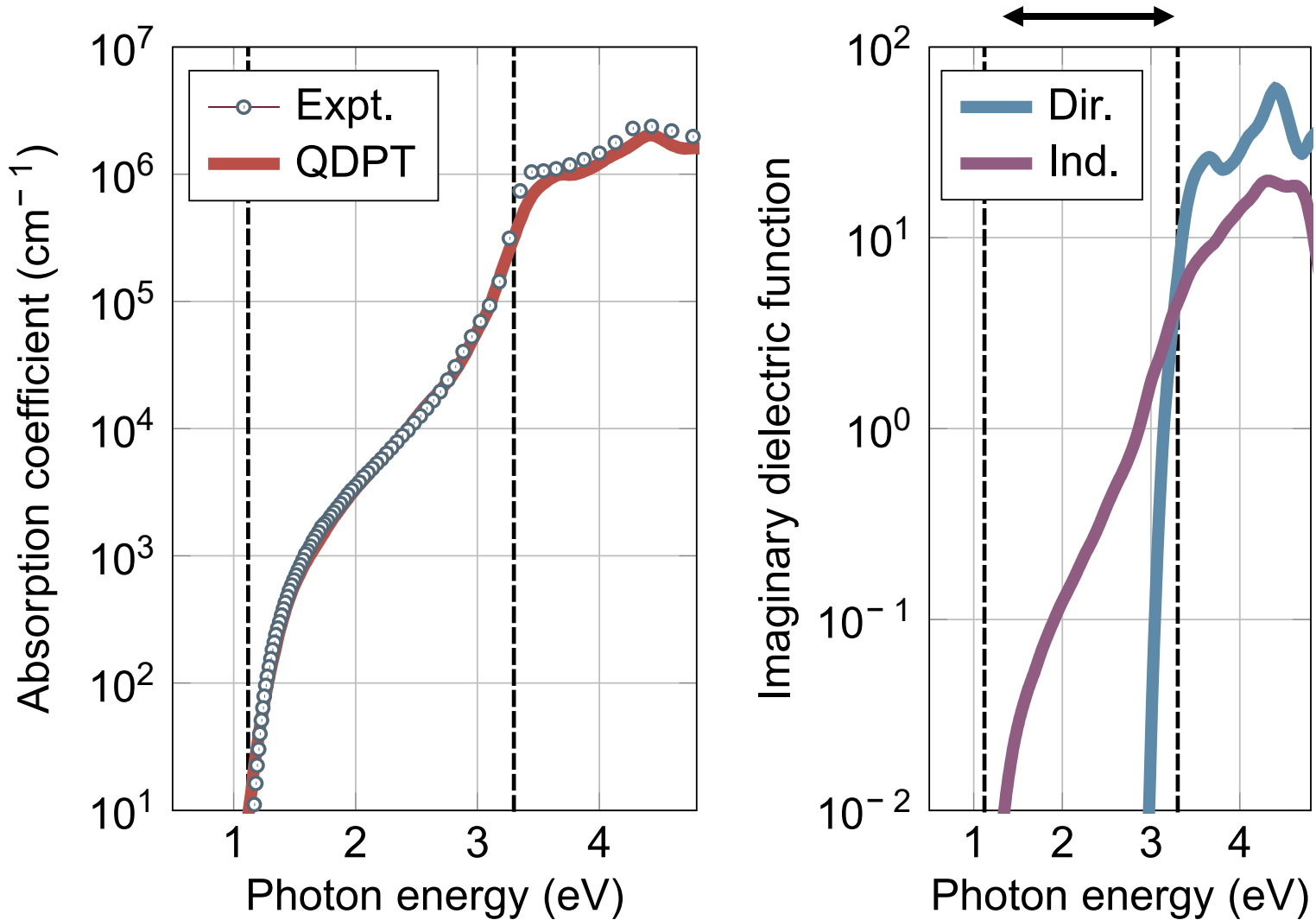
Resistance of Nb₃Sn and superconducting transition
Akimitsu group, in “Superconductors: New Developments”, 2015

Why Electron-Phonon Physics (Light-Matter interaction)?



Solar farm using silicon photovoltaics which absorb sunlight and convert to electricity

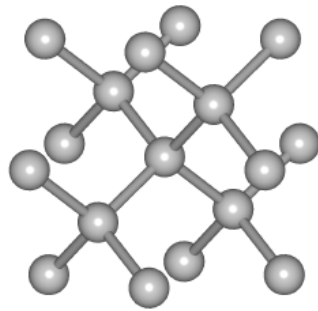
Why Electron-Phonon Physics (Light-Matter interaction)?



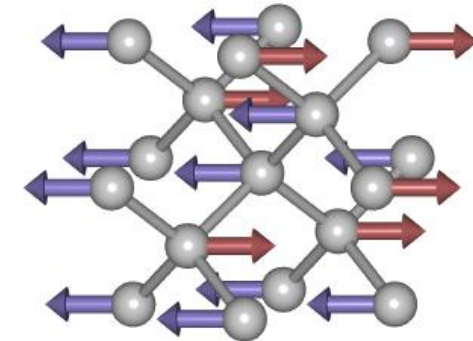
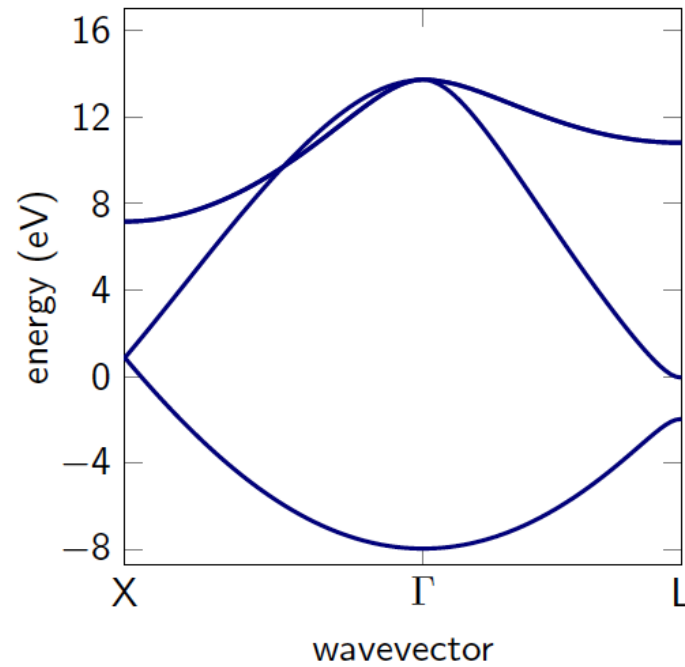
- Visible range:
Phonons

Tiwari et.al., Phys. Rev. B. 109. 195137 (2024)

Where do phonons come from?

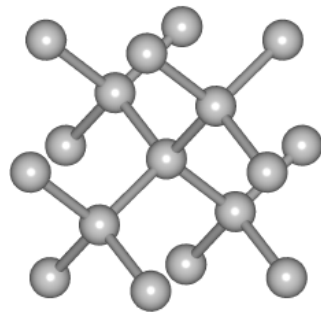


diamond

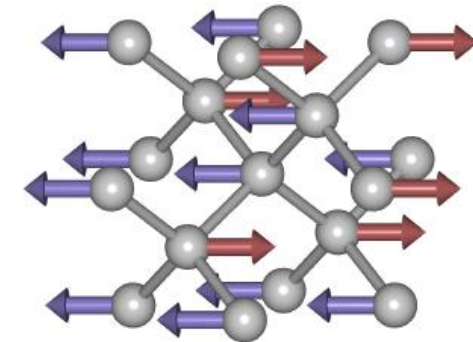
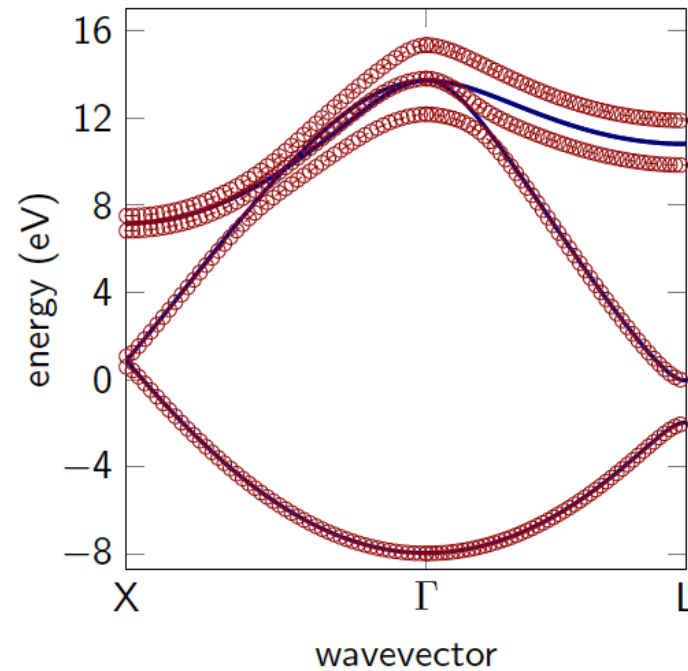


Move atoms along a mode

Where do phonons come from?



diamond



Move atoms along a mode

- Change in band structure

Where do phonons come from?

$$-\frac{\hbar^2}{2m} \nabla^2 \psi_n + V_{\text{SCF}} \psi_n = \epsilon_n \psi_n$$
$$n(r) = \sum_{n \in \text{occ}} |\psi_n|^2$$
$$V_{\text{SCF}}(r) = -\frac{e^2}{4\pi\epsilon_0} \left[\sum_{\kappa} \frac{Z_{\kappa}}{r - \tau_{\kappa}} + \int \frac{n(r') dr'}{r - r'} \right] + V_{\text{xc}}[n(r)]$$

- V_{SCF} : Depends on atomic positions (τ_{κ})

Where do phonons come from?

$$V_{\text{SCF}} = V_{\text{SCF}}(r; \tau_{\kappa_1}, \tau_{\kappa_2}, \tau_{\kappa_3} \dots \tau_{\kappa_N})$$

Displace atoms from equilibrium position: $\tau = \tau_0 + \Delta\tau$

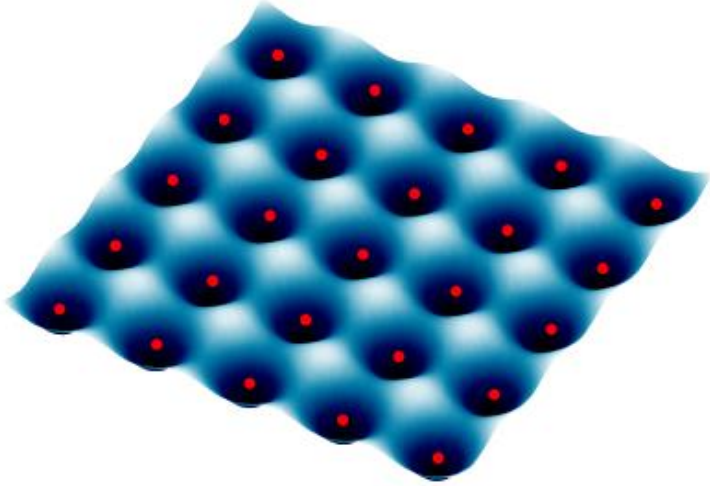
$$V_{\text{SCF}}(\tau_0 + \Delta\tau) = V_{\text{SCF}}(\tau_0) + \frac{\partial V_{\text{SCF}}}{\partial \tau} \Delta\tau + \frac{1}{2} \frac{\partial^2 V_{\text{SCF}}}{\partial \tau^2} \Delta\tau^2 + \dots$$

- Perturbation to atomic positions
 - Temperature

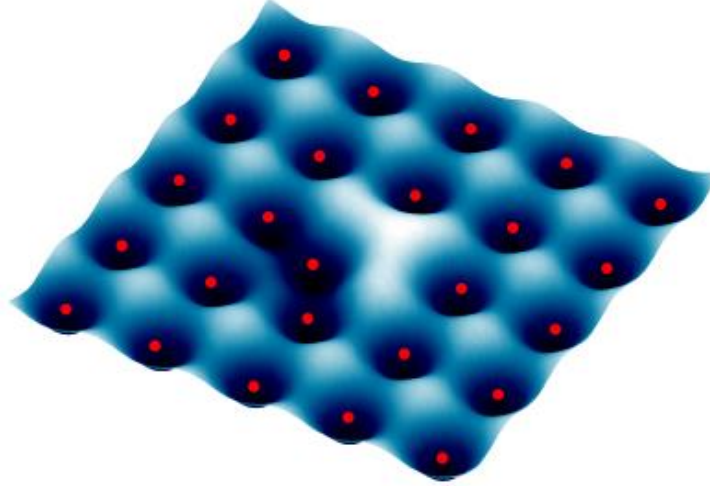
Baroni et.al., Rev. Mod. Phys. 73, 515 (2001)

Where do phonons come from?

