DFT and DFT embedding

Ways to approach the mesoscopic world with ab-initio methods (without cutting corners)

Michele Pavanello

Rutgers University-Newark

http://prg.rutgers.edu

ASESMA School -- June 22, 2023

Ready to work?

Find this Jupyter Notebook at
~/ASESMA2023/Special_Topics_Seminars/Michele_Pavanello

Acknowledgements

Students

Dr Xuecheng Shao

Dr Xin Chen

Alina Umerbekova

Jessica Martinez

Valeria Rios

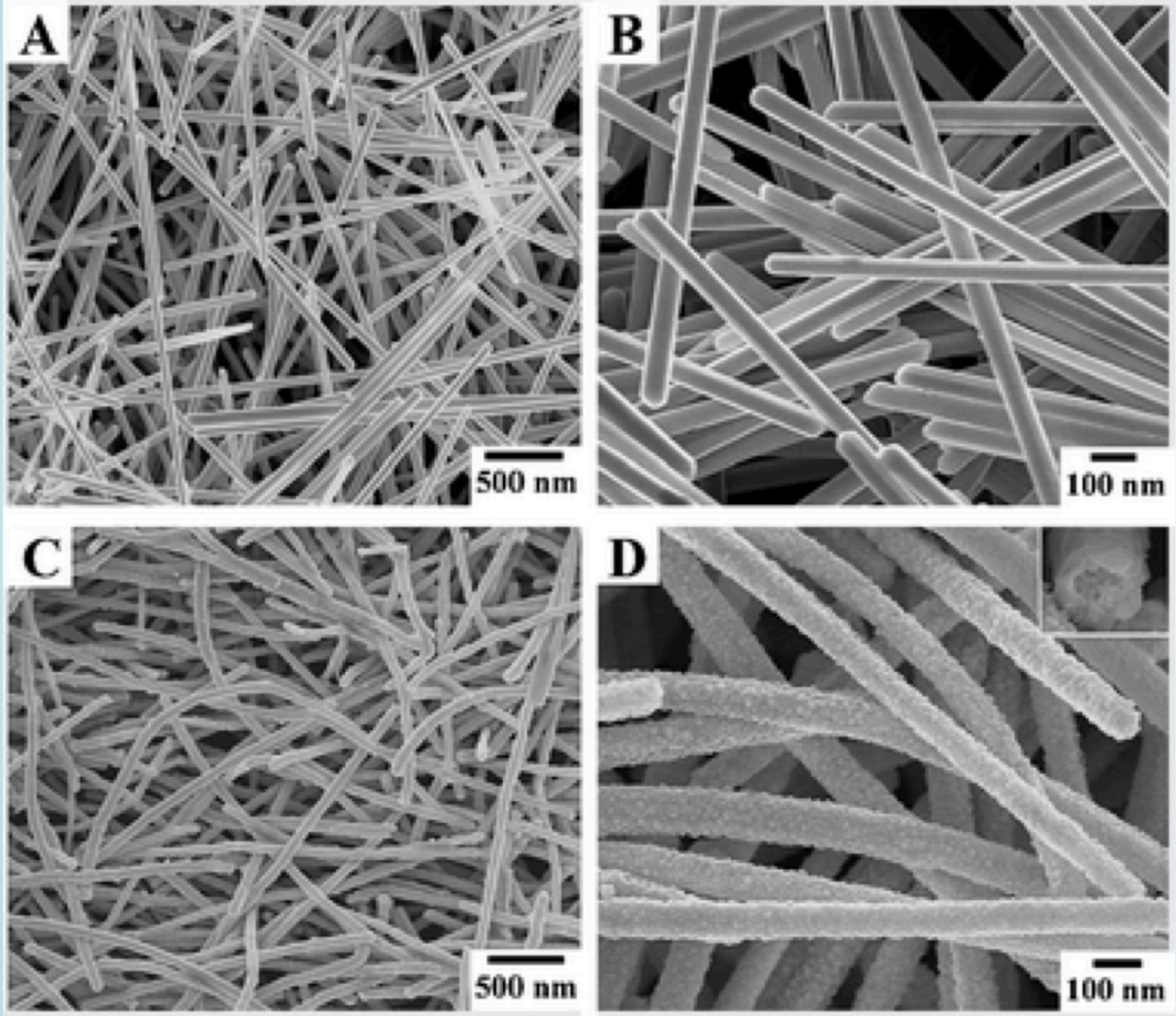
Nick Viot

December 2022

Funding: NSF, DOE, MolSSI, ACS-PRF, Rutgers

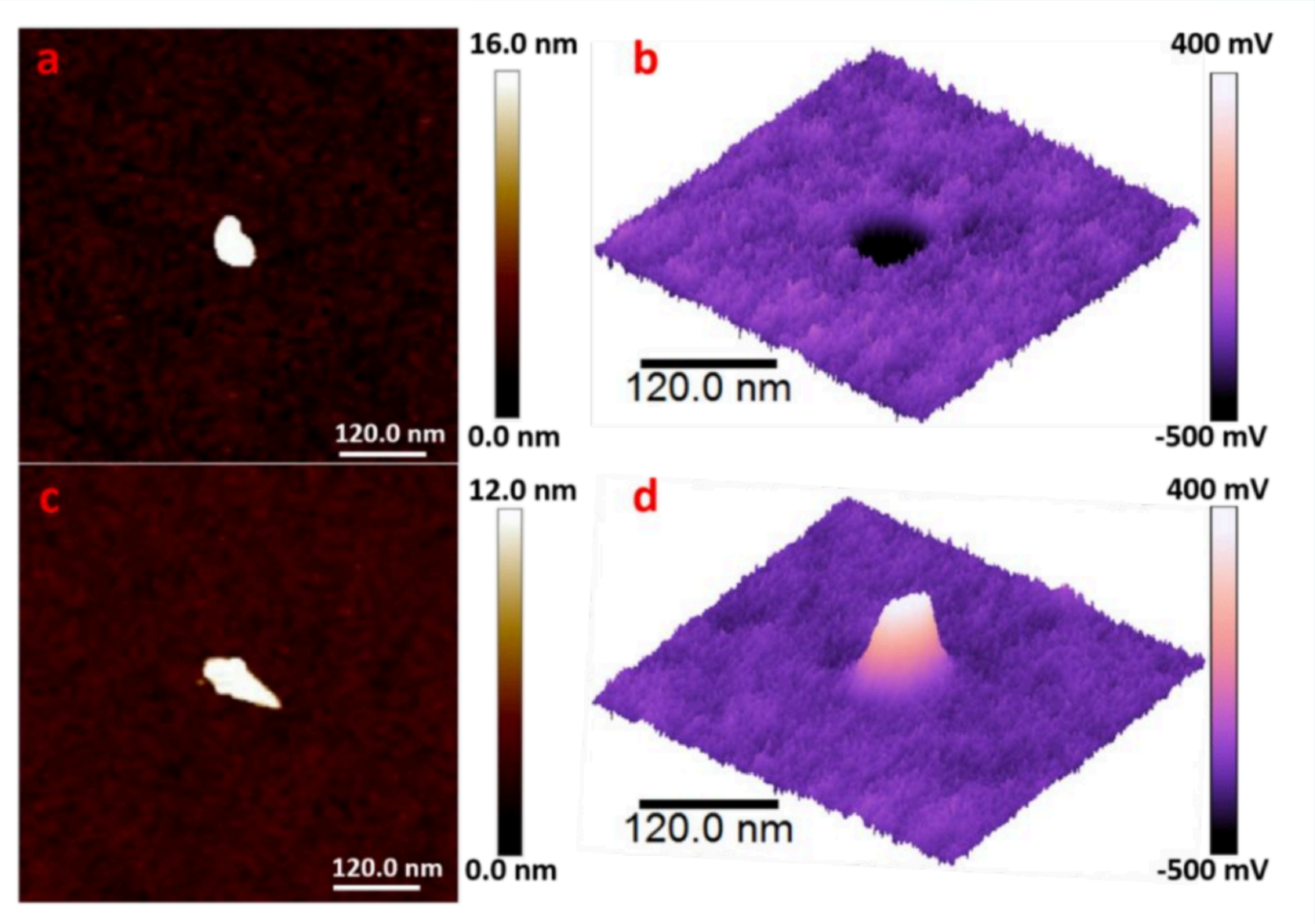
The Real World

Photocatalyst



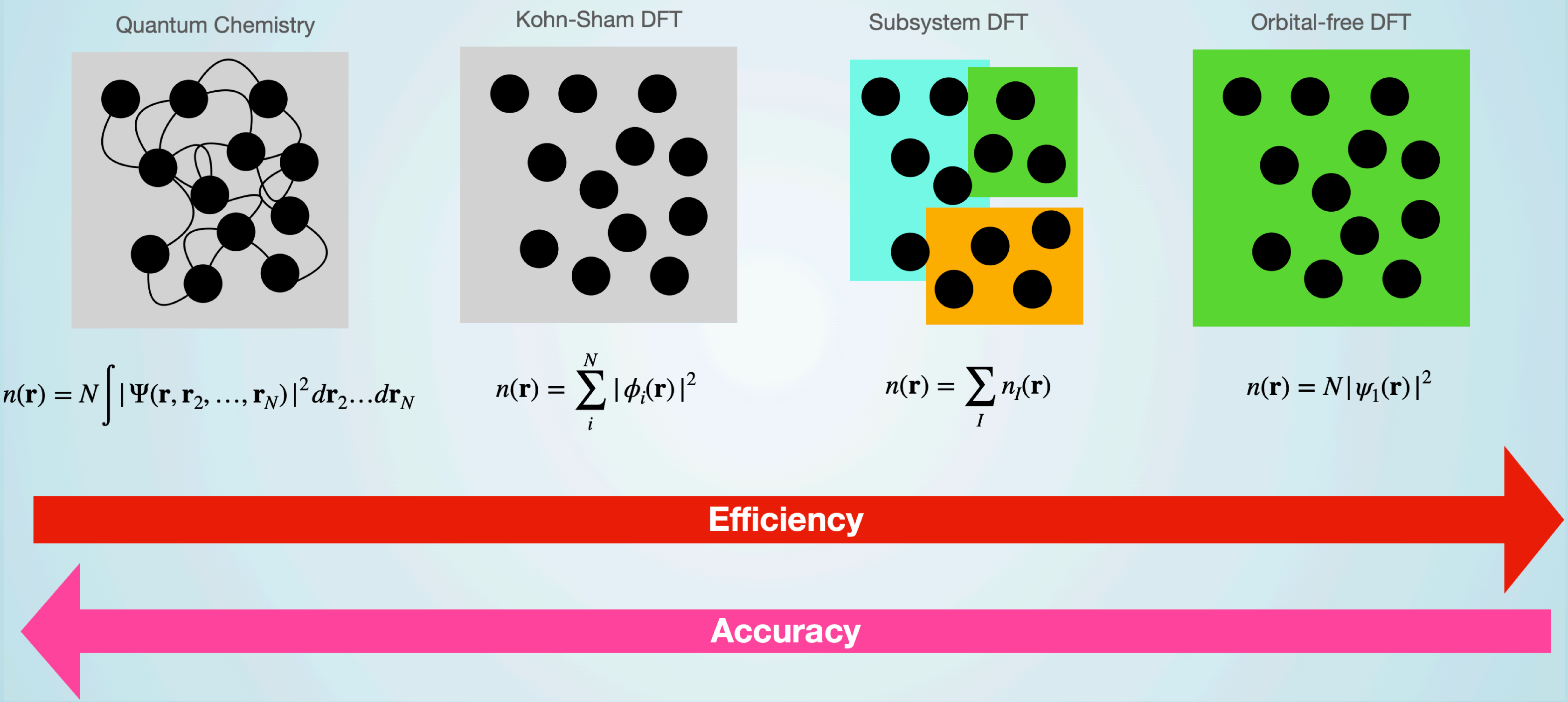
Chem. Comm., 43, 6551 (2009)

Catalytic nanoparticles



PCCP, 21, 15080 (2019)

Available electronic structure methods



Quantum Chemistry

$$n(\mathbf{r}) = N\int |\Psi(\mathbf{r},\mathbf{r}_2,\ldots,\mathbf{r}_N)|^2 d\mathbf{r}_2\ldots d\mathbf{r}_N$$

Kohn-Sham DFT

$$n(\mathbf{r}) = \sum_i^N |\phi_i(\mathbf{r})|^2$$

Subsystem DFT

$$n(\mathbf{r}) = \sum_I n_I(\mathbf{r})$$

Orbital-free DFT

$$n(\mathbf{r}) = N|\psi_1(\mathbf{r})|^2$$

Non-interacting kinetic energy

The non-interacting kinetic energy is a crucial quantity in DFT:

$$T_s[n] = \sum_i \omega_i \int \phi_i(\mathbf{r}) \nabla^2 \phi_i(\mathbf{r}) d\mathbf{r}$$

where ω_i are the `occupation numbers`.

Note: it is **not** the true kinetic energy, but it allows us to define the KS equations, which give us the KS orbitals which give us the density which give us the energy!

The KS equations are:

$$-\frac{1}{2}\nabla^2 \phi_i(\mathbf{r}) + v_s(\mathbf{r})\phi_i(\mathbf{r}) = \varepsilon_i \phi_i(\mathbf{r})$$

We have a nice chain of events:

$$\text{KS equations} \rightarrow \{\phi_i\} \rightarrow n(\mathbf{r}) \rightarrow E[n], \text{forces} \ldots$$

where

$$E[n] = T_s[n] + E_H[n] + E_{xc}[n] + \int n(\mathbf{r}) v_{ext}(\mathbf{r}) d\mathbf{r}$$

Note $E_{xc}[n]$, the exchange-correlation functional

What is the `cost` of getting the KS orbitals?

- Must solve KS equations

How expensive is that? Any guesses?

Like solving an eigenvalue problem

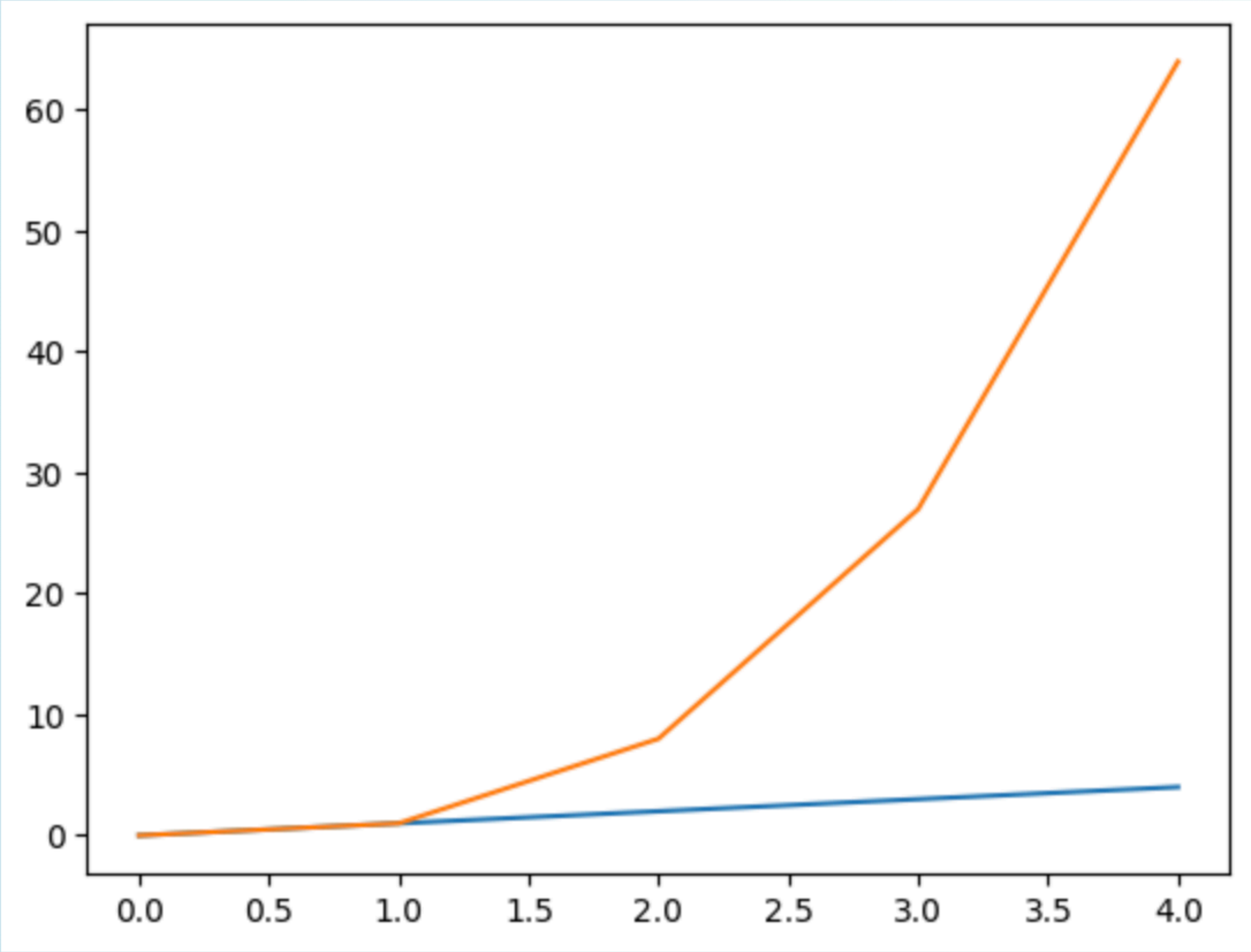
Like diagonalizing a matrix

Oh, no! Bad news, it is like $\mathcal{O}(N^3)$!

Why is it `bad news`?

In [3]:
```python
import matplotlib.pyplot as plt
x=np.array([0,1,2,3,4])
y=x**1
plt.plot(x,y)
plt.plot(x,y**3)
```

Out[3]:
```
[<matplotlib.lines.Line2D at 0x13fff1870>]
```



We need alternative methods if we wish to approach *realistic* systems

An option: Orbital-free DFT (OF-DFT)

- The variational function is the electron density <u>only</u>

$$n(\mathbf{r}) = N|\psi(\mathbf{r})|^2$$

- The energy functional is a **pure** functional of $n(\mathbf{r})$

$$E[n] = T_s[n] + E_H[n] + E_{xc}[n] + \int n(\mathbf{r}) v_{ext}(\mathbf{r}) d\mathbf{r}$$

- Ground state is found by direct minimization

$$n_0(\mathbf{r}) = \arg\min_n \left[E[n] - \mu \left(\int n(\mathbf{r}) d\mathbf{r} - N \right) \right]$$

Opportunities: (1) No need to diagonalize any matrix! (2) Complexity $\mathcal{O}[N \ln N]$

Problem: $T_s[n]$ exists, but it is not known...

Orbital-free vs Kohn-Sham DFT

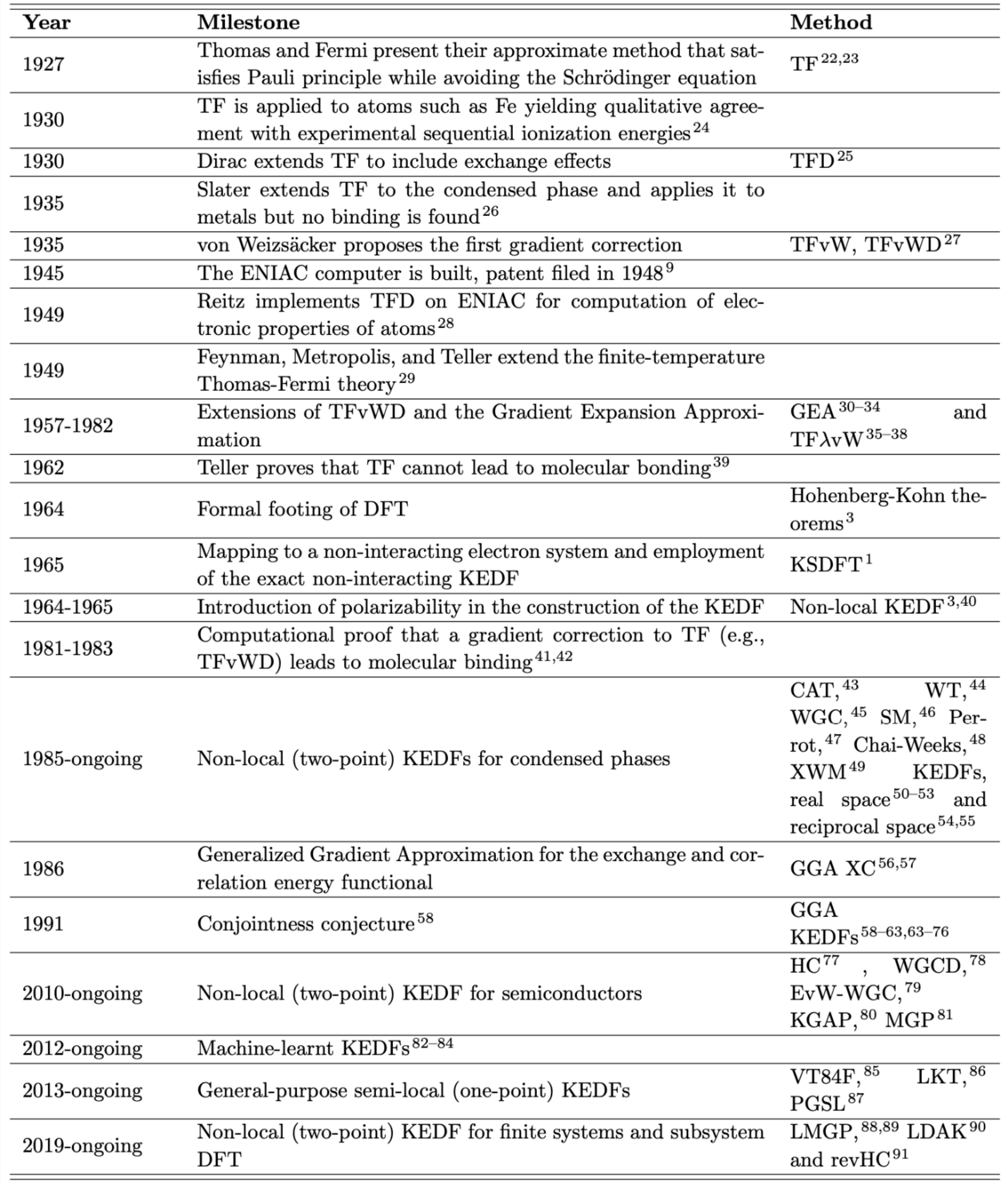
- KS and OF energy functionals are the same

$$E[n] = T_s[n] + \int n(\mathbf{r})v_{ext}(\mathbf{r})d\mathbf{r} + E_{xc}[n] + E_H[n]$$

- **In practice**, the noninteracting kinetic energy is treated differently

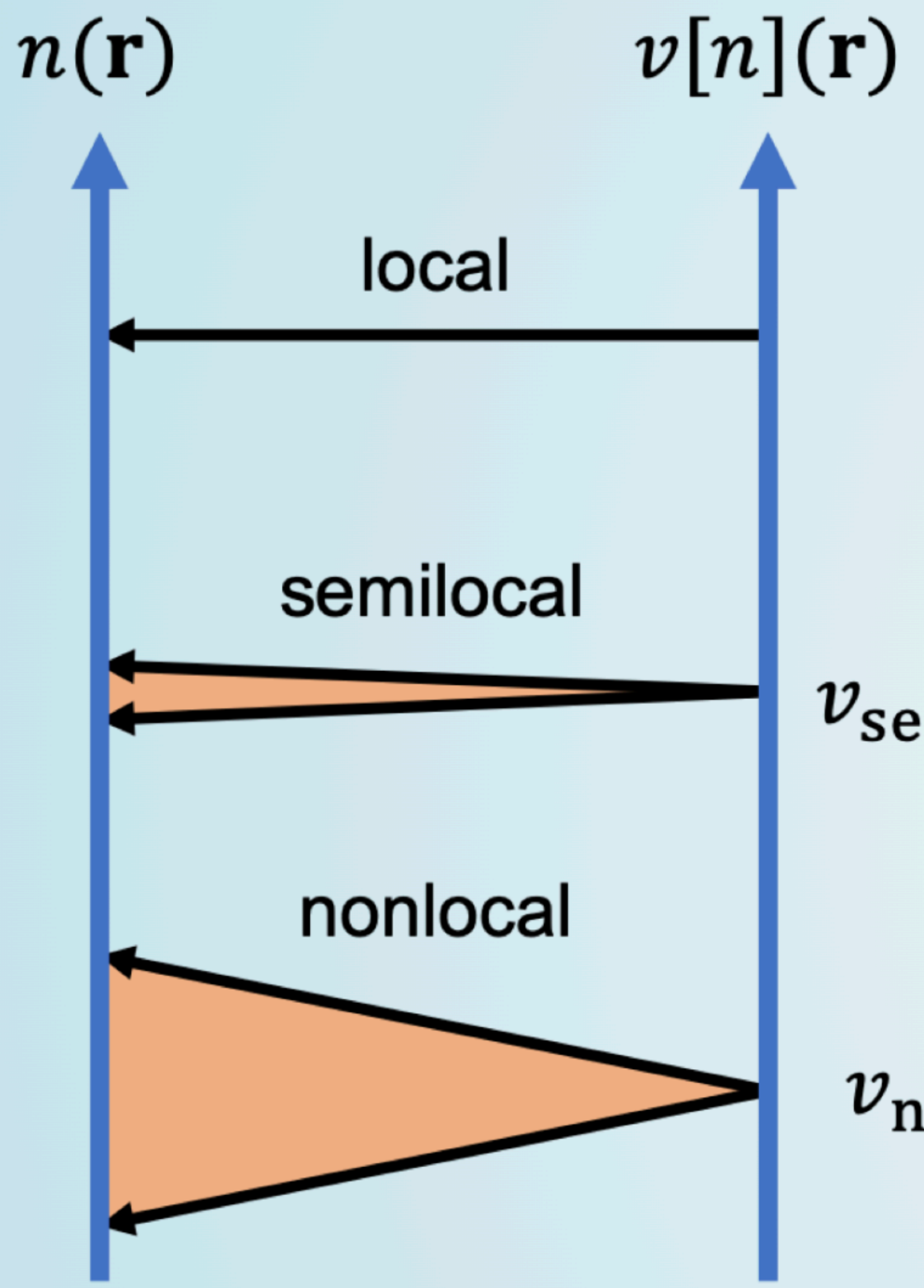
$$T_s[n] \text{ vs } T_s[\{\psi_i[n]\}$$

$T_s[n]$ a long history…



Year	Milestone	Method
1927	Thomas and Fermi present their approximate method that satisfies Pauli principle while avoiding the Schrödinger equation	TF[22,23]
1930	TF is applied to atoms such as Fe yielding qualitative agreement with experimental sequential ionization energies[24]	
1930	Dirac extends TF to include exchange effects	TFD[25]
1935	Slater extends TF to the condensed phase and applies it to metals but no binding is found[26]	
1935	von Weizsäcker proposes the first gradient correction	TFvW, TFvWD[27]
1945	The ENIAC computer is built, patent filed in 1948[9]	
1949	Reitz implements TFD on ENIAC for computation of electronic properties of atoms[28]	
1949	Feynman, Metropolis, and Teller extend the finite-temperature Thomas-Fermi theory[29]	
1957-1982	Extensions of TFvWD and the Gradient Expansion Approximation	GEA[30–34] and TFλvW[35–38]
1962	Teller proves that TF cannot lead to molecular bonding[39]	
1964	Formal footing of DFT	Hohenberg-Kohn theorems[3]
1965	Mapping to a non-interacting electron system and employment of the exact non-interacting KEDF	KSDFT[1]
1964-1965	Introduction of polarizability in the construction of the KEDF	Non-local KEDF[3,40]
1981-1983	Computational proof that a gradient correction to TF (e.g., TFvWD) leads to molecular binding[41,42]	
1985-ongoing	Non-local (two-point) KEDFs for condensed phases	CAT,[43] WT,[44] WGC,[45] SM,[46] Perrot,[47] Chai-Weeks,[48] XWM[49] KEDFs, real space[50–53] and reciprocal space[54,55]
1986	Generalized Gradient Approximation for the exchange and correlation energy functional	GGA XC[56,57]
1991	Conjointness conjecture[58]	GGA KEDFs[58–63,63–76]
2010-ongoing	Non-local (two-point) KEDF for semiconductors	HC[77] , WGCD,[78] EvW-WGC,[79] KGAP,[80] MGP[81]
2012-ongoing	Machine-learnt KEDFs[82–84]	
2013-ongoing	General-purpose semi-local (one-point) KEDFs	VT84F,[85] LKT,[86] PGSL[87]
2019-ongoing	Non-local (two-point) KEDF for finite systems and subsystem DFT	LMGP,[88,89] LDAK[90] and revHC[91]

$T_s[n]$ options: (semi)local and nonlocal functionals



$$v_{\text{local}}(\mathbf{r}) \propto n^{2/3}(\mathbf{r})$$

$$v_{\text{semilocal}}(\mathbf{r}) \propto n(\mathbf{r})\, F(n, |\nabla n|\,)(\mathbf{r})$$

$$v_{\text{nonlocal}}(\mathbf{r}) \propto \int \omega[n](\mathbf{r},\mathbf{r}')n(\mathbf{r}')d\mathbf{r}'$$

Nonlocal with density **independent** kernel

$$T_{NL}[n] = \left\langle n(\mathbf{r})^{\alpha} \middle| \omega(\mathbf{r},\mathbf{r}') \middle| n(\mathbf{r}')^{\beta} \right\rangle$$

Wang & Teter, PRB (1992)

Nonlocal with density **dependent** kernel

$$T_{NL}[n] = \left\langle n(\mathbf{r})^{\alpha} \middle| \omega[n](\mathbf{r},\mathbf{r}') \middle| n(\mathbf{r}')^{\beta} \right\rangle$$

Huang & Carter, PRB (2010)

Mi & Pavanello, PRB (2019)