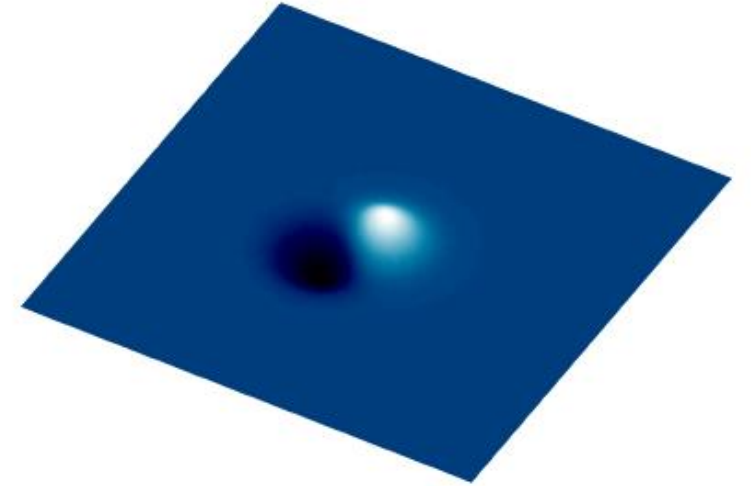
$$V_{\text{SCF}}(r; \tau)$$



$$V_{\text{SCF}}(r; \tau + \Delta\tau)$$



$$V_{\text{SCF}}(r; \tau) - V_{\text{SCF}}(r; \tau + \Delta\tau)$$



Perturbing potential

Energy $\Delta E = \left\langle \psi_m \left| \frac{\partial V_{\text{SCF}}}{\partial \tau} \Delta \tau \right| \psi_n \right\rangle$ Band renormalization

Wavefunction $\Delta \psi = \sum_{m \neq n} \frac{\left\langle \psi_m \left| \frac{\partial V_{\text{SCF}}}{\partial \tau} \Delta \tau \right| \psi_n \right\rangle}{E_m - E_n} \psi_m$ Optical transitions

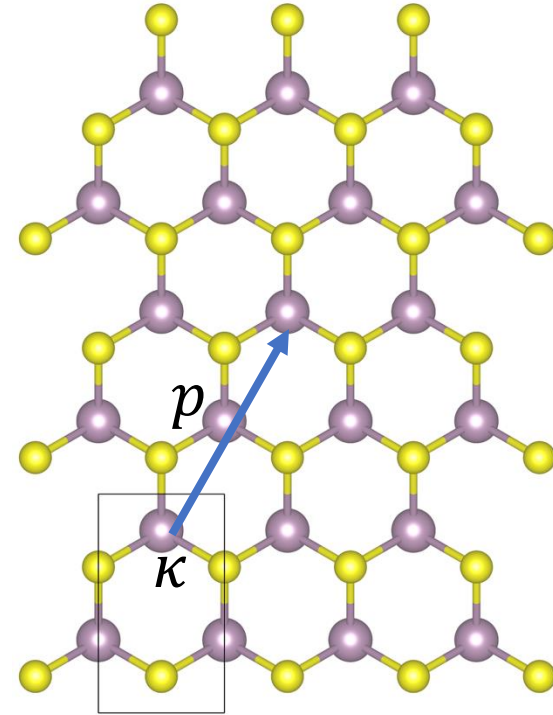
Transition Rate $\Gamma_{n \rightarrow m} = \frac{2\pi}{\hbar} \left| \left\langle \psi_m \left| \frac{\partial V_{\text{SCF}}}{\partial \tau} \Delta \tau \right| \psi_n \right\rangle \right|^2 \delta(E_m - E_n)$ Transport

- Calculation of perturbation potential is key to calculating many properties

$$\left\langle \psi_m \left| \frac{\partial V_{\text{SCF}}}{\partial \tau_{\kappa\alpha p}} \right| \psi_n \right\rangle \rightarrow \left\langle u_{m\mathbf{k}+\mathbf{q}} \left| \Delta_{\mathbf{q}\nu} v_{\text{SCF}} \right| u_{n\mathbf{k}} \right\rangle_{\text{uc}} = g_{mn}^{\nu}(\mathbf{k}, \mathbf{q})$$

↓ Supercell
↓ Unit-cell

$$\Delta_{\mathbf{q}\nu} v_{\text{SCF}} = \sum_{\kappa\alpha p} \underbrace{e^{-i\mathbf{q}\cdot(\mathbf{r}-\mathbf{R}_p)}}_{\text{Incommensurate modulation}} \sqrt{\underbrace{\frac{\hbar}{2M_{\kappa}\omega_{\mathbf{q}\nu}}}}_{\text{Zero-point amplitude}} \underbrace{e_{\kappa\alpha,\nu}(\mathbf{q})}_{\text{Phonon polarization}} \underbrace{\frac{\partial V_{\text{SCF}}(\mathbf{r})}{\partial \tau_{\kappa\alpha p}}}_{\text{Potential variation from ion displacement}}$$

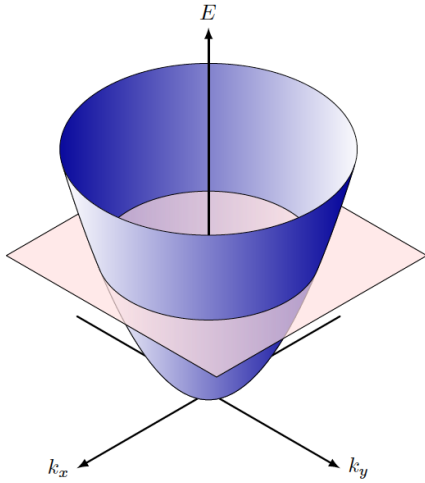


- Go from supercell to unit-cell

Giustino et.al., Rev. Mod. Phys. 89, 015013 (2017)

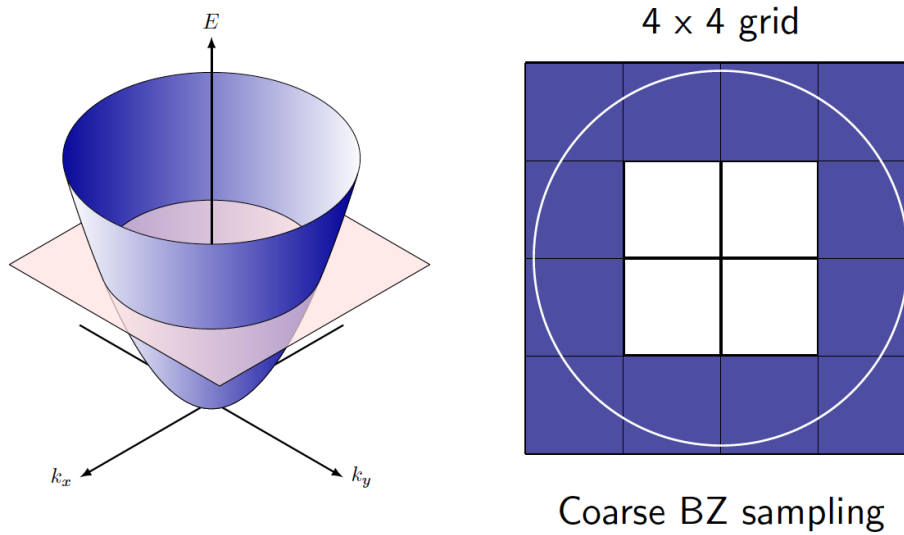
$$g_{mn}^{\nu}(\mathbf{k}, \mathbf{q}) = \langle u_{m\mathbf{k}+\mathbf{q}} | \Delta_{\mathbf{q}\nu} v_{\text{SCF}} | u_{n\mathbf{k}} \rangle$$

$$g_{mn}^{\nu}(\mathbf{k}, \mathbf{q}) = \langle u_{m\mathbf{k}+\mathbf{q}} | \Delta_{\mathbf{q}\nu} v_{\text{SCF}} | u_{n\mathbf{k}} \rangle$$



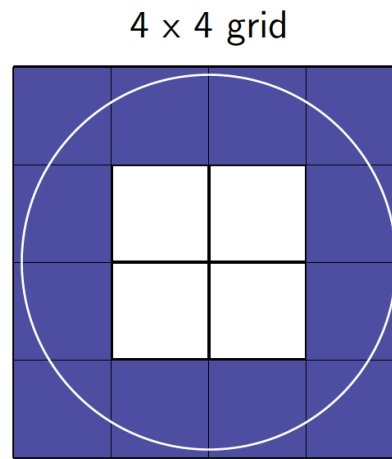
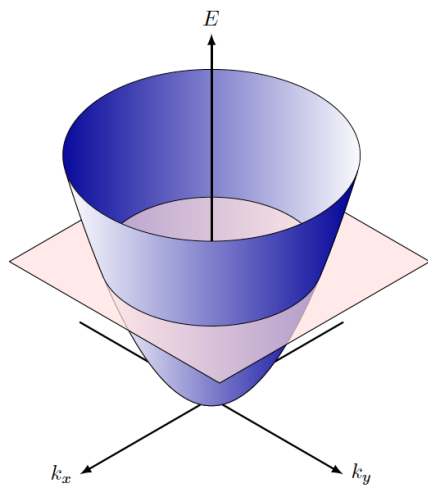
Giustino et.al., Rev. Mod. Phys. 89, 015013 (2017)

$$g_{mn}^v(\mathbf{k}, \mathbf{q}) = \langle u_{m\mathbf{k}+\mathbf{q}} | \Delta_{\mathbf{q}v} v_{\text{SCF}} | u_{n\mathbf{k}} \rangle$$

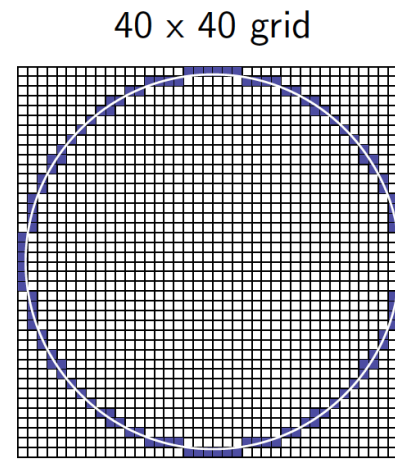


Giustino et.al., Rev. Mod. Phys. 89, 015013 (2017)

$$g_{mn}^v(\mathbf{k}, \mathbf{q}) = \langle u_{m\mathbf{k}+\mathbf{q}} | \Delta_{\mathbf{q}v} v_{\text{SCF}} | u_{n\mathbf{k}} \rangle$$



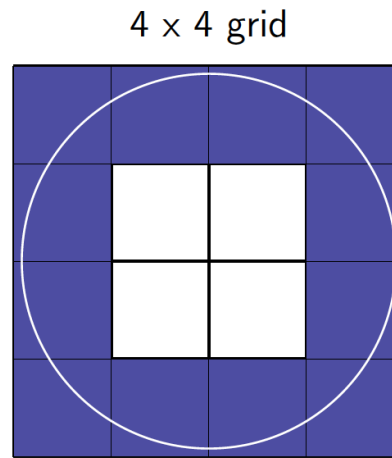
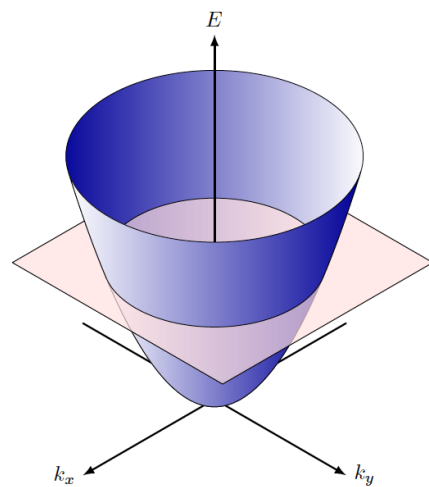
Coarse BZ sampling



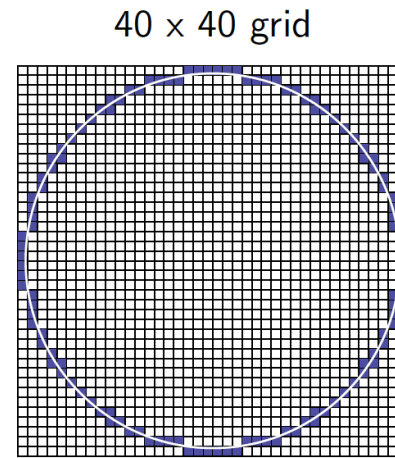
Fine BZ sampling

Giustino et.al., Rev. Mod. Phys. 89, 015013 (2017)

$$g_{mn}^v(\mathbf{k}, \mathbf{q}) = \langle u_{m\mathbf{k}+\mathbf{q}} | \Delta_{\mathbf{q}v} v_{\text{SCF}} | u_{n\mathbf{k}} \rangle$$



Coarse BZ sampling

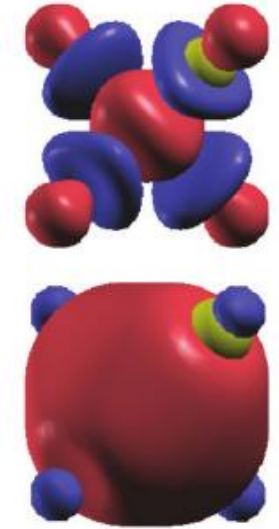


Fine BZ sampling

- Dense $\mathbf{k}, \mathbf{q}$ grids needed for calculating observables
- Wavefunctions: NSCF calculations (DFT)
 - Cheap for unit-cells
- Displacements/derivatives V_{SCF} : Phonon calculations
 - Expensive
 - Dense grids not possible

Giustino et.al., Rev. Mod. Phys. 89, 015013 (2017)

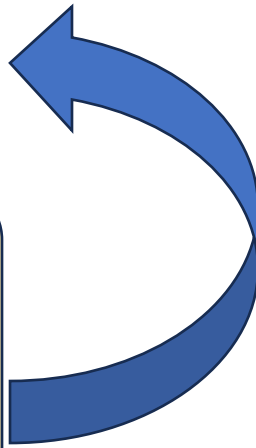
$$w_m(r; \mathbf{R}_p) = \frac{1}{N_p} \sum_{n\mathbf{k}} \exp(i\mathbf{k} \cdot \mathbf{R}_p) U_{nm}^{\mathbf{k}} |\psi_{n\mathbf{k}}\rangle$$
$$|\psi_{n\mathbf{k}}\rangle = \sum_{m\mathbf{R}_p} \exp(-i\mathbf{k} \cdot \mathbf{R}_p) U_{mn}^{\mathbf{k}*} w_m(r; \mathbf{R}_p)$$



- Wavefunctions can be localized using Wannier representation
- Interpolated to arbitrary wavevectors using Fourier transform
 - $U_{mn}^{\mathbf{k}}$: Obtained by diagonalizing Wannier Hamiltonian

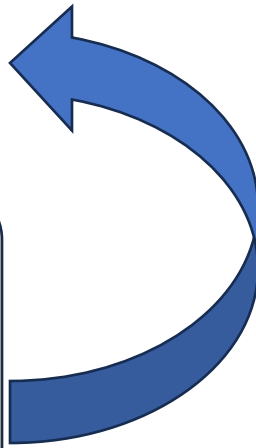
Marzari et.al., Rev. Mod. Phys. 84, 1419 (2012)

$$g_{mn}^{\nu}(\mathbf{k}, \mathbf{q}) = \langle u_{m\mathbf{k}+\mathbf{q}} | \Delta_{\mathbf{q}\nu} v_{\text{SCF}} | u_{n\mathbf{k}} \rangle$$

$$w_m(r; \mathbf{R}_p) = \frac{1}{N_p} \sum_{n\mathbf{k}} \exp(i\mathbf{k} \cdot \mathbf{R}_p) U_{nm}^{\mathbf{k}} |\psi_{n\mathbf{k}}\rangle$$
$$|\psi_{n\mathbf{k}}\rangle = \sum_{m\mathbf{R}_p} \exp(-i\mathbf{k} \cdot \mathbf{R}_p) U_{mn}^{\mathbf{k}*} w_m(r; \mathbf{R}_p)$$


$$g_{mn}^v(\mathbf{k}, \mathbf{q}) = \langle u_{m\mathbf{k}+\mathbf{q}} | \Delta_{\mathbf{q}v} v_{\text{SCF}} | u_{n\mathbf{k}} \rangle$$

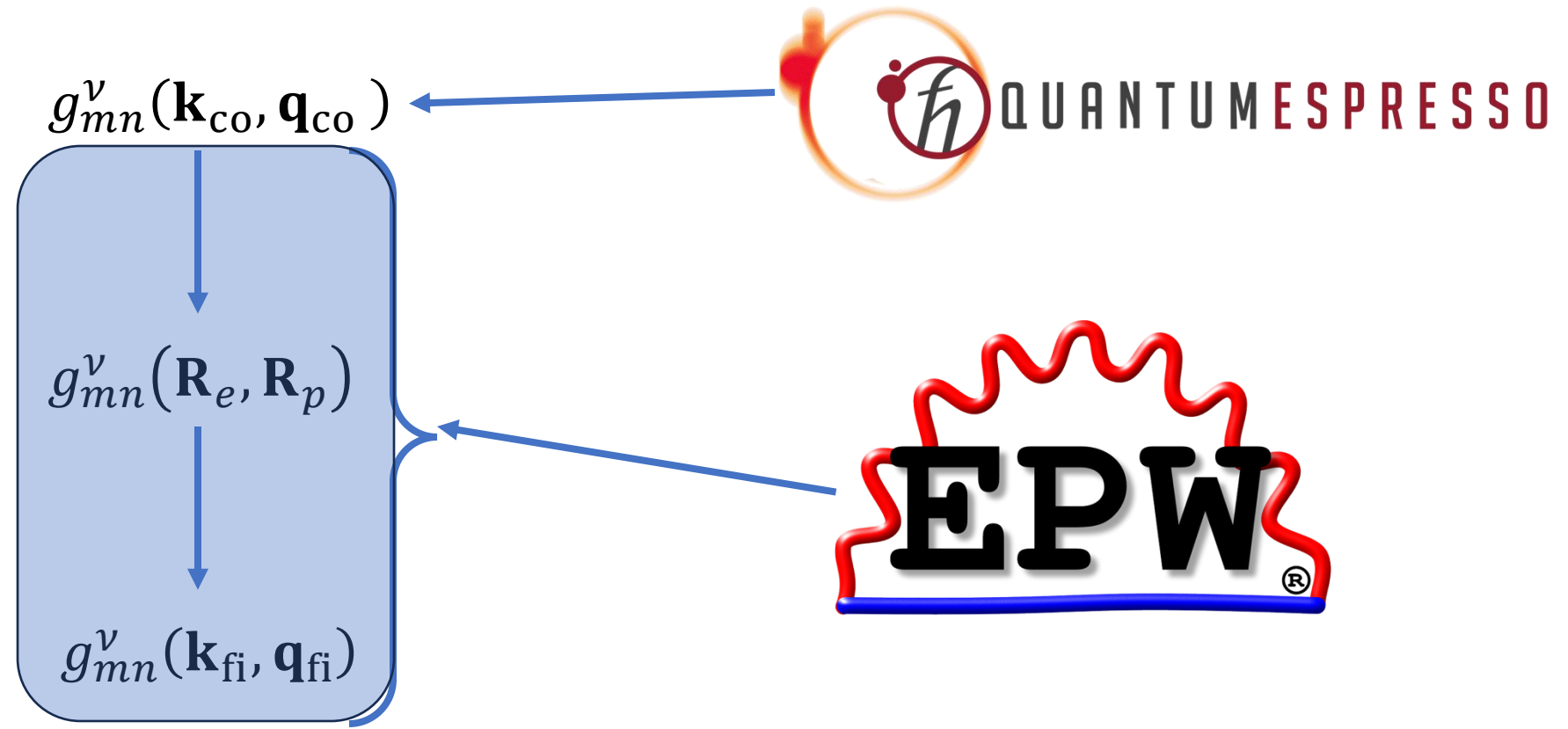
$$w_m(r; \mathbf{R}_p) = \frac{1}{N_p} \sum_{n\mathbf{k}} \exp(i\mathbf{k} \cdot \mathbf{R}_p) U_{nm}^{\mathbf{k}} |\psi_{n\mathbf{k}}\rangle$$

$$|\psi_{n\mathbf{k}}\rangle = \sum_{m\mathbf{R}_p} \exp(-i\mathbf{k} \cdot \mathbf{R}_p) U_{mn}^{\mathbf{k}*} w_m(r; \mathbf{R}_p)$$


$$g_{mn}^v(\mathbf{k}_{\text{co}}, \mathbf{q}_{\text{co}}) \rightarrow g_{mn}^v(\mathbf{R}_e, \mathbf{R}_p) \rightarrow g_{mn}^v(\mathbf{k}_{\text{fi}}, \mathbf{q}_{\text{fi}})$$

- A method for reliably interpolating e-p matrix element

Giustino et.al., Phys. Rev. B. 76, 165108 (2007)



- QE: $\Delta_{\mathbf{q}v} v_{\text{SCF}}; \Delta\tau$
- EPW: Wannierization and interpolation

Ponce et.al., Comput. Phys. Comm, 209, 116-133 (2016)