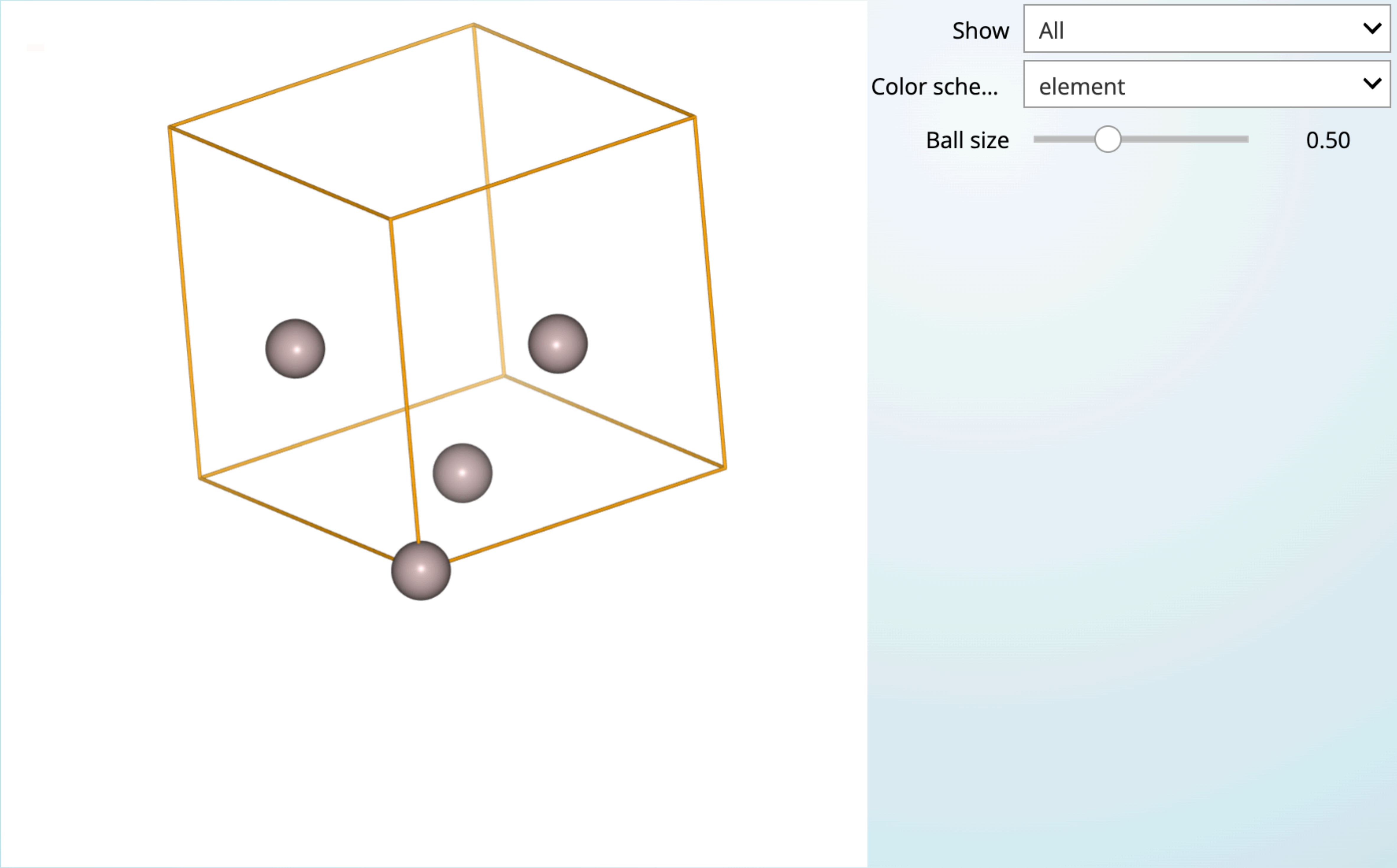
Wenhui Mi, MP JCP (2018) • Wenhui Mi, MP PRB (2019) • Xuecheng Shao, WM, MP PRB (2021) Xuecheng Shao, WM, MP JPCL (2021) • Xuecheng Shao, WM, MP JCTC (2021) • Wenhui Mi, MP JPCL (2020)

How hard is it to run OF-DFT simulations?

First generate a model system: bulk Al (cubic cell)

In [4]:

```
from ase.build import bulk
atoms = bulk('Al', 'fcc', a=4.05, cubic=True)
ions = Ions.from_ase(atoms)
view(ions, viewer='ngl')
```



Show: All

Color sche...: element

Ball size: 0.50

Then run a OF-DFT simulation with `DFTpy`

In [8]:
```
IFrame('http://dftpy.rutgers.edu', width=800, height=300)
```

Out[8]:

DFTpy: Density Functional Theory with Python

In [9]:
```
nr = ecut2nr(ecut=35, lattice=ions.cell)
nr
```

Out[9]:
```
array([20, 20, 20], dtype=int32)
```

In [10]:
```
grid = DirectGrid(lattice=ions.cell, nr=nr)
PSEUDO = LocalPseudo(grid = grid, ions=ions, PP_list=PP_list)
rho_ini = DirectField(grid=grid)
rho_ini[:] = ions.get_ncharges()/ions.cell.volume
HARTREE = Functional(type='HARTREE')
XC = Functional(type='XC',name='LDA')
```

```
setting key: Al -> Al_lda.oe01.recpot
```

Let's pick a $T_s[n]$

- $T_s[n]$ also called KEDF. In this example we use Thomas Fermi + von Weizsacker:

$$T_s[n] = \int C_{TF} n^{5/3}(\mathbf{r}) + \frac{\nabla n^{1/2}(\mathbf{r}) \cdot \nabla n^{1/2}(\mathbf{r})}{2} d\mathbf{r}$$

```
In [5]: KE = Functional(type='KEDF',name='x_TF_y_vW')
```

Define "total energy" and minimize $E[n]$

In [11]:

```
evaluator = TotalFunctional(KE=KE, XC=XC, HARTREE=HARTREE, PSEUDO=PSEUDO)
optimization_options = {'econv' : 1e-6*ions.nat}
opt = Optimization(EnergyEvaluator=evaluator, optimization_options = optimization_options,
        optimization_method = 'TN')
%timeit -n1 -r1 rho = opt.optimize_rho(guess_rho=rho_ini)
```

```
Step    Energy(a.u.)            dE              dP              Nd      Nls     Time(s)
0       -8.090977705021E+00     -8.090978E+00   7.877088E-01    1       1       2.329612E-02
1       -8.273226046462E+00     -1.822483E-01   7.033208E-02    2       1       3.579021E-02
2       -8.280858134665E+00     -7.632088E-03   4.803551E-03    7       1       6.396222E-02
3       -8.281101057233E+00     -2.429226E-04   3.640546E-04    5       1       8.519292E-02
4       -8.281135973142E+00     -3.491591E-05   3.590712E-05    6       1       1.159899E-01
5       -8.281138619986E+00     -2.646845E-06   2.500920E-06    5       1       1.379251E-01
6       -8.281138990518E+00     -3.705314E-07   4.568870E-08    8       1       1.676691E-01
#### Density Optimization Converged ####
Chemical potential (a.u.): 0.30115196527266436
Chemical potential (eV)  : 8.194761597103438
170 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

Opportunities: Can manipulate rho as you wish - many functionalities are available in DFTpy

...let's try to re-run it with a different $T_s[n]$.

Re-run of OF-DFT with nonlocal $T_s[n]$

In [12]:

```
KE = Functional(type='KEDF',name='WT')
evaluator = TotalFunctional(KE=KE, XC=XC, HARTREE=HARTREE, PSEUDO=PSEUDO)
optimization_options = {'econv' : 1e-5*ions.nat}
opt = Optimization(EnergyEvaluator=evaluator, optimization_options = optimization_options,
        optimization_method = 'TN')
%timeit -n1 -r2 rho = opt.optimize_rho(guess_rho=rho_ini)
```

```
Step    Energy(a.u.)            dE              dP              Nd      Nls     Time(s)
0       -8.090977705021E+00     -8.090978E+00   7.877088E-01    1       1       6.896973E-03
1       -8.338206651608E+00     -2.472289E-01   5.654301E-02    7       1       5.253196E-02
2       -8.343069740849E+00     -4.863089E-03   3.268387E-03    7       1       9.050608E-02
3       -8.343656792601E+00     -5.870518E-04   3.047555E-04    9       1       1.358750E-01
4       -8.343674713047E+00     -1.792045E-05   2.723036E-05    5       1       1.638479E-01
5       -8.343676820175E+00     -2.107128E-06   1.881566E-06    7       1       1.982689E-01
#### Density Optimization Converged ####
Chemical potential (a.u.): 0.2927374982051514
Chemical potential (eV)  : 7.9657923073878125
Step    Energy(a.u.)            dE              dP              Nd      Nls     Time(s)
0       -8.090977705021E+00     -8.090978E+00   7.877088E-01    1       1       6.319046E-03
1       -8.338206651608E+00     -2.472289E-01   5.654301E-02    7       1       4.166794E-02
2       -8.343069740849E+00     -4.863089E-03   3.268387E-03    7       1       7.957697E-02
3       -8.343656792601E+00     -5.870518E-04   3.047555E-04    9       1       1.243918E-01
4       -8.343674713047E+00     -1.792045E-05   2.723036E-05    5       1       1.514030E-01
5       -8.343676820175E+00     -2.107128E-06   1.881566E-06    7       1       1.866379E-01
#### Density Optimization Converged ####
Chemical potential (a.u.): 0.2927374982051514
Chemical potential (eV)  : 7.9657923073878125
195 ms ± 5.67 ms per loop (mean ± std. dev. of 2 runs, 1 loop each)
```

Nonlocal $T_s[n]$ are somewhat slower than semilocal functionals

How does OF-DFT compare with KS-DFT?

We will run a Quantum ESPRESSO calculation of the same system using QEpy

In [13]:

```python
from qepy.driver import Driver
from qepy.io import QEInput
```

In [14]:

```python
IFrame('http://qepy.rutgers.edu', width=1000, height=400)
```

Out[14]:

QEpy - Quantum ESPRESSO in Python

QEpy turns Quantum ESPRESSO (QE) into a Python DFT engine for nonstandard workflows.

Contributors and funding

- Main author: Xuecheng Shao (Rutgers)
- Oliviero Andreussi (UNT), Davide Ceresoli (CNR, Italy), Matthew Truscott

QEpy's input file is a dictionary

In [15]:
```
qe_options = {
    '&control': {
        'calculation': "'scf'",
        'pseudo_dir': "'./'",
    },
    '&system': {
        'ibrav' : 0,
        'degauss': 0.005,
        'ecutwfc': 30,
        'occupations': "'smearing'"
    },
    'atomic_species': ['Al  26.98 Al.pbe-nl-kjpaw_psl.1.0.0.UPF'],
    'k_points gamma': [],
}
```

!wget http://pseudopotentials.quantum-espresso.org/upf_files/Al.pbe-nl-kjpaw_psl.1.0.0.UPF

In [16]:
```
options = {
    '&electrons': {
        'mixing_beta': 0.5},
    'cell_parameters angstrom':[
        '0.     2.025  2.025',
        '2.025  0.     2.025',
        '2.025  2.025  0.   '],
    'atomic_positions crystal': ['Al    0.0  0.0  0.0'],
    'k_points automatic': ['11 11 11 1 1 1'],
}
```

Initialize QEpy

In [17]:

```
qe_options = QEInput.update_options(options, qe_options=qe_options)
```

In [18]:

```
driver = Driver(qe_options=qe_options, logfile=True)
%timeit -n1 -r1 driver.scf()
```

```
8 s ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

In [19]:

```
driver.get_scf_error()
```

Out[19]:

```
8.733828545114252e-08
```

In [20]:

```
driver.stop()
```

KS-DFT is ★ times slower than nonlocal OF-DFT for this small system.

Poll time

- People say OF-DFT is a "linear-scaling method". Do you agree?

A: Yep! I do!

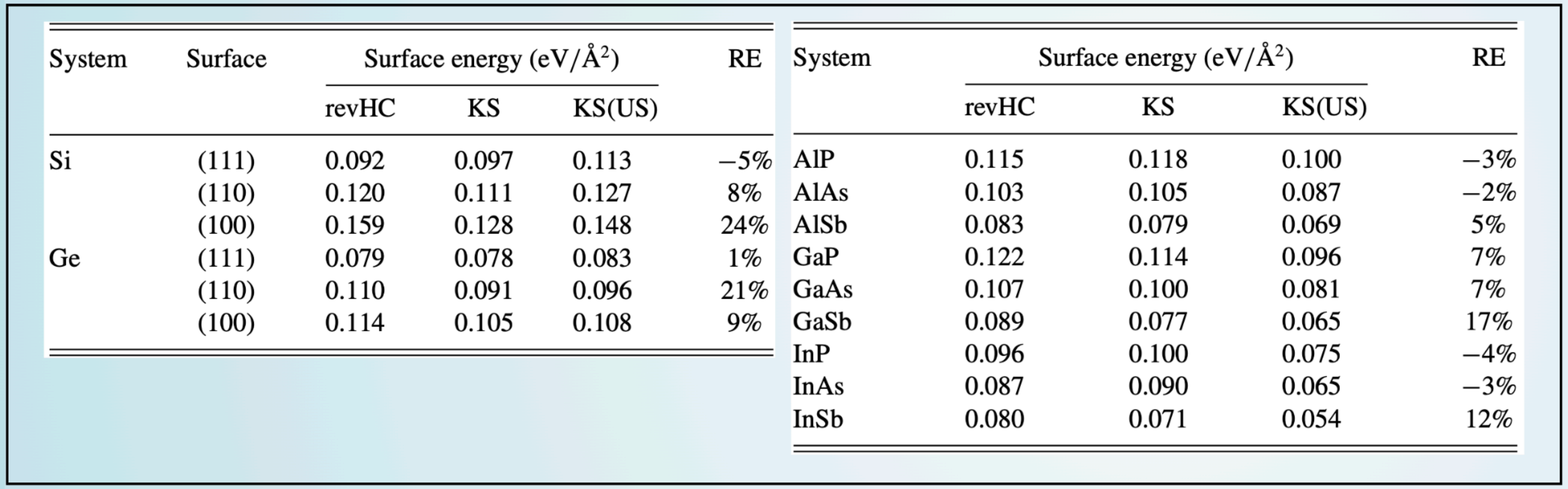
B: Nah. Not really.

C: Sometimes.

Can OF-DFT approach surfaces?

Shao, Mi & Pavanello, Phys. Rev. B **104**, 045118 (2021)

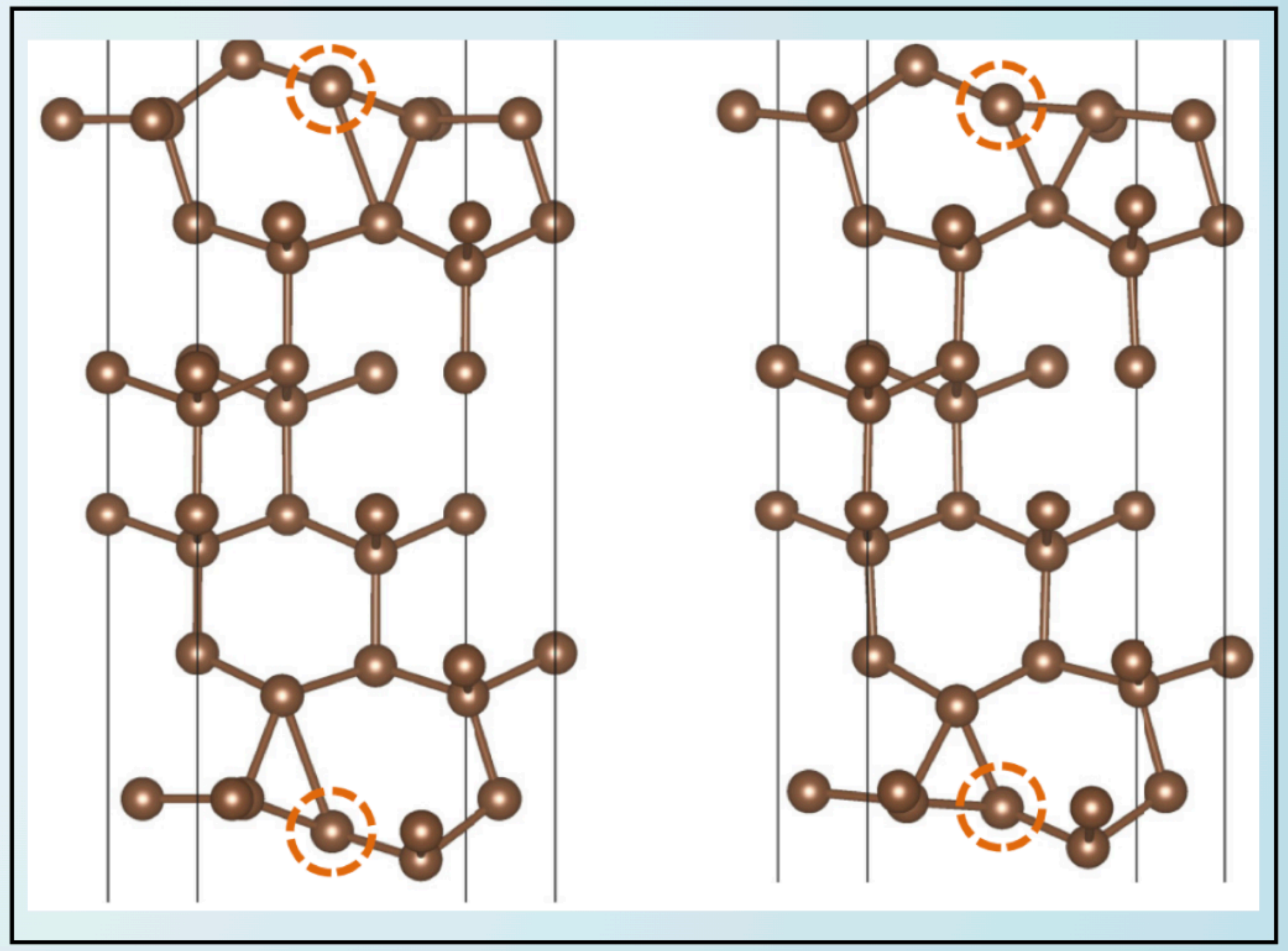
Surface energies



System	Surface	Surface energy (eV/Å^2)			RE
		revHC	KS	KS(US)	
Si	(111)	0.092	0.097	0.113	−5%
	(110)	0.120	0.111	0.127	8%
	(100)	0.159	0.128	0.148	24%
Ge	(111)	0.079	0.078	0.083	1%
	(110)	0.110	0.091	0.096	21%
	(100)	0.114	0.105	0.108	9%

System	Surface energy (eV/Å^2)			RE
	revHC	KS	KS(US)	
AlP	0.115	0.118	0.100	−3%
AlAs	0.103	0.105	0.087	−2%
AlSb	0.083	0.079	0.069	5%
GaP	0.122	0.114	0.096	7%
GaAs	0.107	0.100	0.081	7%
GaSb	0.089	0.077	0.065	17%
InP	0.096	0.100	0.075	−4%
InAs	0.087	0.090	0.065	−3%
InSb	0.080	0.071	0.054	12%

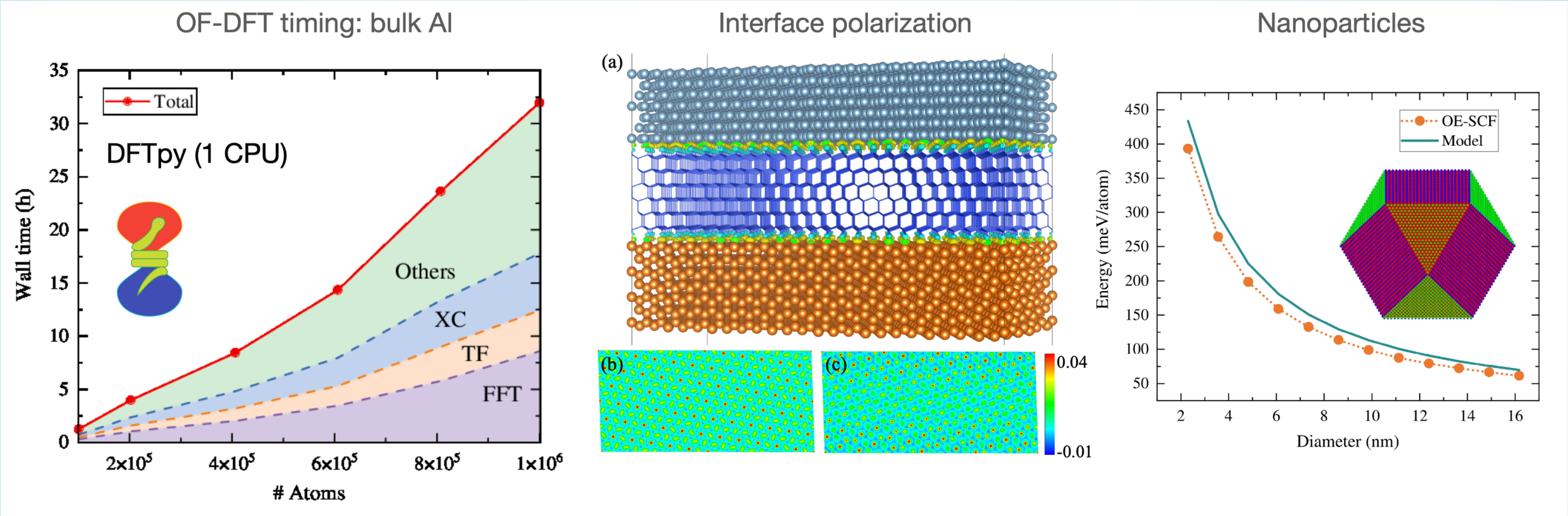
Surface reconstruction Si(111)



Excellent results for difficult semiconductors and metals!

What else can OF-DFT do?

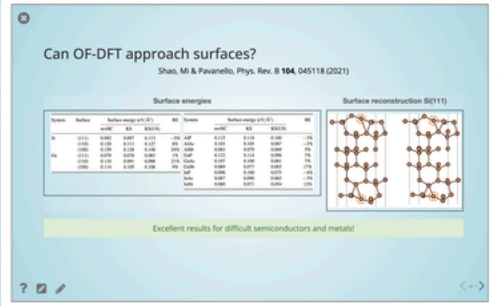
J. Phys. Chem. Lett. **12**, 4134 (2021) ◆ WIREs: Comp. Mol. Sci. **11**, e1482 (2021)


OF-DFT timing: bulk Al

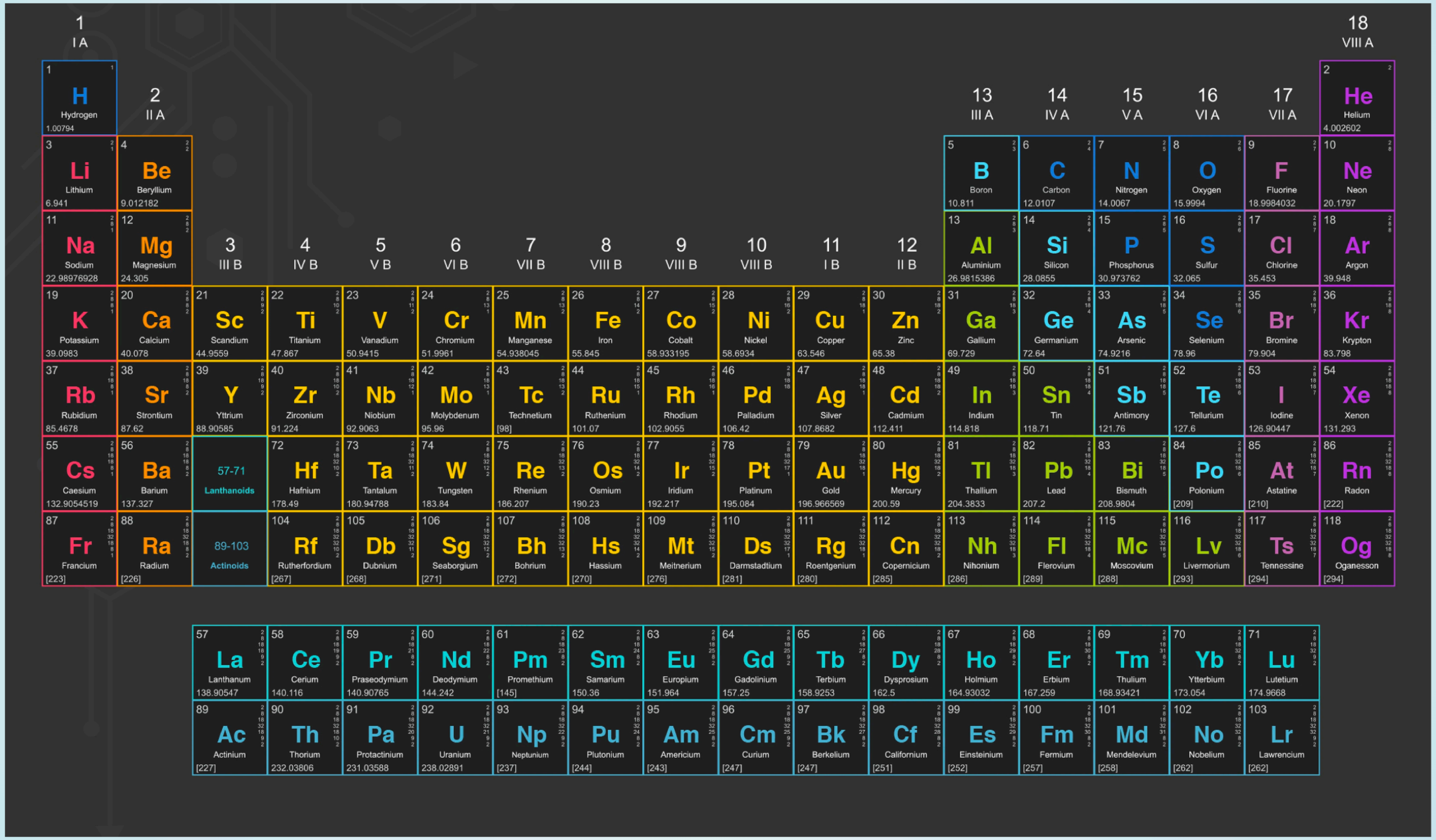
Interface polarization

Nanoparticles

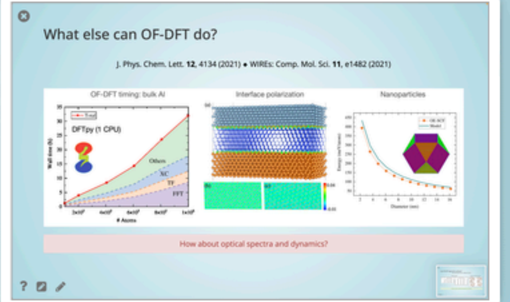
How about optical spectra and dynamics?



OF-DFT is limited...