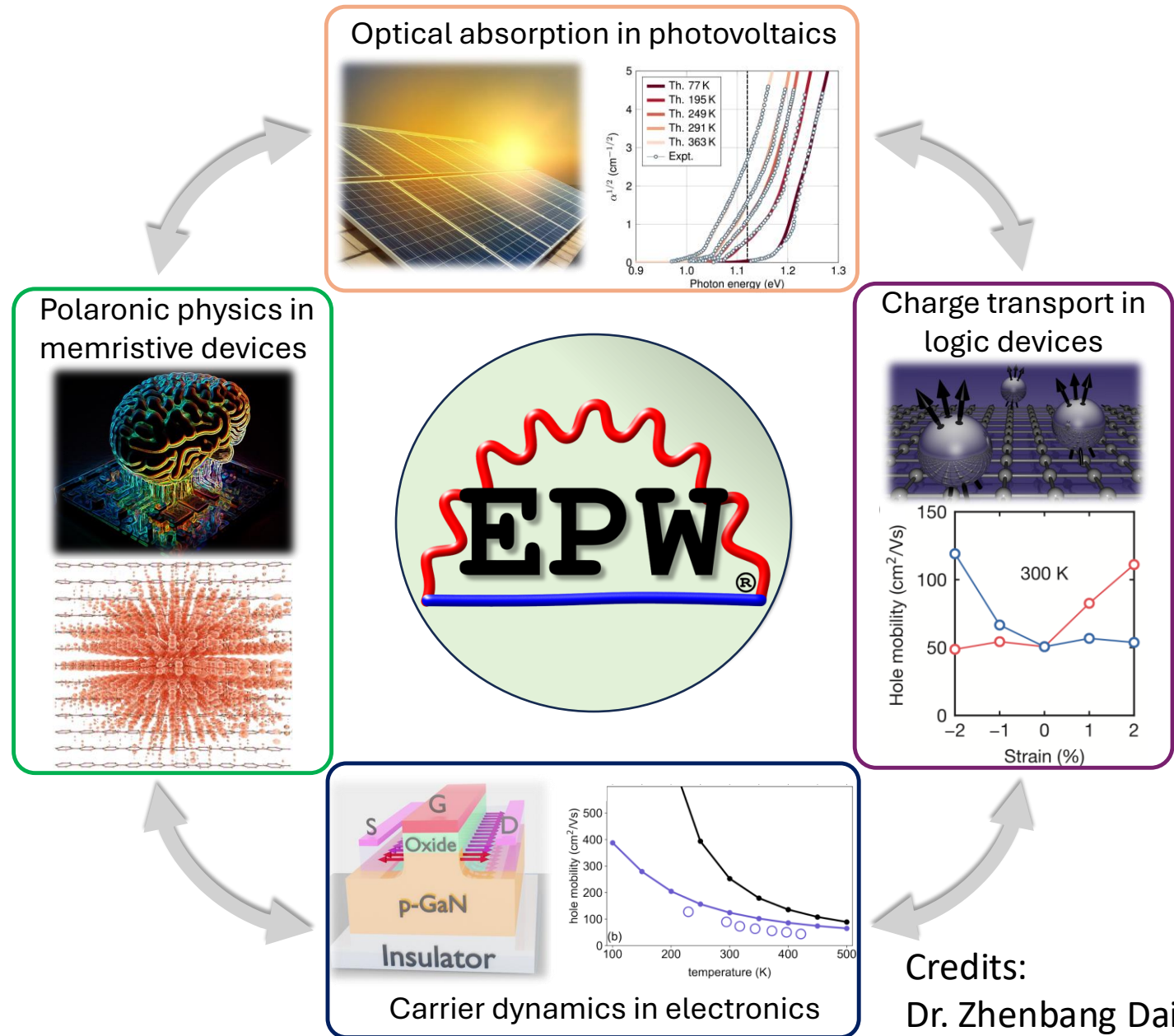
Lee, ... Tiwari et.al., npj comput. mater., 9, 156 (2023)

- Electron-Phonon Wannier (EPW) code
 - A sub-package within the Quantum Espresso DFT package
 - Interpolates e-p matrix elements obtained from DFPT (QE)
 - Allows calculations on very fine $\mathbf{k}$ and $\mathbf{q}$ grids
 - **Transport (IBTE)**
 - **Polarons**
 - Superconductivity
 - Temperature dependent Bandgap
 - Optical absorption



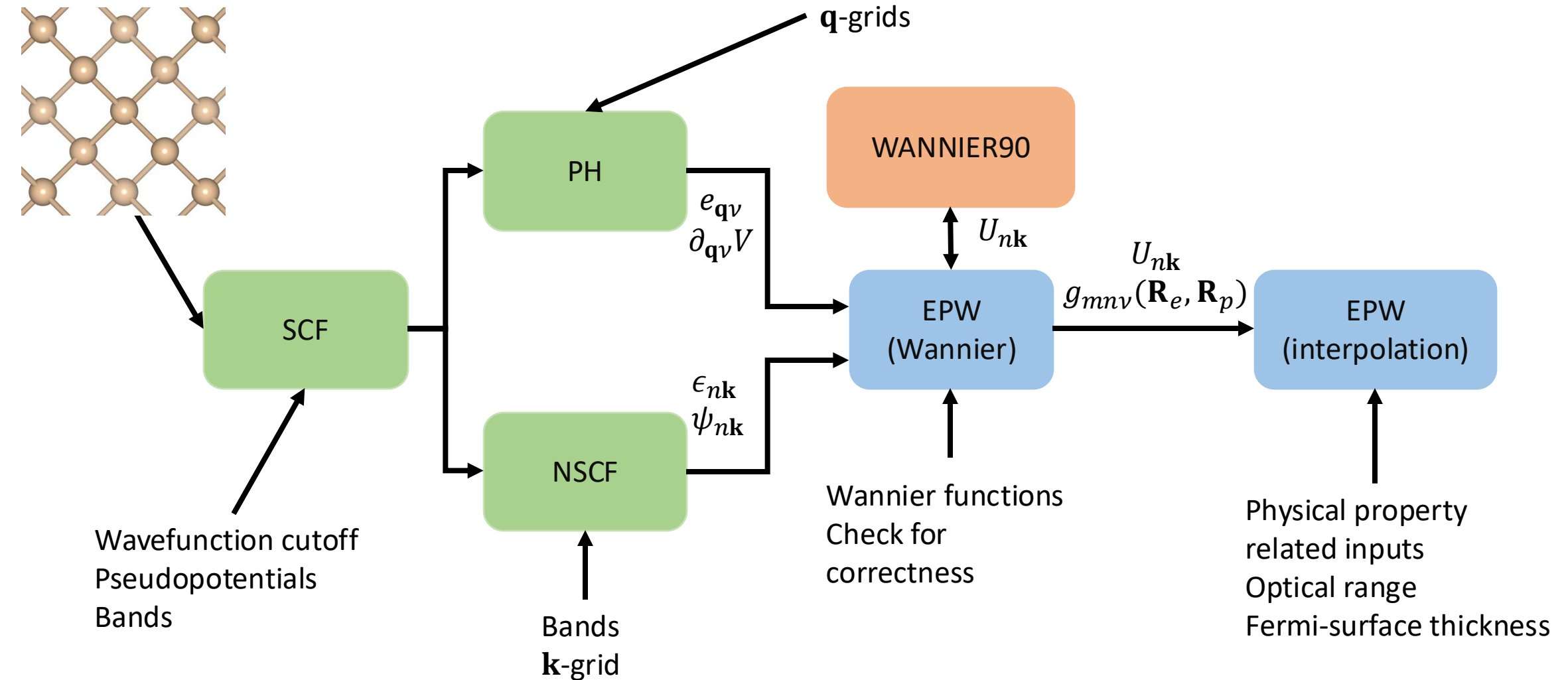
Current computational materials design landscape (EPW)

- Computational materials design impacts all walks of life today
- These calculations require very complex workflows and **user intervention**
- High-throughput study of large materials databases is **currently out of reach**
- Challenge is two-fold
 - Streamlining calculations
 - Analysis