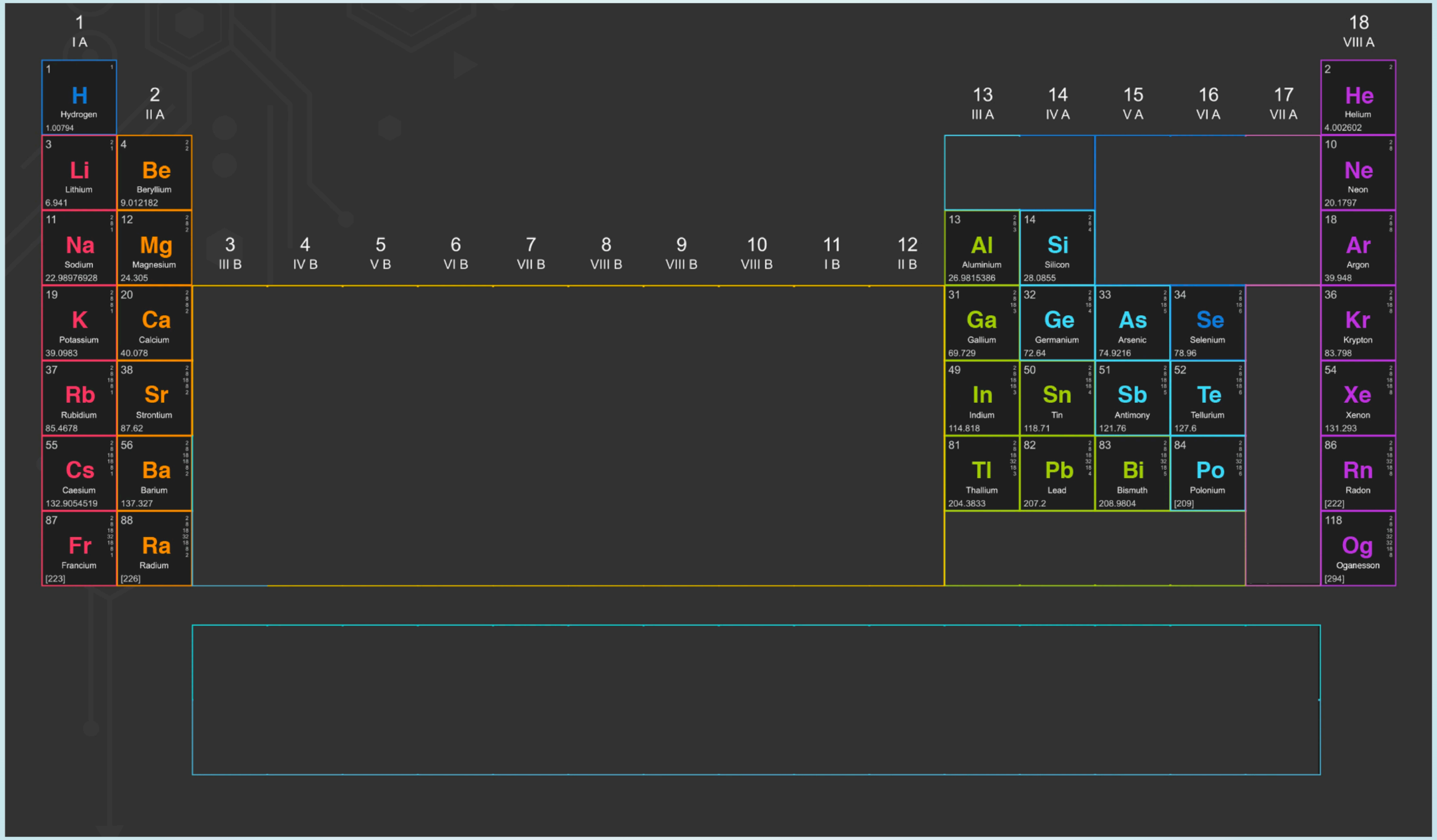


1 I A	2 II A	3 III B	4 IV B	5 V B	6 VI B	7 VII B	8 VIII B	9 VIII B	10 VIII B	11 I B	12 II B	13 III A	14 IV A	15 V A	16 VI A	17 VII A	18 VIII A
1 H Hydrogen 1.00794																	2 He Helium 4.002602
3 Li Lithium 6.941	4 Be Beryllium 9.012182											5 B Boron 10.811	6 C Carbon 12.0107	7 N Nitrogen 14.0067	8 O Oxygen 15.9994	9 F Fluorine 18.9984032	10 Ne Neon 20.1797
11 Na Sodium 22.98976928	12 Mg Magnesium 24.305											13 Al Aluminium 26.9815386	14 Si Silicon 28.0855	15 P Phosphorus 30.973762	16 S Sulfur 32.065	17 Cl Chlorine 35.453	18 Ar Argon 39.948
19 K Potassium 39.0983	20 Ca Calcium 40.078	21 Sc Scandium 44.9559	22 Ti Titanium 47.867	23 V Vanadium 50.9415	24 Cr Chromium 51.9961	25 Mn Manganese 54.938045	26 Fe Iron 55.845	27 Co Cobalt 58.933195	28 Ni Nickel 58.6934	29 Cu Copper 63.546	30 Zn Zinc 65.38	31 Ga Gallium 69.729	32 Ge Germanium 72.64	33 As Arsenic 74.9216	34 Se Selenium 78.96	35 Br Bromine 79.904	36 Kr Krypton 83.798
37 Rb Rubidium 85.4678	38 Sr Strontium 87.62	39 Y Yttrium 88.90585	40 Zr Zirconium 91.224	41 Nb Niobium 92.9063	42 Mo Molybdenum 95.96	43 Tc Technetium [98]	44 Ru Ruthenium 101.07	45 Rh Rhodium 102.9055	46 Pd Palladium 106.42	47 Ag Silver 107.8682	48 Cd Cadmium 112.411	49 In Indium 114.818	50 Sn Tin 118.71	51 Sb Antimony 121.76	52 Te Tellurium 127.6	53 I Iodine 126.90447	54 Xe Xenon 131.293
55 Cs Caesium 132.9054519	56 Ba Barium 137.327	57-71 Lanthanoids	72 Hf Hafnium 178.49	73 Ta Tantalum 180.94788	74 W Tungsten 183.84	75 Re Rhenium 186.207	76 Os Osmium 190.23	77 Ir Iridium 192.217	78 Pt Platinum 195.084	79 Au Gold 196.966569	80 Hg Mercury 200.59	81 Tl Thallium 204.3833	82 Pb Lead 207.2	83 Bi Bismuth 208.9804	84 Po Polonium [209]	85 At Astatine [210]	86 Rn Radon [222]
87 Fr Francium [223]	88 Ra Radium [226]	89-103 Actinoids	104 Rf Rutherfordium [267]	105 Db Dubnium [268]	106 Sg Seaborgium [271]	107 Bh Bohrium [272]	108 Hs Hassium [270]	109 Mt Meitnerium [276]	110 Ds Darmstadtium [281]	111 Rg Roentgenium [280]	112 Cn Copernicium [285]	113 Nh Nihonium [286]	114 Fl Flerovium [289]	115 Mc Moscovium [288]	116 Lv Livermorium [293]	117 Ts Tennessine [294]	118 Og Oganesson [294]
		57 La Lanthanum 138.90547	58 Ce Cerium 140.116	59 Pr Praseodymium 140.90765	60 Nd Deodymium 144.242	61 Pm Promethium [145]	62 Sm Samarium 150.36	63 Eu Europium 151.964	64 Gd Gadolinium 157.25	65 Tb Terbium 158.9253	66 Dy Dysprosium 162.5	67 Ho Holmium 164.93032	68 Er Erbium 167.259	69 Tm Thulium 168.93421	70 Yb Ytterbium 173.054	71 Lu Lutetium 174.9668	
		89 Ac Actinium [227]	90 Th Thorium 232.03806	91 Pa Protactinium 231.03588	92 U Uranium 238.02891	93 Np Neptunium [237]	94 Pu Plutonium [244]	95 Am Americium [243]	96 Cm Curium [247]	97 Bk Berkelium [247]	98 Cf Californium [251]	99 Es Einsteinium [252]	100 Fm Fermium [257]	101 Md Mendelevium [258]	102 No Nobelium [262]	103 Lr Lawrencium [262]	



What else can OF-DFT do?

OF-DFT is limited… but why?



1 I A	2 II A	3 III B	4 IV B	5 V B	6 VI B	7 VII B	8 VIII B	9 VIII B	10 VIII B	11 I B	12 II B	13 III A	14 IV A	15 V A	16 VI A	17 VII A	18 VIII A
1 H Hydrogen 1.00794																	2 He Helium 4.002602
3 Li Lithium 6.941	4 Be Beryllium 9.012182																10 Ne Neon 20.1797
11 Na Sodium 22.98976928	12 Mg Magnesium 24.305											13 Al Aluminium 26.9815386	14 Si Silicon 28.0855				18 Ar Argon 39.948
19 K Potassium 39.0983	20 Ca Calcium 40.078											31 Ga Gallium 69.729	32 Ge Germanium 72.64	33 As Arsenic 74.9216	34 Se Selenium 78.96		36 Kr Krypton 83.798
37 Rb Rubidium 85.4678	38 Sr Strontium 87.62											49 In Indium 114.818	50 Sn Tin 118.71	51 Sb Antimony 121.76	52 Te Tellurium 127.6		54 Xe Xenon 131.293
55 Cs Caesium 132.9054519	56 Ba Barium 137.327											81 Tl Thallium 204.3833	82 Pb Lead 207.2	83 Bi Bismuth 208.9804	84 Po Polonium [209]		86 Rn Radon [222]
87 Fr Francium [223]	88 Ra Radium [226]																118 Og Oganesson [294]

What are nonlocal PPs?

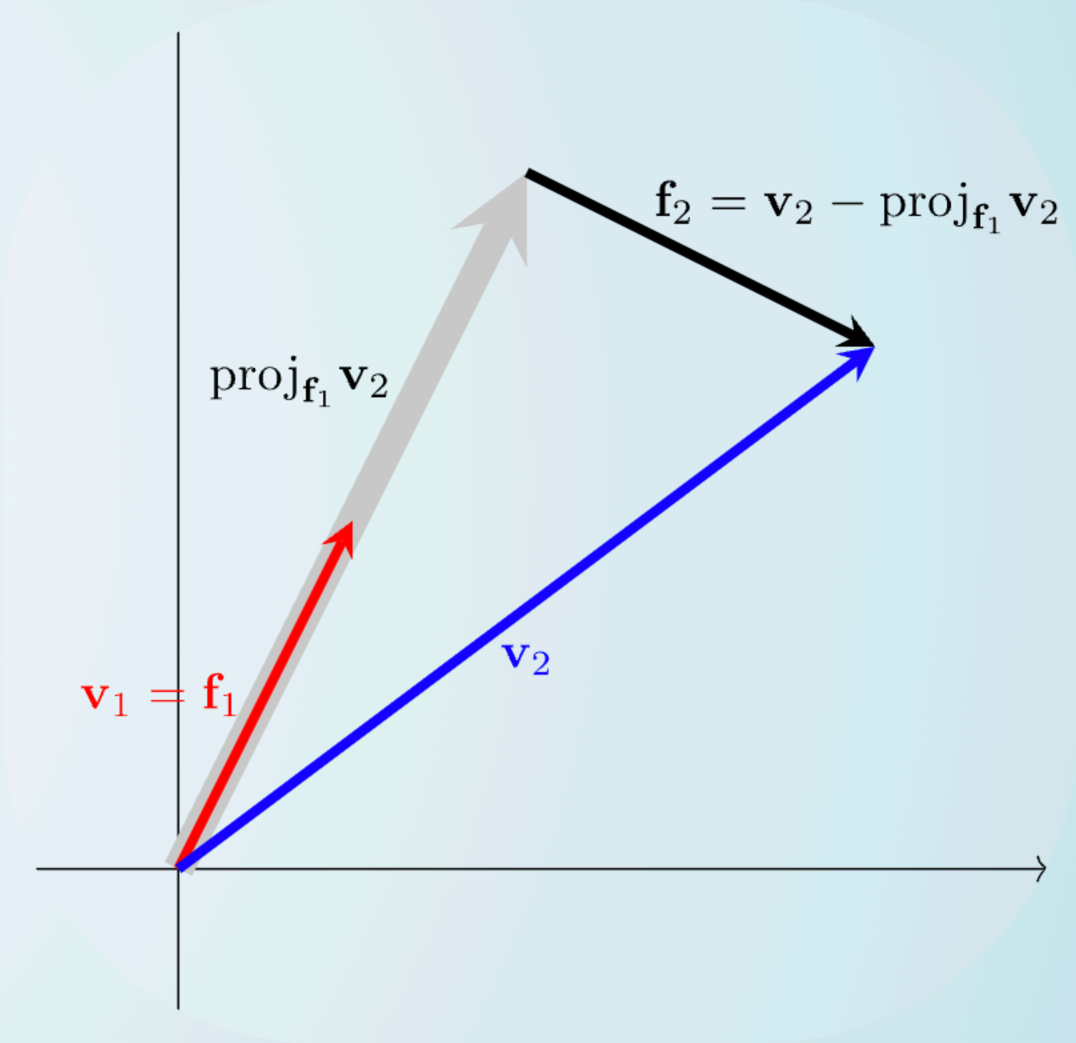
OF-DFT is limited… but why?

Core Electrons

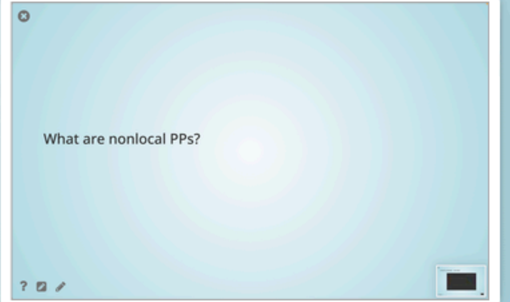
Using pseudo-potentials, core electrons are "hidden"

Valence Electrons

Need to be orthogonal to the core electrons



The Gram-Schmidt orthogonalization is a good example


What are nonlocal PPs?

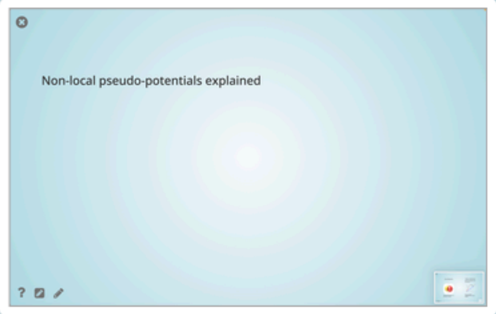
Non-local pseudo-potentials explained

Apply Gram-Schmidt to the set of valence, $|\phi\rangle$, and core, $|\mathrm{core}\rangle$, electrons:

$$|\phi\rangle \to |\phi\rangle - |\mathrm{core}\rangle\langle\mathrm{core}|\phi\rangle = \Big[1 - |\mathrm{core}\rangle\langle\mathrm{core}|\Big]|\phi\rangle$$

$\hat{P}_{nl} = \Big[1 - |\mathrm{core}\rangle\langle\mathrm{core}|\Big]$ is the **non-local pseudopotential**

In OF-DFT $|\phi\rangle$ is not available…


Non-local pseudo-potentials explained

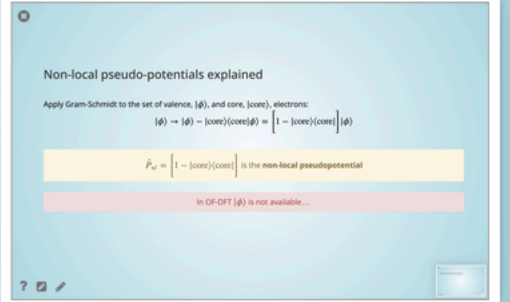
Poll time / challenge

- How would you fix OF-DFT's impossibility to use nonlocal PPs?

A: Use KS-DFT for atoms only, or smaller fragments of the system..and then combine them together with OF-DFT

B: Throw OF-DFT away!

C: Introduce nonlocal pseudopotentials in OF-DFT somehow



Subsystem DFT (sDFT) - Theory

Idea:

- Split system into (smaller) subsystems: $n(\mathbf{r}) = \sum_I n_I(\mathbf{r})$
- Use KS-DFT **or** OF-DFT for the subsystems

- Energy is split in *additive* and *non-additive* terms:

$$E[n] \equiv E[\{n_I\}] = \sum_I E[n_I] + E^{nadd}[\{n_I\}]$$

- The nonadditive energy given by:

$$E^{nadd}[\{n_I\}] = E_{Coul}[n] - \sum_I E_{Coul}[n_I] + E_{xc}[n] - \sum_I E_{xc}[n_I] + T_s[n] - \sum_I T_s[n_I]$$

Problem: The non-additive energy contains the non-additive kinetic energy, $T_s^{nadd}[\{n_I\}]$, which not known.

Opportunity: We can use *fancy non-local* $T_s^{nadd}[\{n_I\}]$, previously developed for OF-DFT.

Poll time / challenge

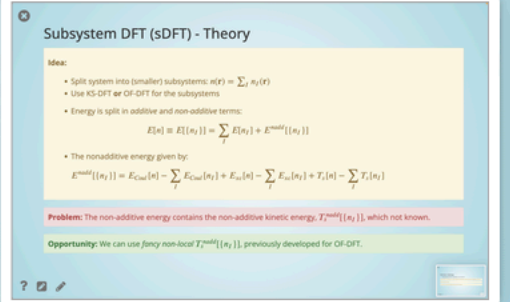
sDFT implementations

Our codes:

eQE	eDFTpy
2017-2022	2022-present

The competition:

Serenity	CP2k
Neugebauer (Münster)	Luber, Hutter (Zurich)



Subsystem DFT (sDFT) - Theory

Idea:

- Split system into (smaller) subsystems: $n(\mathbf{r}) = \sum_I n_I(\mathbf{r})$
- Use KS-DFT **or** OF-DFT for the subsystems
- Energy is split in *additive* and *non-additive* terms:

$$E[n] \equiv E[\{n_I\}] = \sum_I E[n_I] + E^{nadd}[\{n_I\}]$$

- The nonadditive energy given by:

$$E^{nadd}[\{n_I\}] = E_{Coul}[n] - \sum_I E_{Coul}[n_I] + E_{xc}[n] - \sum_I E_{xc}[n_I] + T_s[n] - \sum_I T_s[n_I]$$

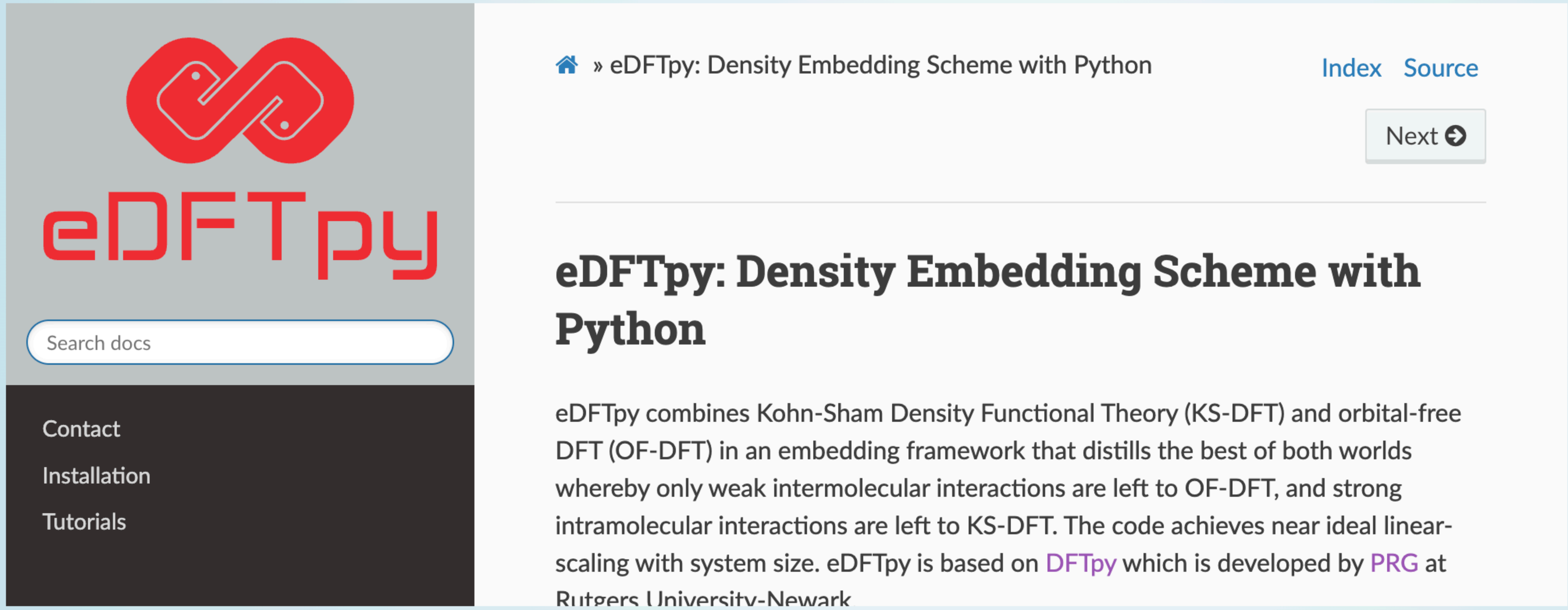
Problem: The non-additive energy contains the non-additive kinetic energy, $T_s^{nadd}[\{n_I\}]$, which not known.

Opportunity: We can use *fancy non-local* $T_s^{nadd}[\{n_I\}]$, previously developed for OF-DFT.

Want to learn more about `eDFTpy`?

In [21]:
```
IFrame('http://edftpy.rutgers.edu', width=1000, height=400)
```

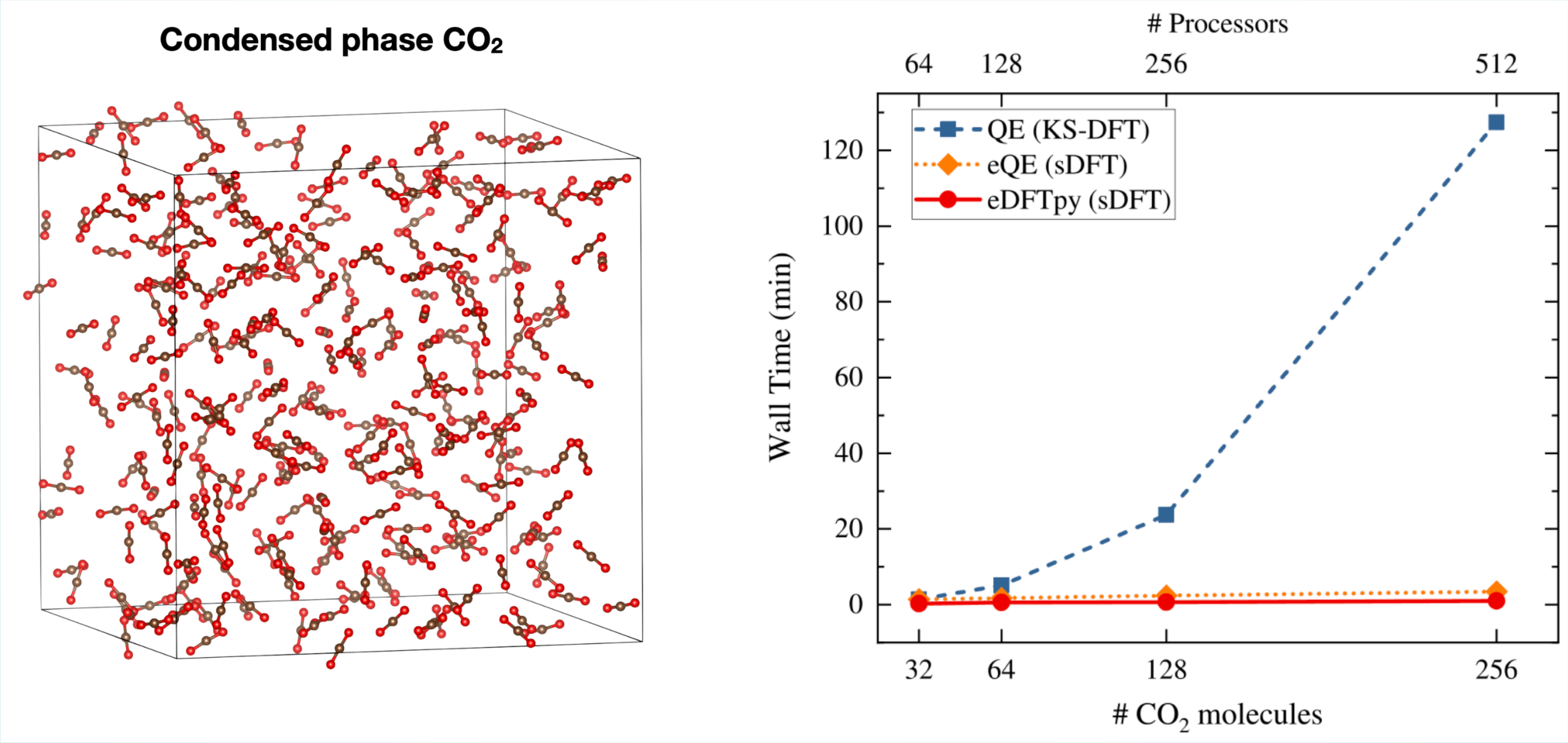
Out[21]:



eDFTpy: Density Embedding Scheme with Python

eDFTpy combines Kohn-Sham Density Functional Theory (KS-DFT) and orbital-free DFT (OF-DFT) in an embedding framework that distills the best of both worlds whereby only weak intermolecular interactions are left to OF-DFT, and strong intramolecular interactions are left to KS-DFT. The code achieves near ideal linear-scaling with system size. eDFTpy is based on DFTpy which is developed by PRG at Rutgers University-Newark

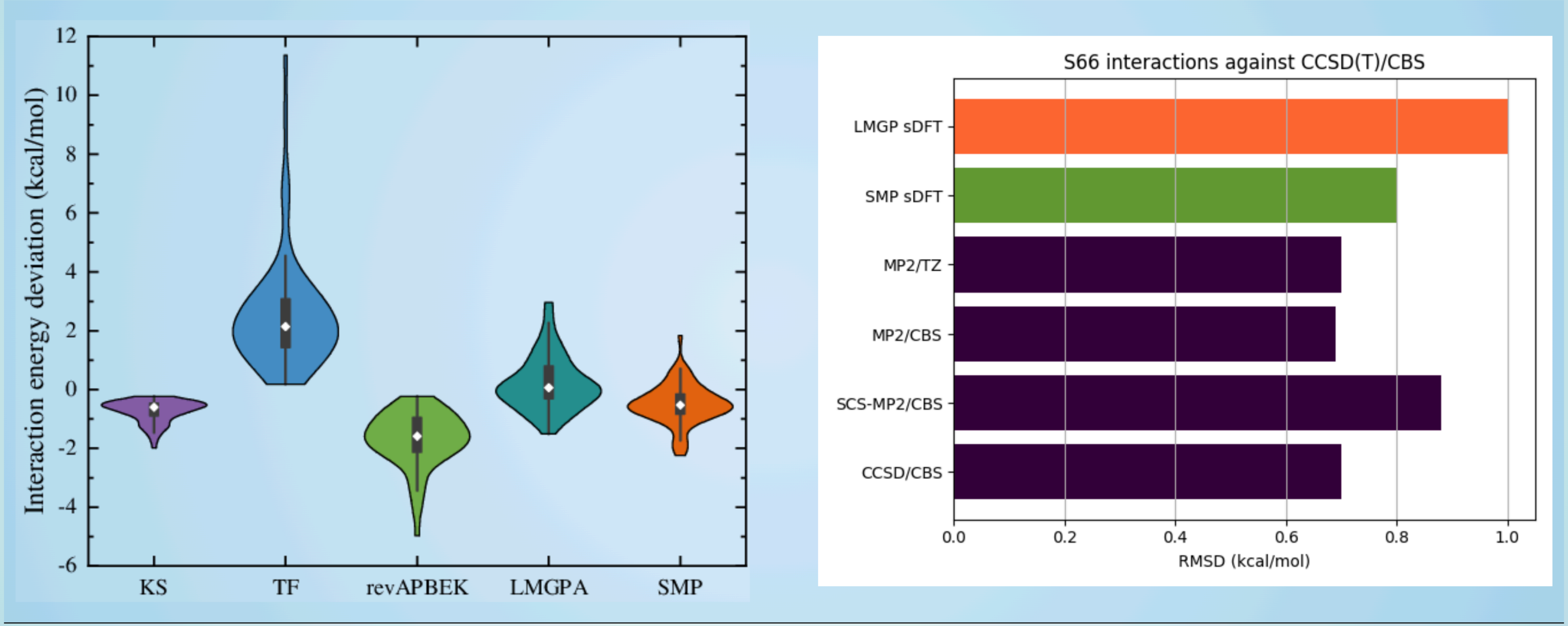
Subsystem DFT (sDFT) - Timings



- Comp. Phys. Comm. **269**, 108122 (2021)
- Int. J. Quantum Chem. **117** e25401 (2017)
- J. Phys. Chem. Lett. **10** 7554 (2019)

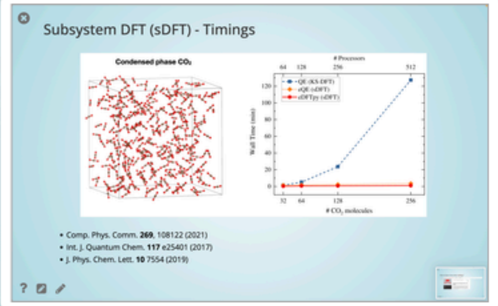
How accurate is sDFT?

Deviation against CCSD(T) for the S66 test set



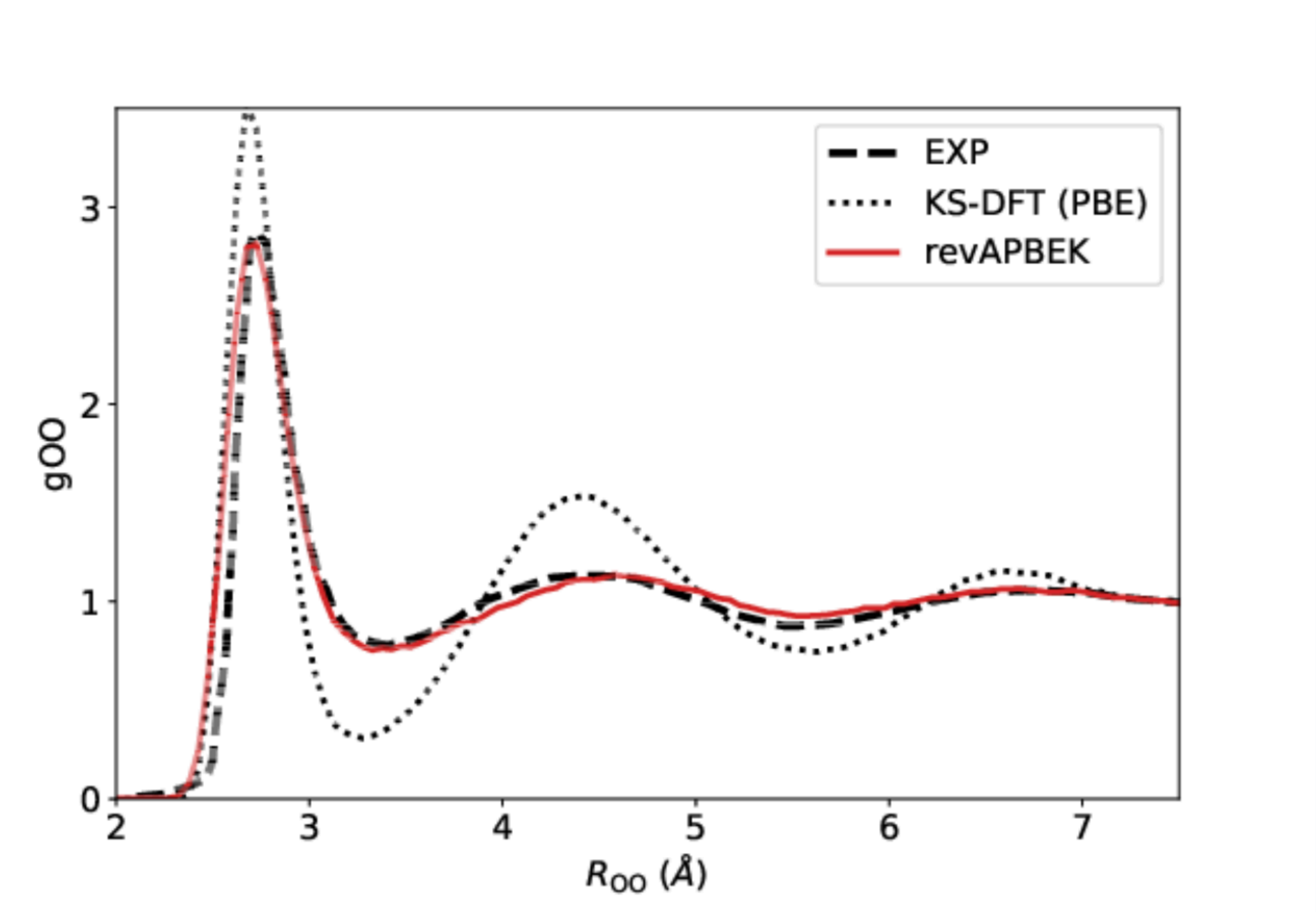
- Comp. Phys. Comm. **269**, 108122 (2021)
- J. Chem. Theory Comput. **17** 3455 (2021)
- J. Phys. Chem. Lett. **11** 272 (2020)

- J. Chem. Theory Comput. **7** 2427 (2011)