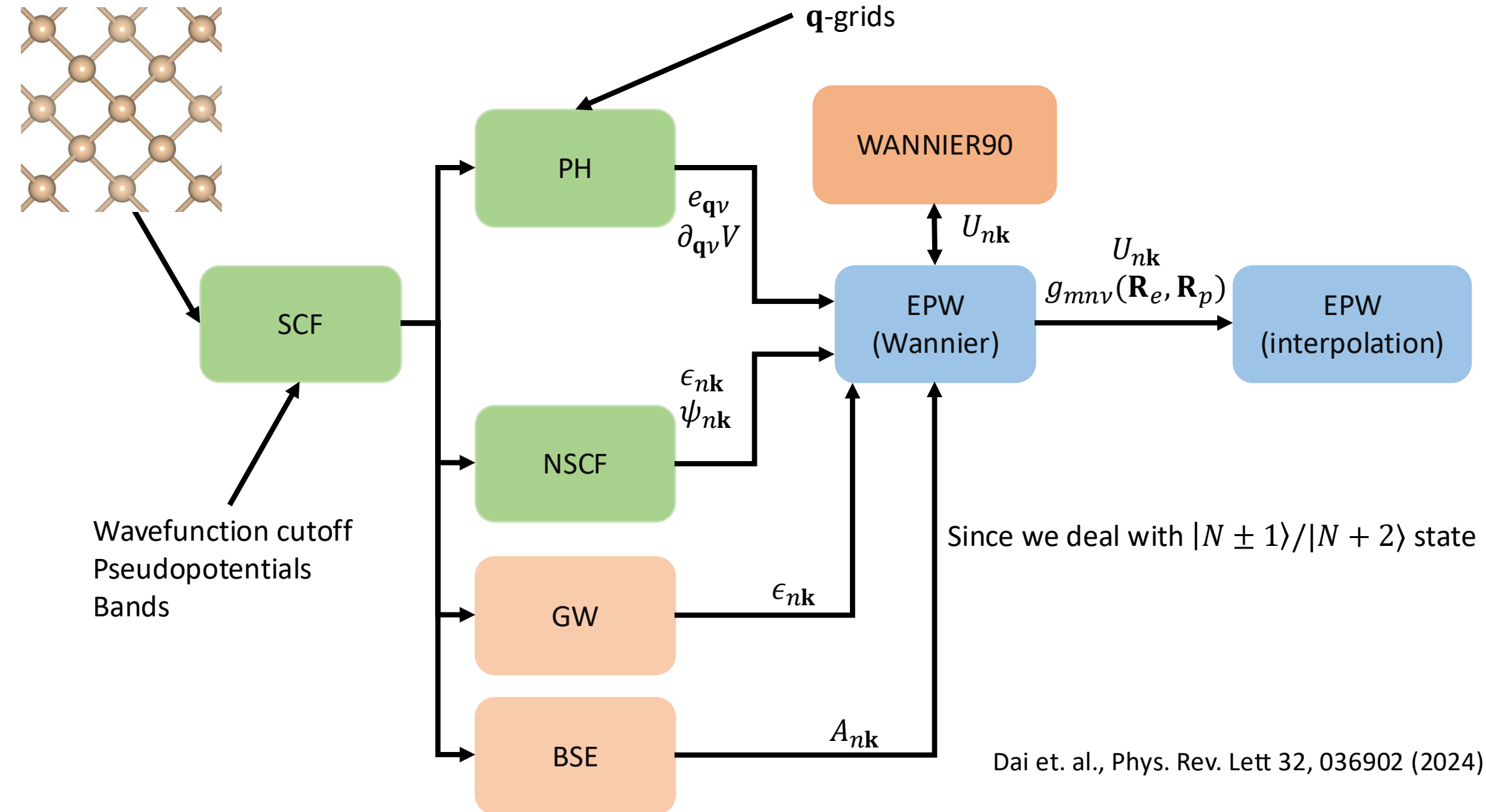


Credits:
Dr. Zhenbang Dai

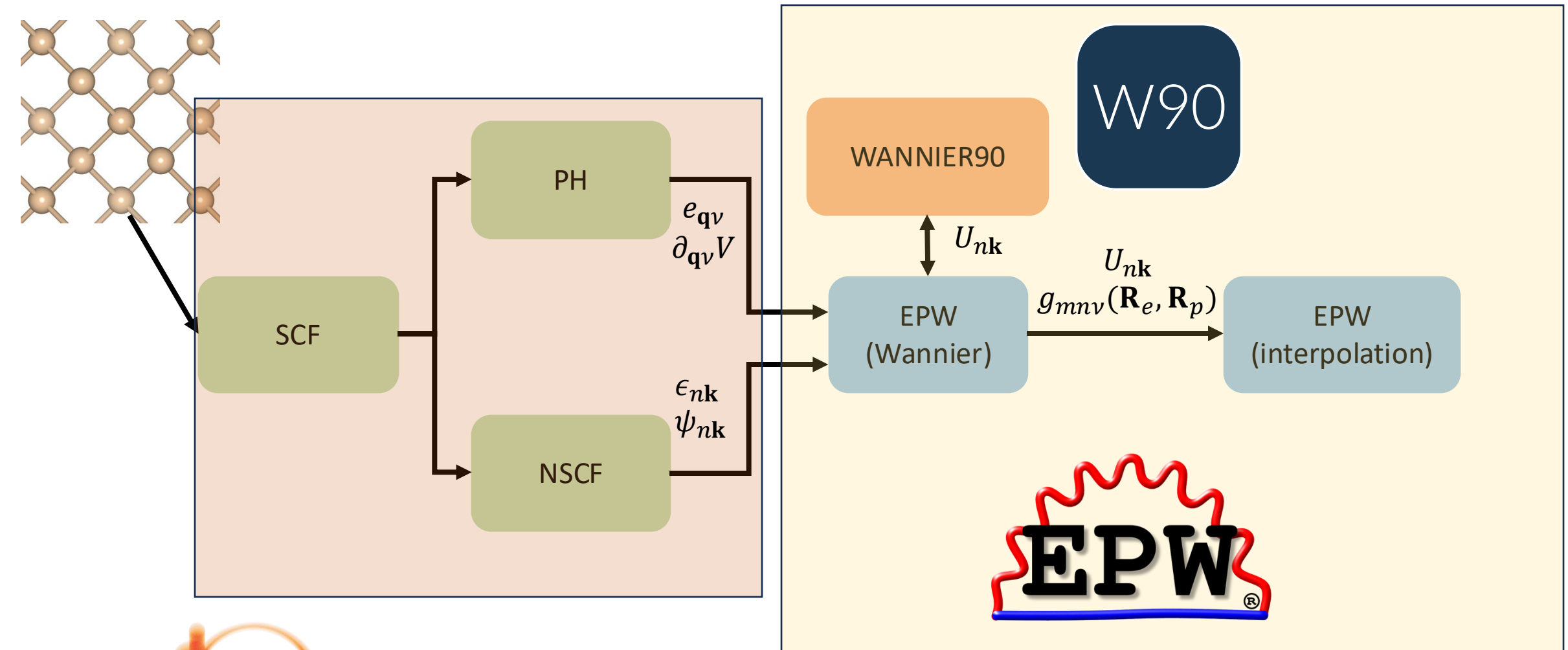
A typical workflow for EPW



A typical workflow for EPW (more complex)



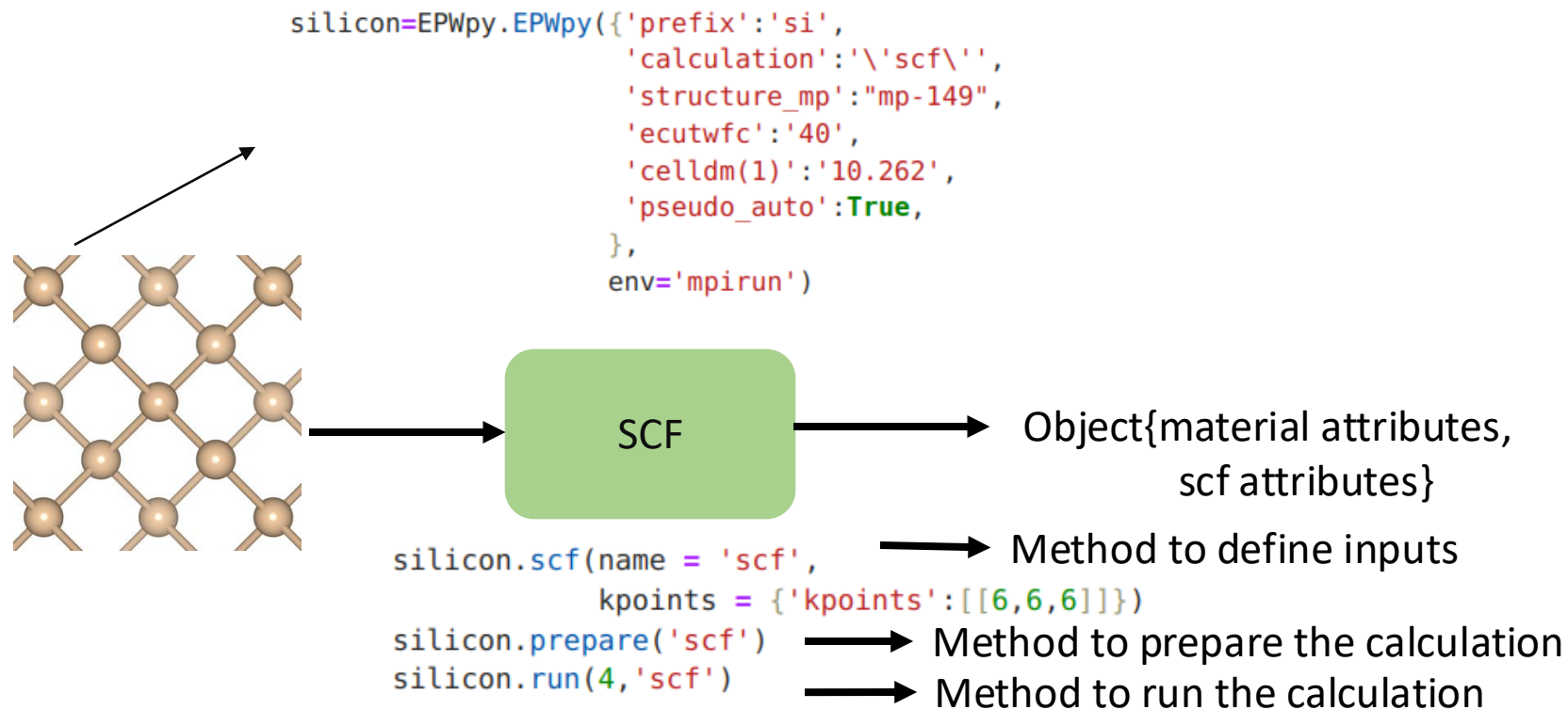
A typical workflow for EPW



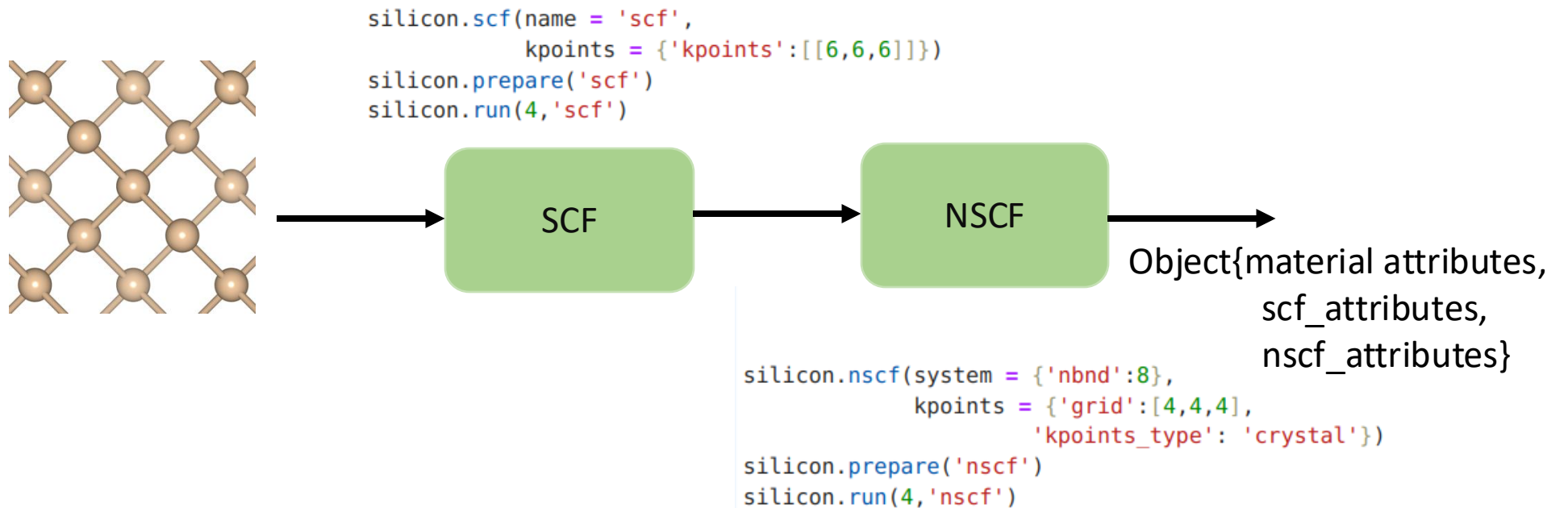
- Multiple codes communicating with each-other
- Data transfer and user intervention
- There is a need for a unified method to streamline calculations

- Electron-Phonon-Physics with EPW
- Programmable abstraction using EPWpy
- Integrated MATCSSI portal
- Conclusion

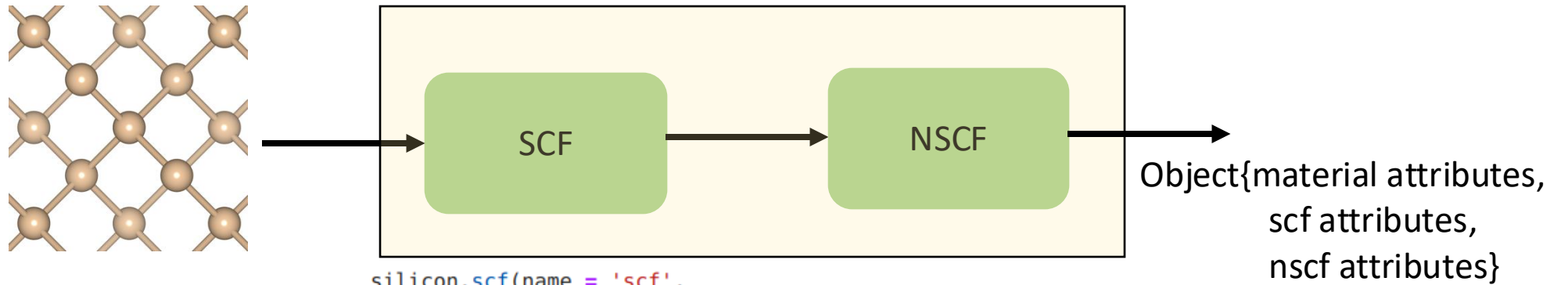
Functional abstraction



- Material: **Object**
- Calculations: **Methods**
- Each calculation can be seen as a **functional block**
- These blocks **can take user input** else **automated**