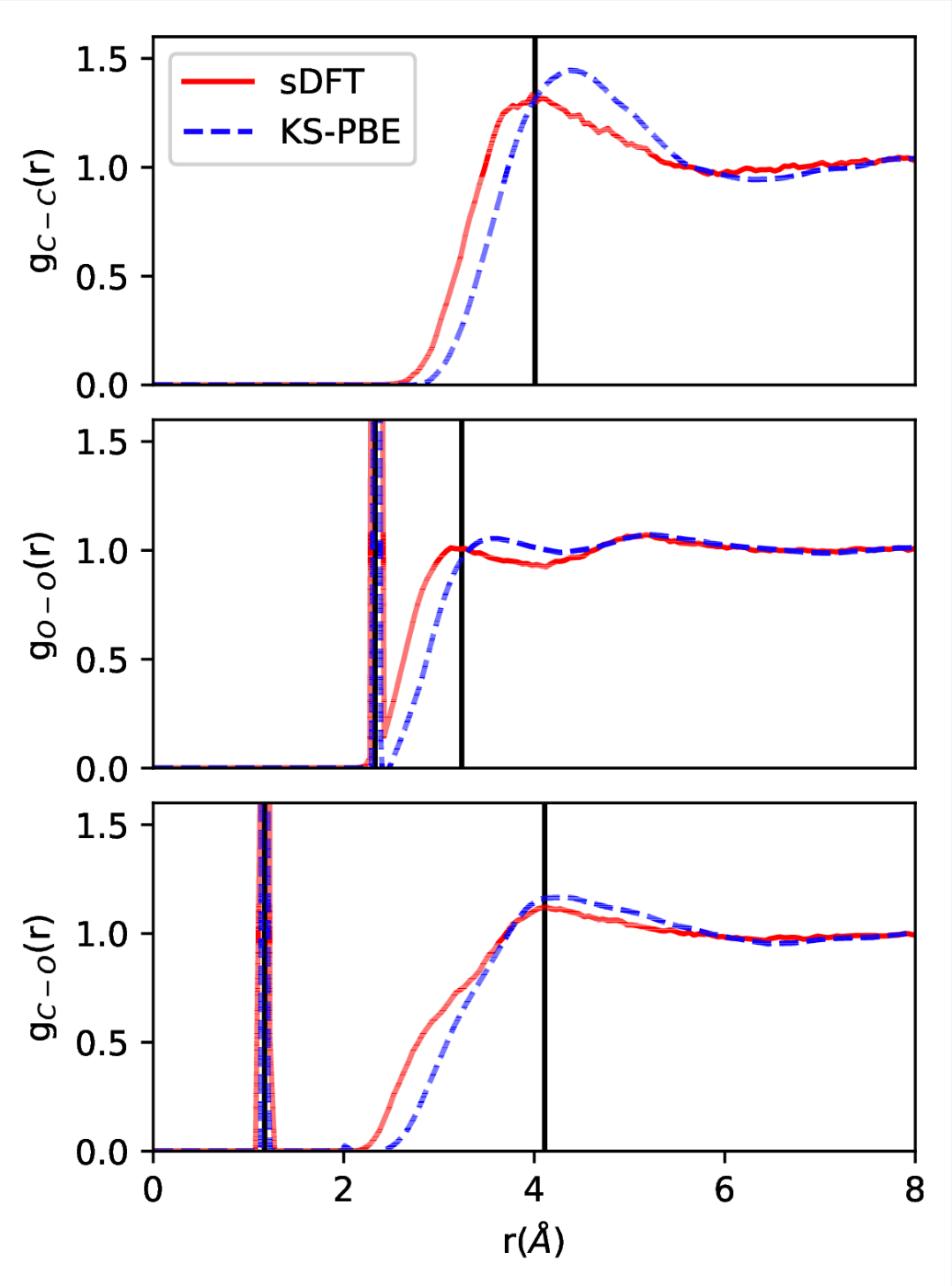

Subsystem DFT (sDFT) - Timings

sDFT success stories

Liquid water



Fluid CO_2



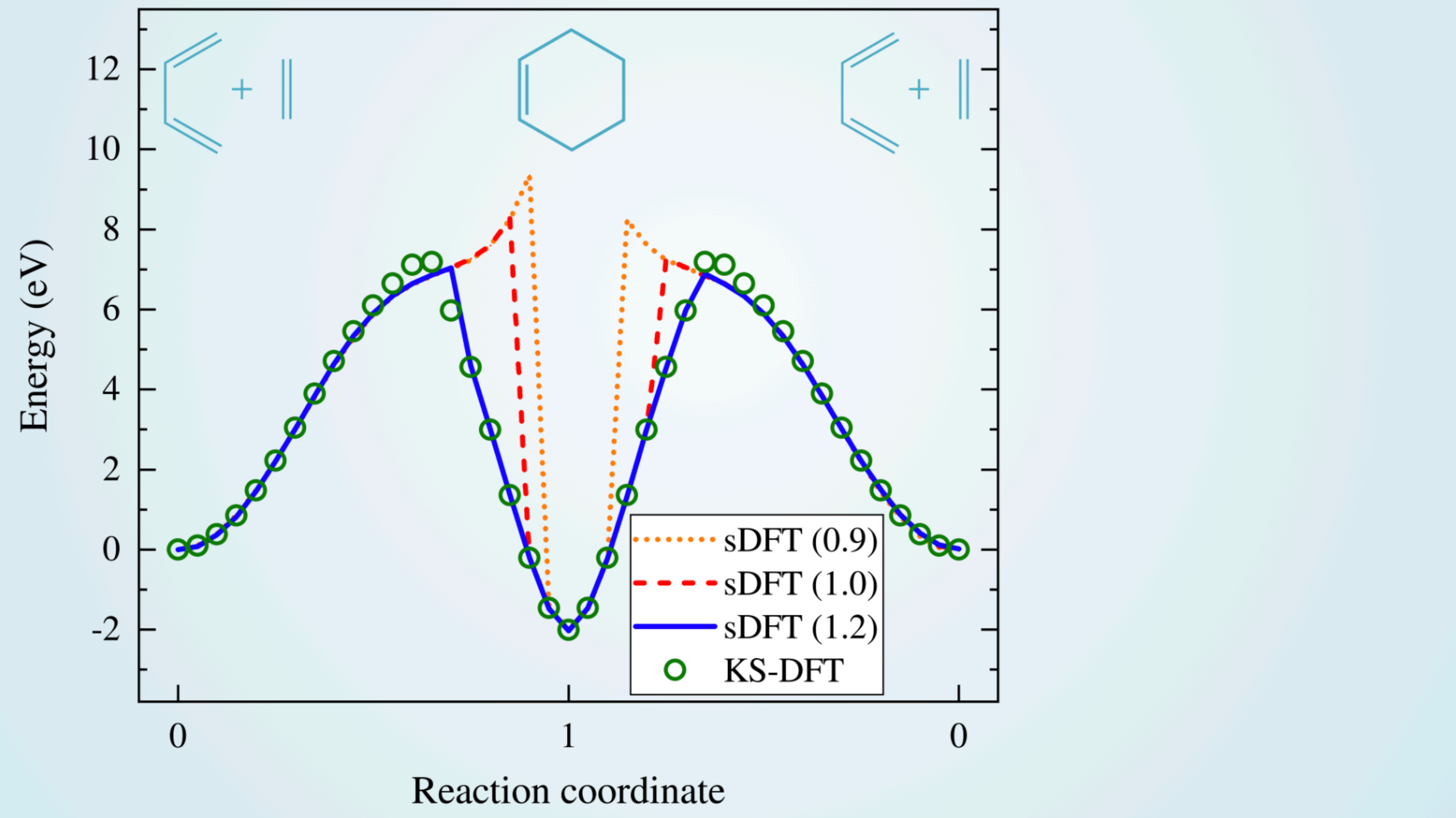
Structure and dynamics recovered!

100% Reset

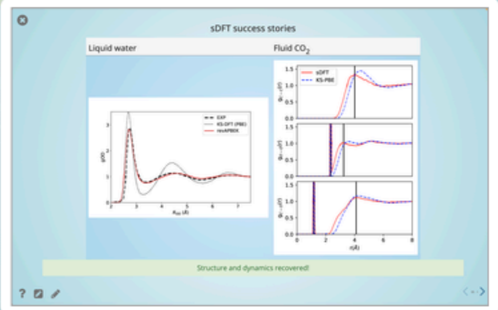
sDFT is not perfect either...

The nonadditive kinetic energy, $T_s^{nadd}[n]$, is not accurate enough when...

- ...there are covalent bonds between subsystems
- ...subsystems are too close



Adaptive sDFT has only 1 parameter (rcut) related to the covalent radii.



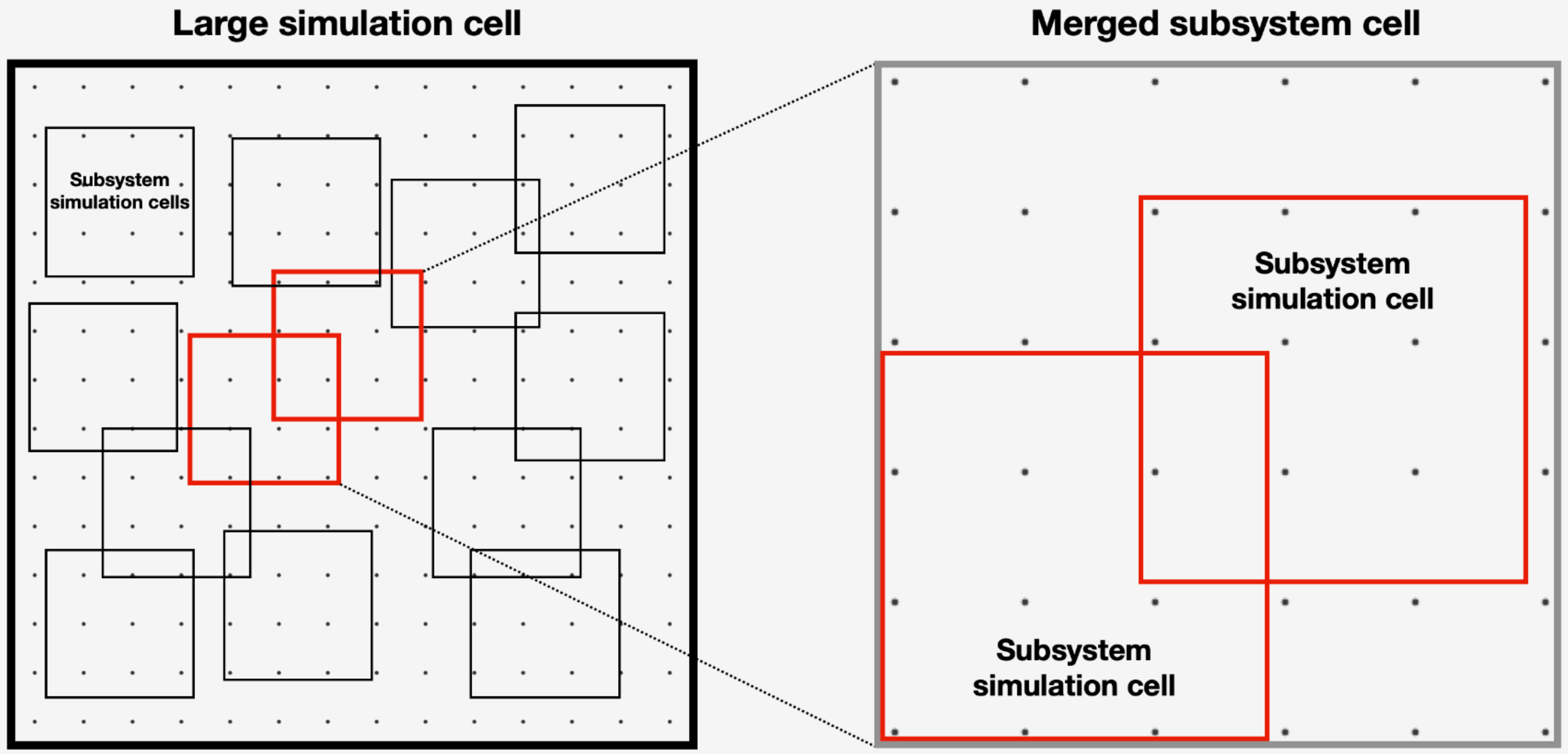
Adaptive sDFT

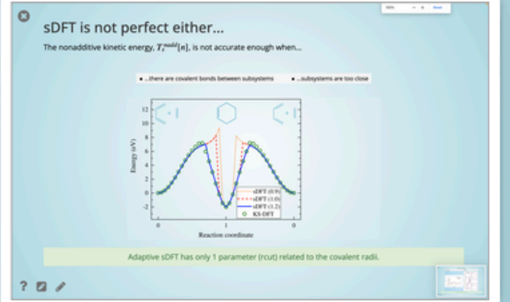
To do:

- Must be able to "split" and "merge" subsystems along a dynamics
- Do so *automatically*

We did it... but it was challenging

- Adaptive computational settings
- Adaptive MPI communicators
- Adaptive data allocation



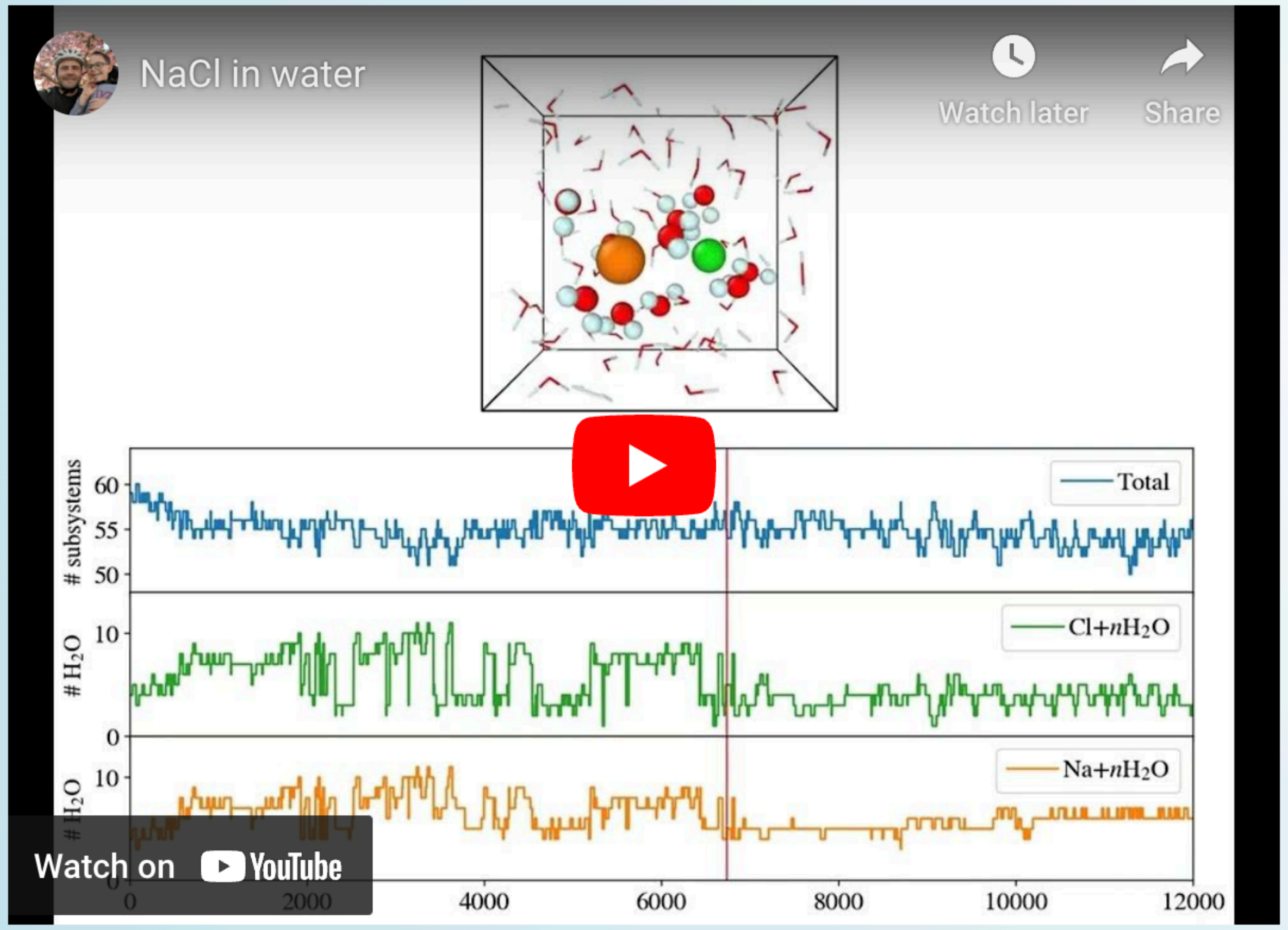


Adaptive sDFT: dissociation of NaCl in water

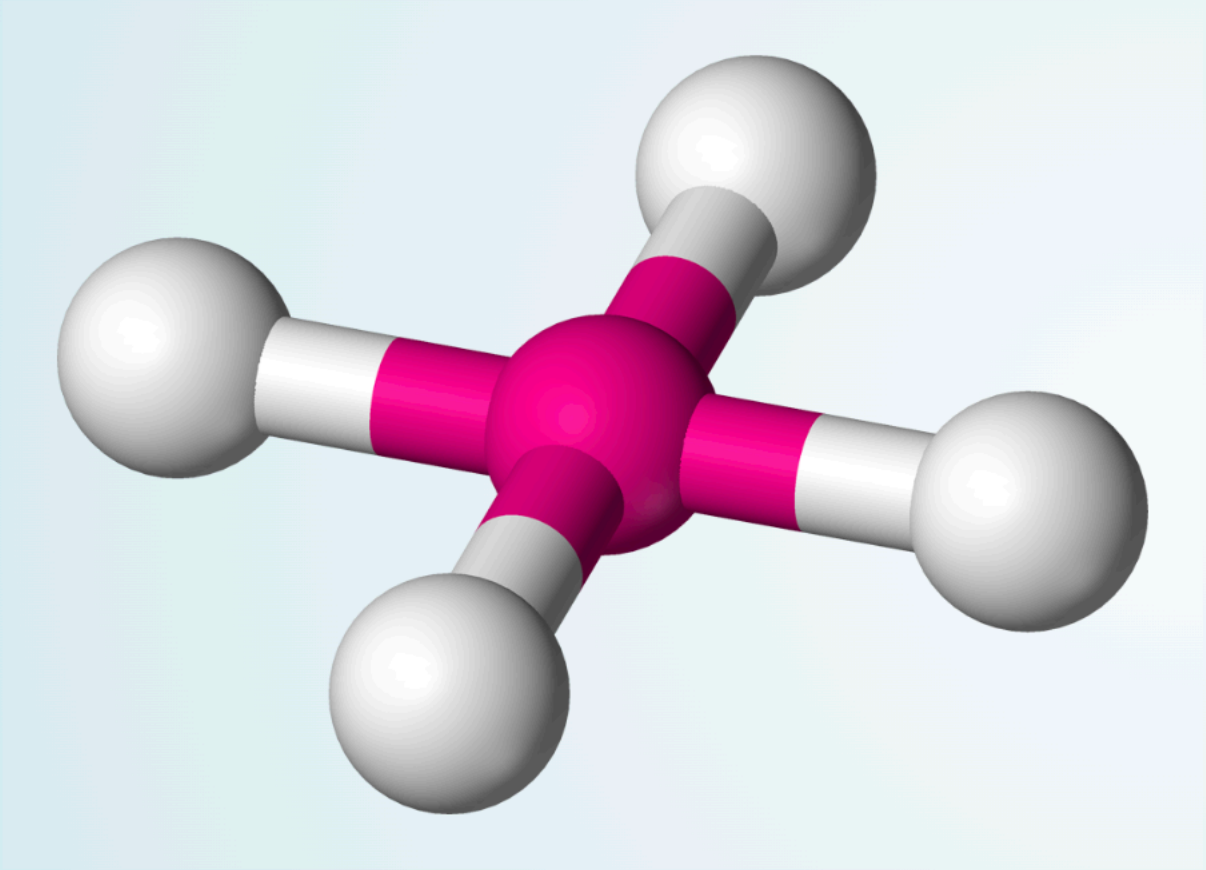
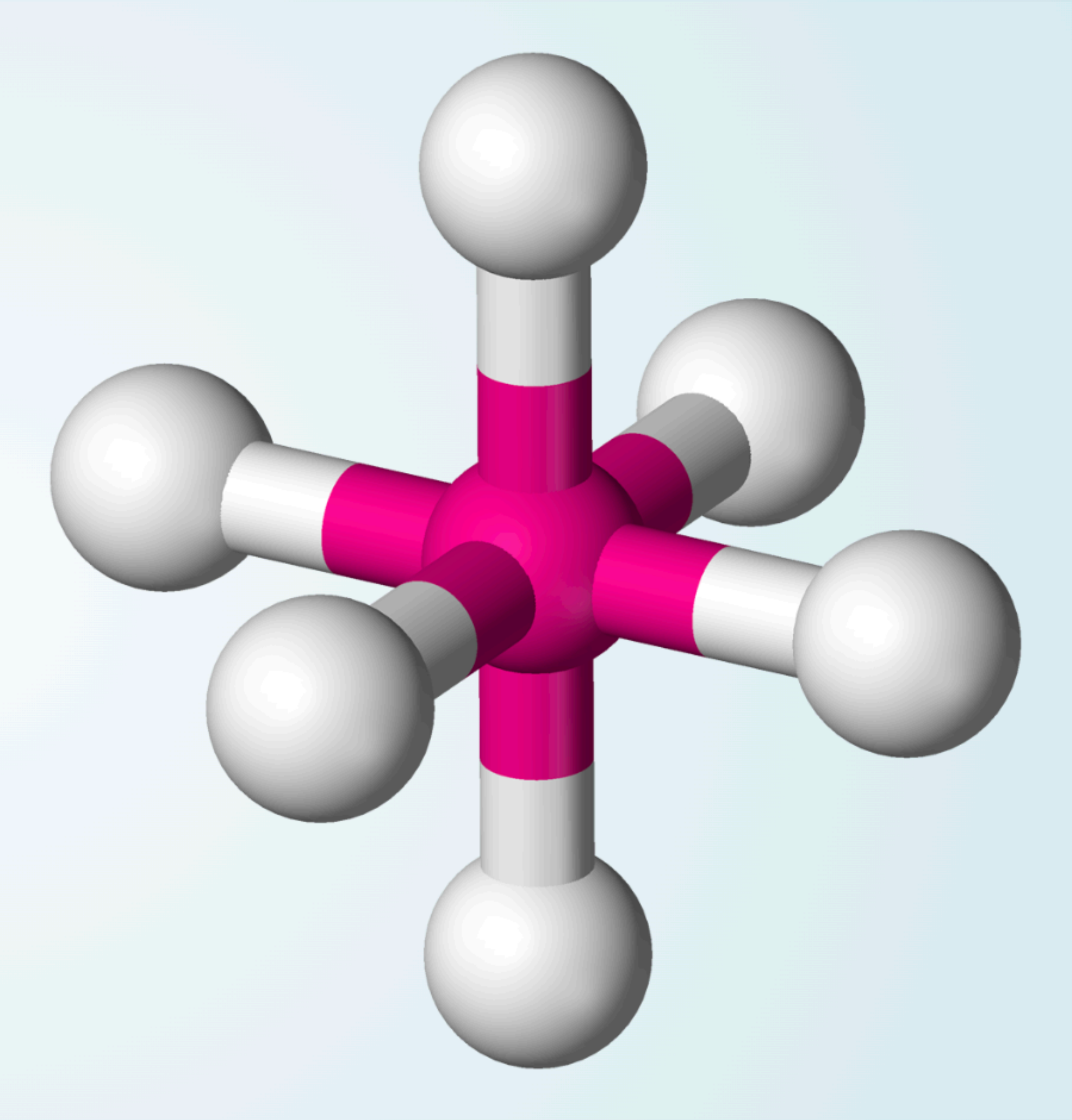
https://www.youtube.com/watch?v=-lmdPHzJfeU

```
In [22]: YouTubeVideo('-lmdPHzJfeU', width=600, height=450)
```

Out[22]:



Adaptive sDFT: hydration of Ni^{2+}, Pd^{2+} and Pt^{2+}

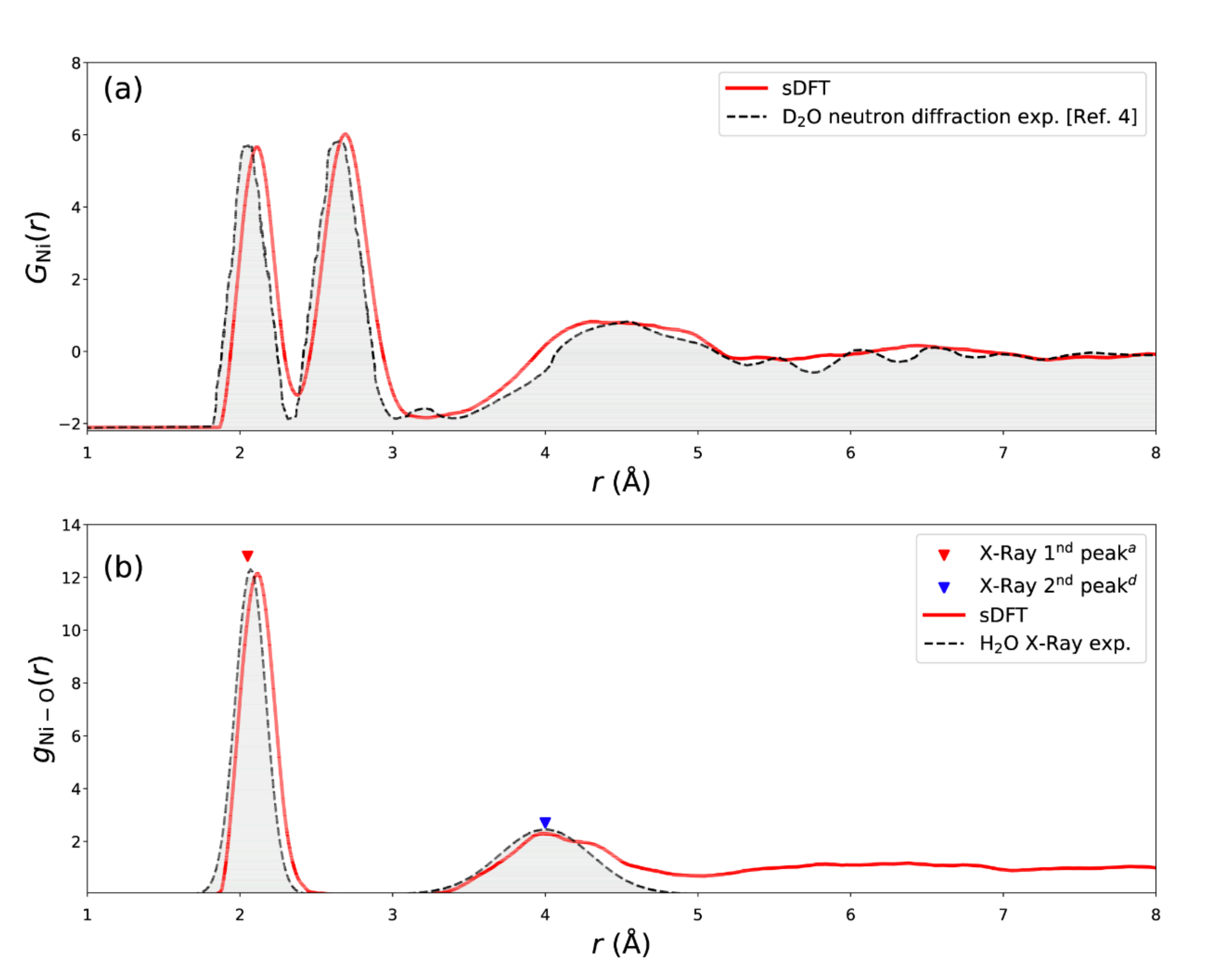
Square Planar	Octahedral
	

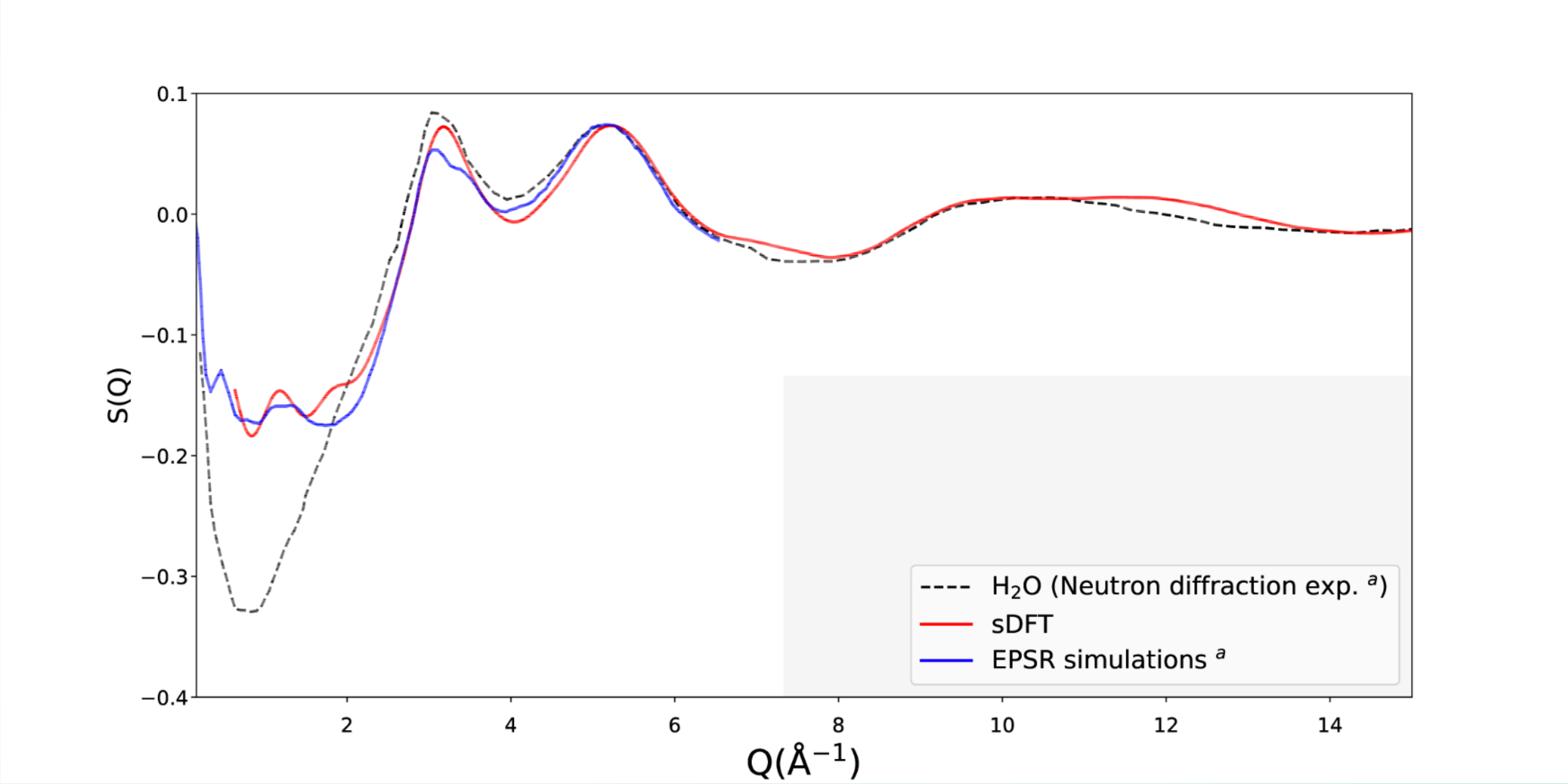
Stay tuned for an upcoming paper!

Comparison with experiments for Ni and Pd

Ni(II) -- Total $G_{Ni}(r)$ and partial $G_{Ni-O}(r)$

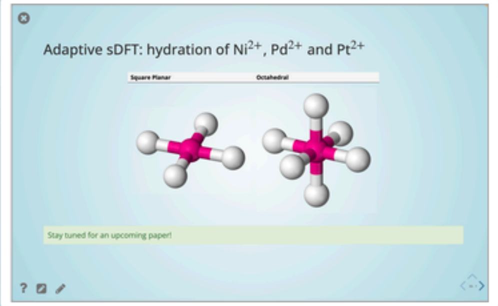
Pd(II) -- Total structure factor $S(Q)$





- Almost quantitative agreement with experiments
- XRay: Powell et al 1989 J. Phys.: Condens. Matter 1 8721;
- N scattering: Nelson et al J. Phys. C 11, 15 L625 (1978)

- Better agreement with the experiment compared with the EPSR result of Bowron *et al.*


Adaptive sDFT: hydration of Ni^{2+}, Pd^{2+} and Pt^{2+}

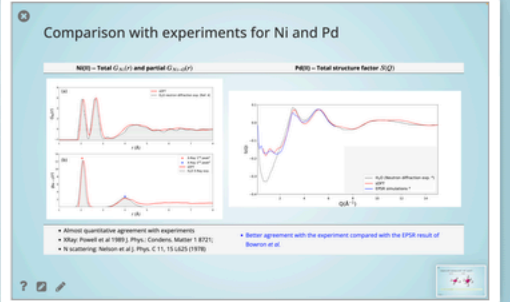
Despite the speaker, you learned:

Good things:

- OF-DFT is very fast and linear scaling with system size
- OF-DFT is useful for metals and semiconductors of any shape and size
- Where OF-DFT fails we can use Subsystem DFT. Still linear scaling, and applicable to any inter-subsystem interaction strength
- Therefore, the applicability of KS-DFT is extended to larger, more meaningful systems

Bad things:

- OF-DFT has limited applicability because of the "nonlocal pseudopotential problem"
- Subsystem DFT is fast but only if subsystems are small



Comparison with experiments for Ni and Pd