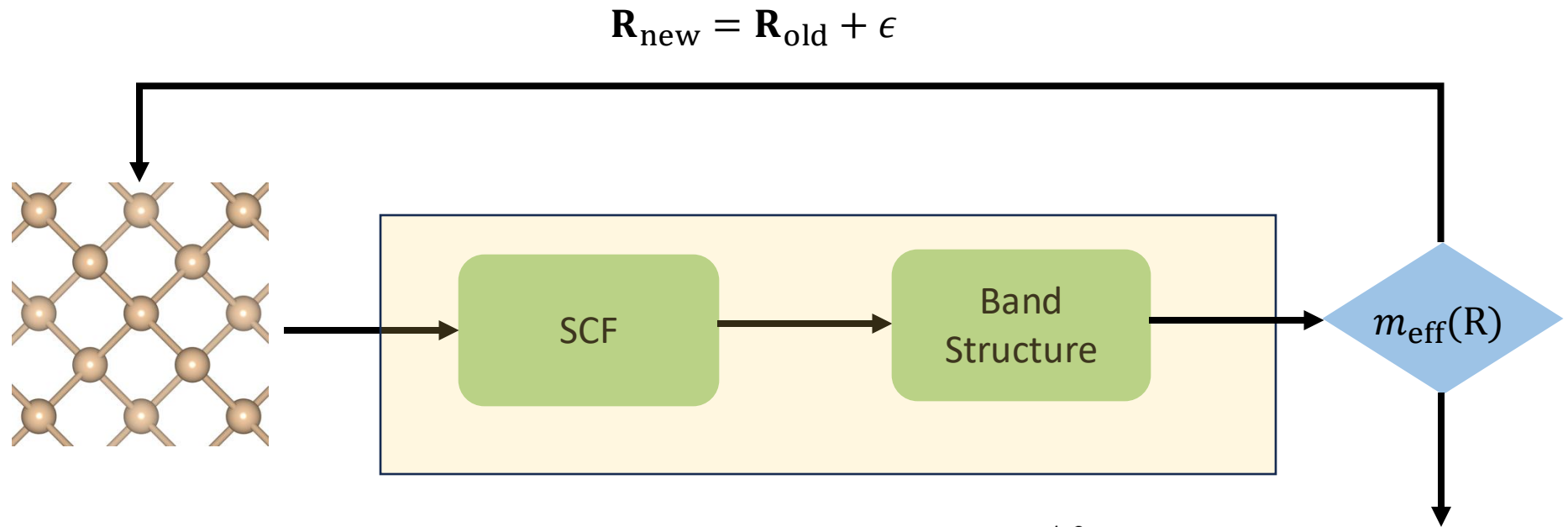
- Block to block connection **through position**
- Data transfer is handled accordingly



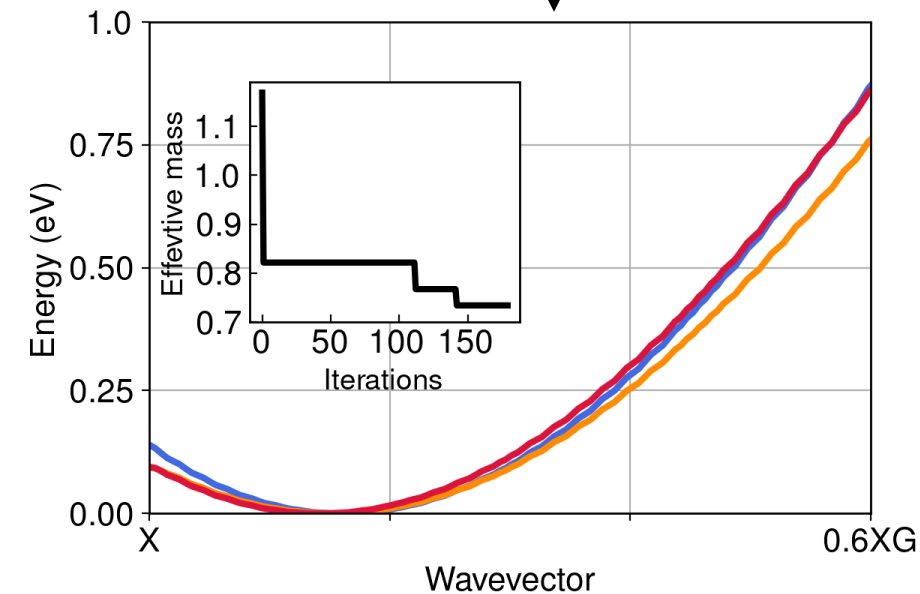
```
silicon.scf(name = 'scf',  
            kpoints = {'kpoints': [[6,6,6]]})  
silicon.prepare('scf')  
silicon.run(4, 'scf')  
  
silicon.nscf(system = {'nbnd': 8},  
             kpoints = {'grid': [4,4,4],  
                       'kpoints_type': 'crystal'})  
silicon.prepare('nscf')  
silicon.run(4, 'nscf')
```

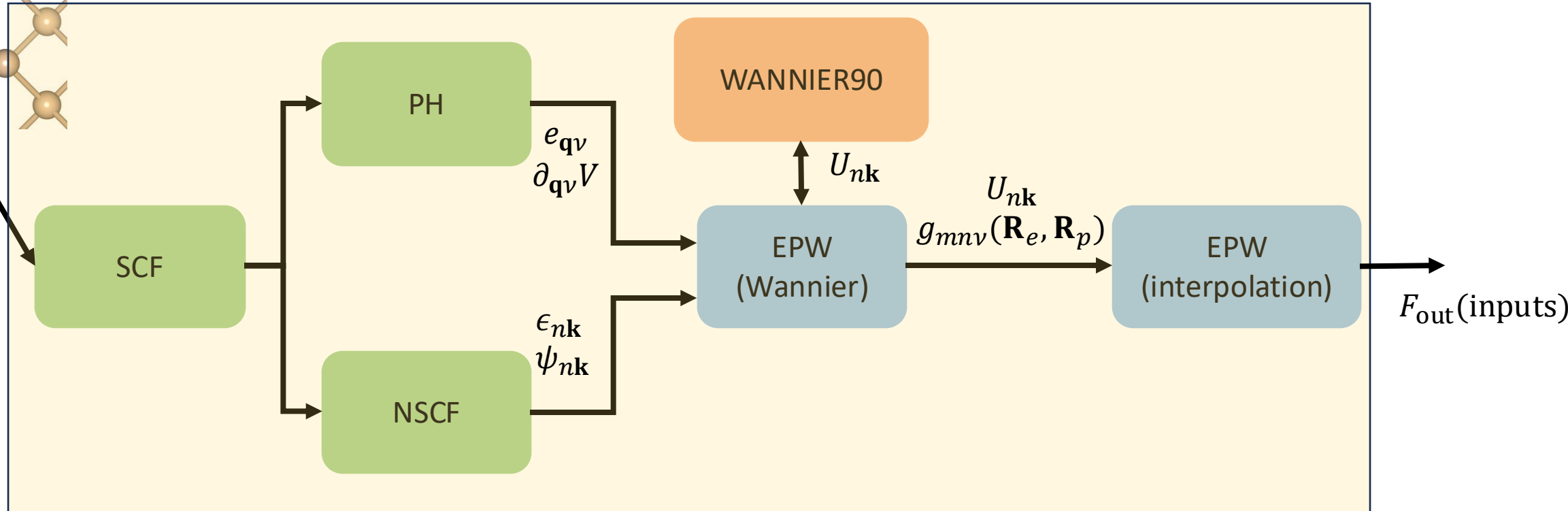
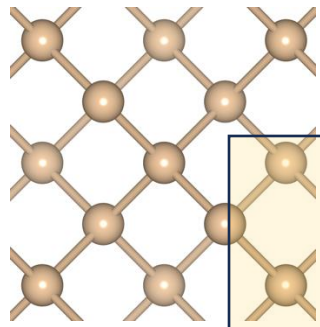
- Blocks combined to give single abstract function

Optimization of these abstract functions

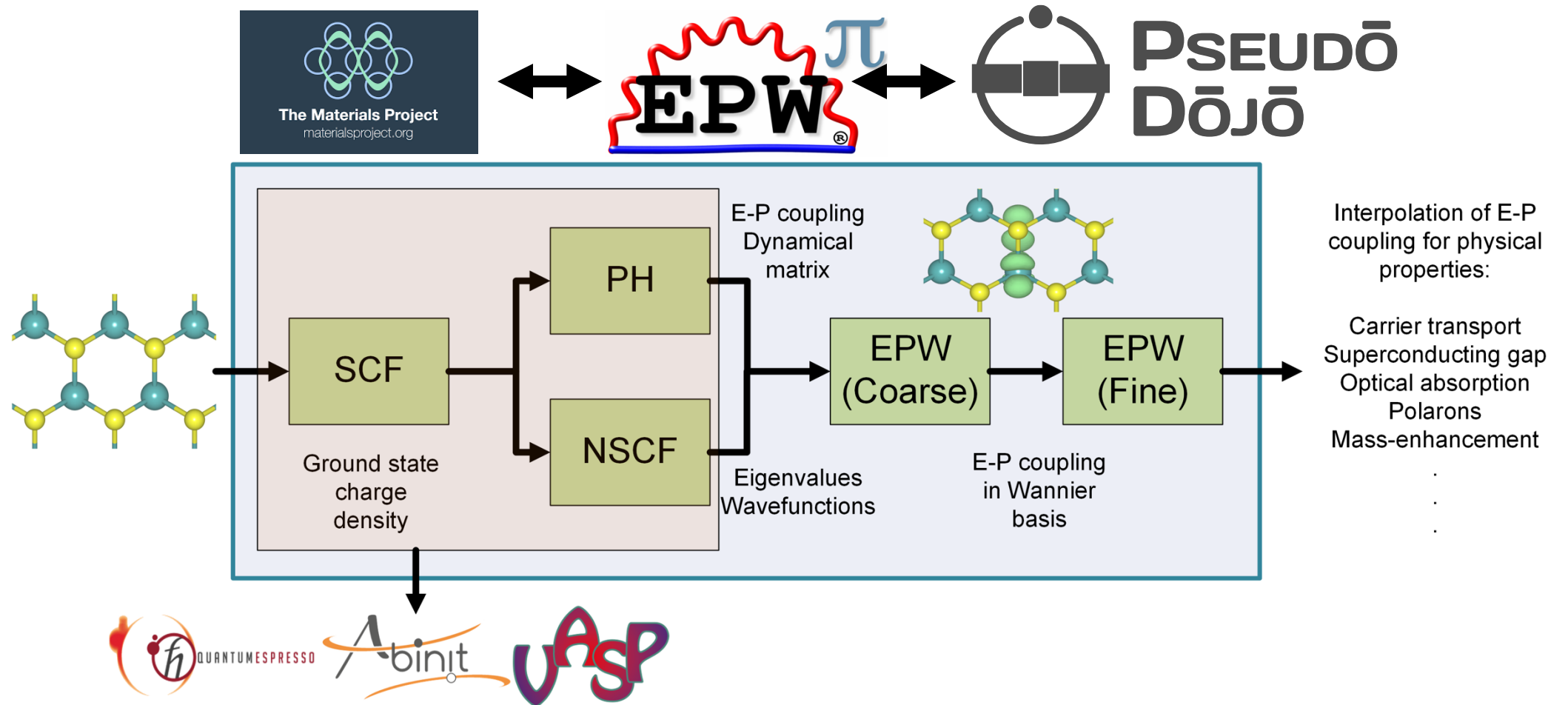


- Entire physical calculation can be encapsulated
- These abstract functions can be optimized using top level optimizers



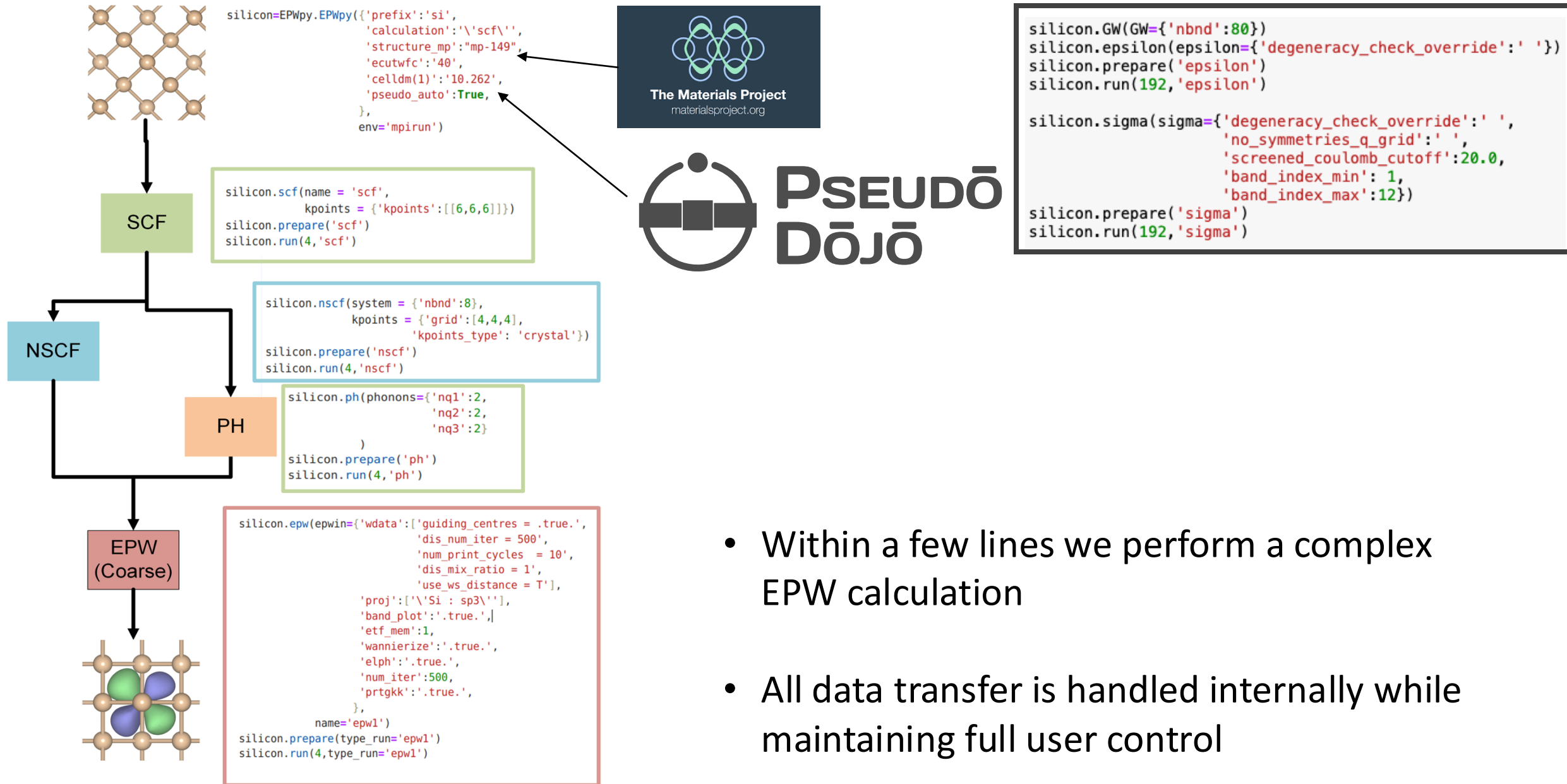


- Entire physical calculation can be encapsulated



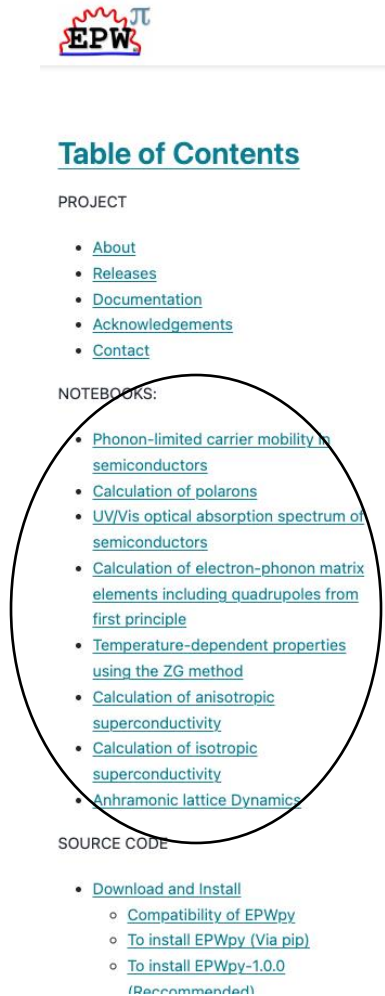
- EPWpy is a python program allowing such object-oriented abstractions
- Currently, interface with BGW, partial interface with VASP and Abinit
- Comes with interface with Materials project and PseudoDojo

A typical EPWpy script



- Within a few lines we perform a complex EPW calculation
- All data transfer is handled internally while maintaining full user control

- EPWpy has a dedicated website
- Contains tutorials
- Notebooks can be run on TACC infrastructure



The screenshot shows the EPWpy website with a navigation bar at the top containing links for About, Releases, Documentation, Acknowledgements, Contact, and More. A search bar is located on the right. The main content area is divided into sections: PROJECT (with links to About, Releases, Documentation, Acknowledgements, and Contact), NOTEBOOKS (with a list of 10 notebooks, including 'Phonon-limited carrier mobility in semiconductors' and 'Calculation of electron-phonon matrix elements including quadrupoles from first principle'), and SOURCE CODE (with links to Download and Install, Compatibility of EPWpy, To install EPWpy (Via pip), and To install EPWpy-1.0.0 (Recommended)). A large circle highlights the NOTEBOOKS section.

Table of Contents

PROJECT

- [About](#)
- [Releases](#)
- [Documentation](#)
- [Acknowledgements](#)
- [Contact](#)

NOTEBOOKS:

- [Phonon-limited carrier mobility in semiconductors](#)
- [Calculation of polarons](#)
- [UV/Vis optical absorption spectrum of semiconductors](#)
- [Calculation of electron-phonon matrix elements including quadrupoles from first principle](#)
- [Temperature-dependent properties using the ZG method](#)
- [Calculation of anisotropic superconductivity](#)
- [Calculation of isotropic superconductivity](#)
- [Anharmonic lattice Dynamics](#)

SOURCE CODE

- [Download and Install](#)
 - [Compatibility of EPWpy](#)
 - [To install EPWpy \(Via pip\)](#)
 - [To install EPWpy-1.0.0 \(Recommended\)](#)

make run-tests-local

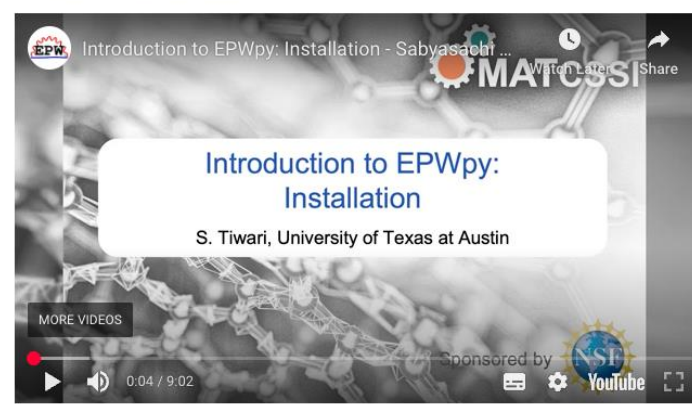
5. In calculations involving EPWpy you will need to direct EPWpy to a location where QE and EPW executables are present. However, EPWpy can itself download and build QE+EPW. To use this feature use:

```
python setup.py install --with-QE True --config <QE configuration> --cores <No. of cores>
```

6. This will install QE+EPW in a folder called build inside the EPWpy folder. You will find executables inside:

```
EPWpy/build/q-e/bin/
```

Video tutorial



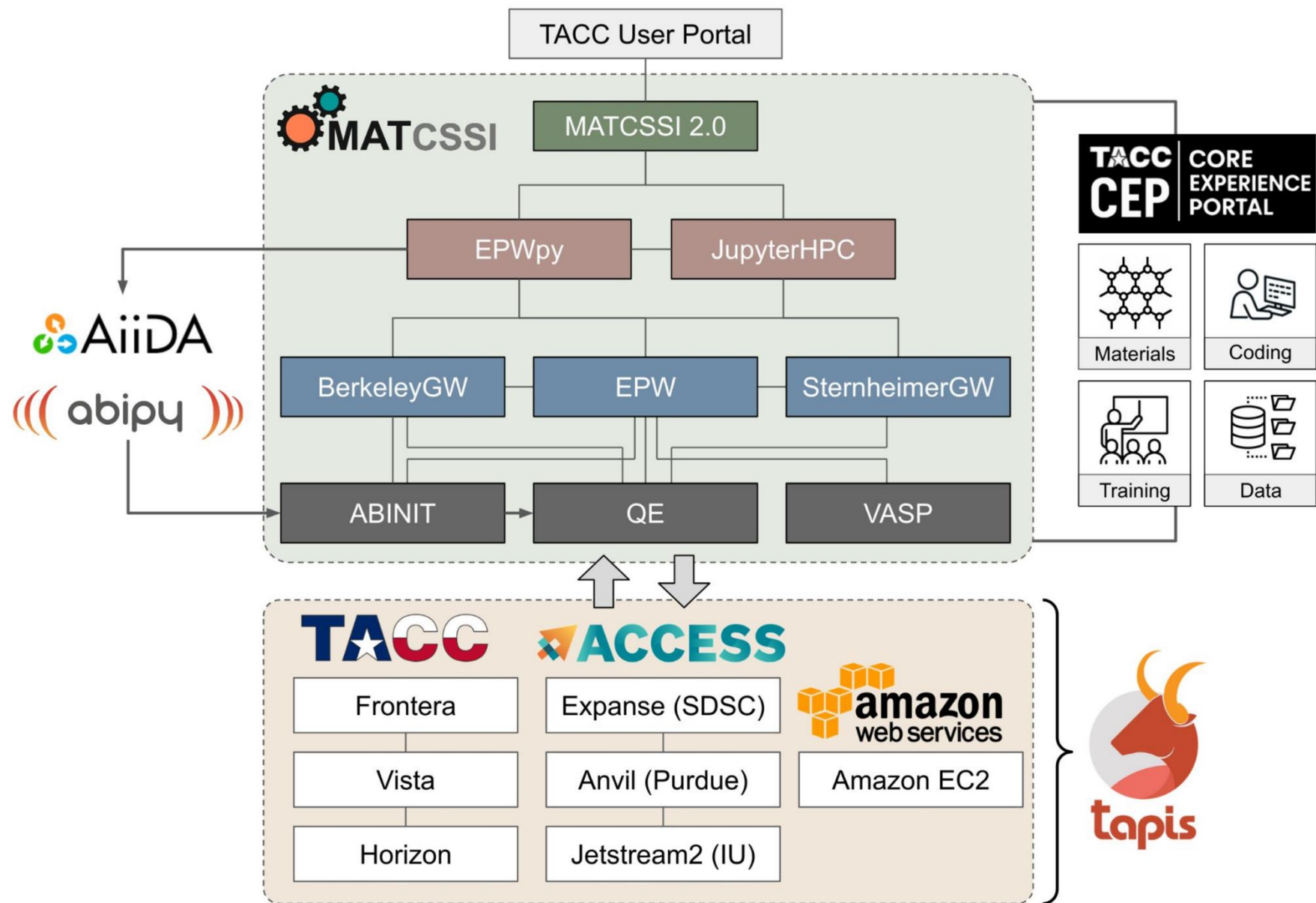
The video player shows a thumbnail for 'Introduction to EPWpy: Installation' by S. Tiwari, University of Texas at Austin. The video is sponsored by MATCSSI and has a duration of 9:02. The video player includes controls for play, volume, and progress.

This video tutorial can also be found here: <https://www.youtube.com/watch?v=q6kyn48KmgE&t=3s>

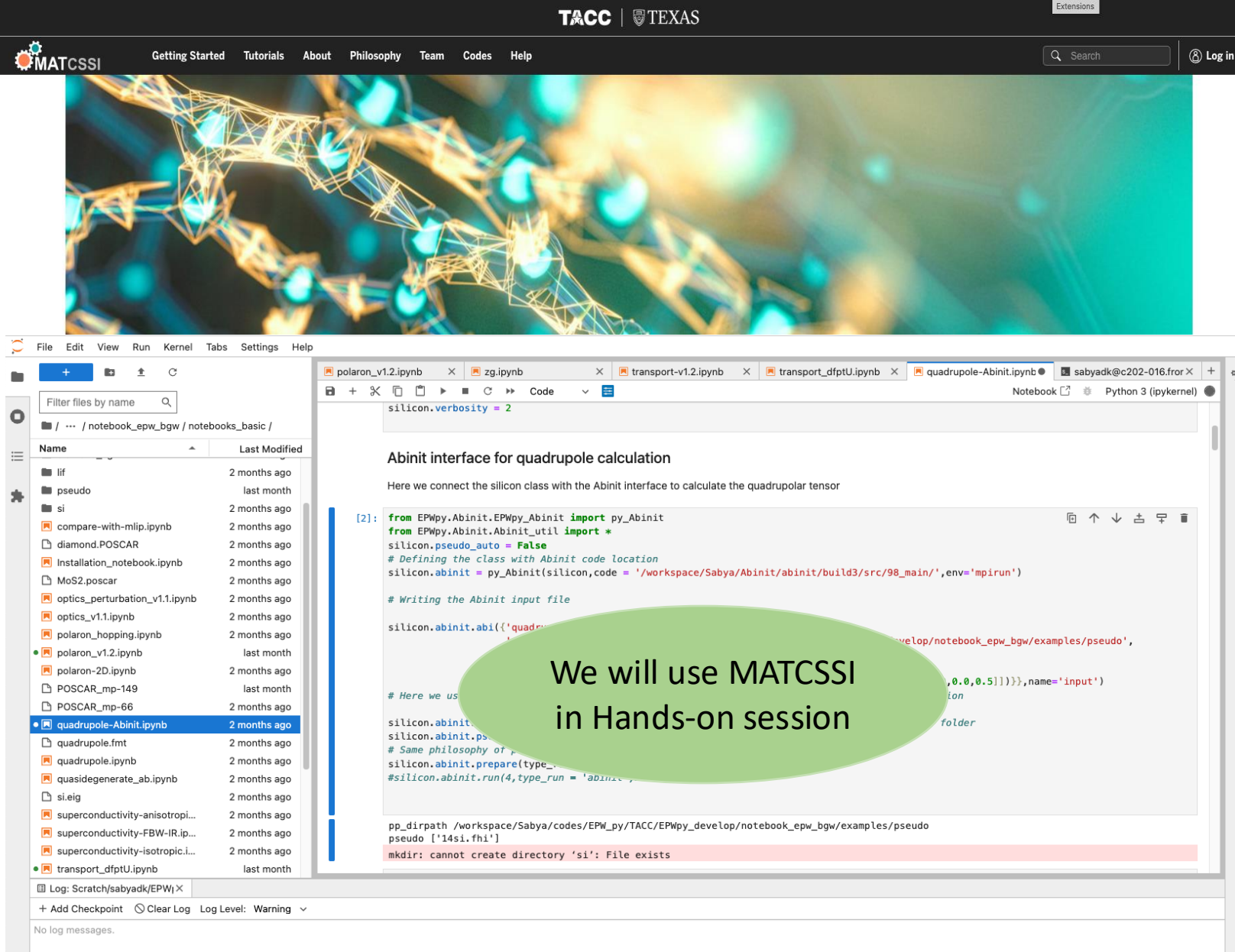
Previous [Anharmonic lattice Dynamics](#) Next [EPWpy](#)

- Electron-Phonon-Physics with EPW
- Programmable abstraction using EPWpy
- Integrated MATCSSI portal
- Conclusion

Enabling interoperable *ab initio* simulation environment



- MATCSSI is a portal powered by TACC
- Large scale calculations using EPWpy can be performed using Jupyter notebook (MPI-TACC)
- Allocations can be requested



The screenshot displays the MATCSSI portal interface. At the top, the TACC | TEXAS logo is visible. Below the header, there is a navigation bar with links: Getting Started, Tutorials, About, Philosophy, Team, Codes, and Help. A search bar and a Log in button are also present. The main content area features a large, abstract image of a molecular structure. Below this, a Jupyter notebook interface is shown. The left sidebar displays a file browser with a list of files and folders, including 'lif', 'pseudo', 'si', 'compare-with-mlip.ipynb', 'diamond.POSCAR', 'Installation_notebook.ipynb', 'MoS2.poscar', 'optics_perturbation_v1.1.ipynb', 'optics_v1.1.ipynb', 'polaron_hopping.ipynb', 'polaron_v1.2.ipynb', 'polaron-2D.ipynb', 'POSCAR_mp-149', 'POSCAR_mp-66', 'quadrupole-Abinit.ipynb' (highlighted), 'quadrupole.fmt', 'quadrupole.ipynb', 'quasidegenerate_ab.ipynb', 'si.eig', 'superconductivity-anisotropi...', 'superconductivity-FBW-IR.ip...', 'superconductivity-isotropic...', and 'transport_dfptU.ipynb'. The main area shows the 'quadrupole-Abinit.ipynb' notebook, which contains code for calculating the quadrupole tensor using the Abinit interface. A green oval with the text 'We will use MATCSSI in Hands-on session' is overlaid on the notebook content.

TACC | TEXAS

MATCSSI

Getting Started Tutorials About Philosophy Team Codes Help

Search Log in

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name Last Modified

lif 2 months ago

pseudo last month

si 2 months ago

compare-with-mlip.ipynb 2 months ago

diamond.POSCAR 2 months ago

Installation_notebook.ipynb 2 months ago

MoS2.poscar 2 months ago

optics_perturbation_v1.1.ipynb 2 months ago

optics_v1.1.ipynb 2 months ago

polaron_hopping.ipynb 2 months ago

polaron_v1.2.ipynb last month

polaron-2D.ipynb 2 months ago

POSCAR_mp-149 last month

POSCAR_mp-66 2 months ago

quadrupole-Abinit.ipynb 2 months ago

quadrupole.fmt 2 months ago

quadrupole.ipynb 2 months ago

quasidegenerate_ab.ipynb 2 months ago

si.eig 2 months ago

superconductivity-anisotropi... 2 months ago

superconductivity-FBW-IR.ip... 2 months ago

superconductivity-isotropic... 2 months ago

transport_dfptU.ipynb last month

Log: Scratch/sabyadk/EPWj X

+ Add Checkpoint Clear Log Log Level: Warning

No log messages.

quadrupole-Abinit.ipynb

silicon.verbosity = 2

Abinit interface for quadrupole calculation

Here we connect the silicon class with the Abinit interface to calculate the quadrupolar tensor

```
[2]: from EPWpy.Abinit.EPWpy_Abinit import py_Abinit
from EPWpy.Abinit.Abinit_util import *
silicon.pseudo_auto = False
# Defining the class with Abinit code location
silicon.abinit = py_Abinit(silicon, code = '/workspace/Sabya/Abinit/abinit/build3/src/98_main/', env='mpirun')

# Writing the Abinit input file

silicon.abinit.abi({'quadrupole', 'pseudopotential', 'develop/notebook_epw_bgw/examples/pseudo',
                    '0.0,0.5']]), name='input')

# Here we use the Abinit interface to calculate the quadrupolar tensor
silicon.abinit.pseudo = silicon.abinit.pseudo
# Same philosophy of the Abinit interface
silicon.abinit.prepare(type='quadrupole', pseudo='pseudo', folder='pseudo')
#silicon.abinit.run(4, type_run = 'abinit',

pp_dirpath /workspace/Sabya/codes/EPW_py/TACC/EPWpy_develop/notebook_epw_bgw/examples/pseudo
pseudo ['14si.fhi']
mkdir: cannot create directory 'si': File exists
```



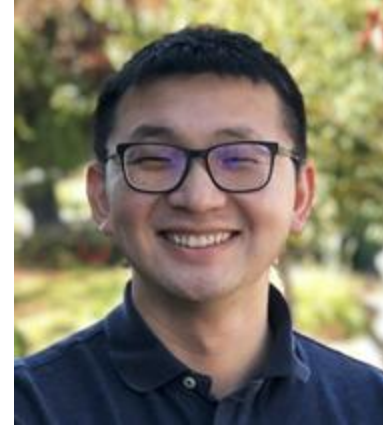
Feliciano Giustino
UT Austin



Dan Stanzione
TACC



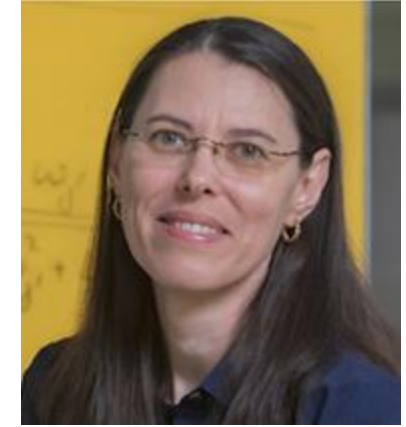
Maytal Dahan
TACC



Zhenglu Li
USC



Steven Louie
UC Berkeley



Roxana Margine
Binghamton U

This initiative is supported by the [Office of Advanced Cyberinfrastructure](#) and the [Condensed Matter and Materials Theory](#) division of the [National Science Foundation](#) under the CSSI Frameworks program, [Award #2513830](#)



- Electron-Phonon-Physics with EPW
- Programmable abstraction using EPWpy
- Integrated MATCSSI portal
- Conclusion

- Electron-Phonon physics is important for many physical properties
- EPW and EPWpy provide a convenient tool for exploring electron-phonon physics
- EPW GitLab: <https://gitlab.com/epw/q-e>
- EPWpy GitLab: <https://gitlab.com/epwpy/epwpy>

Thank You

Questions?