Computing the properties of all known inorganic crystals

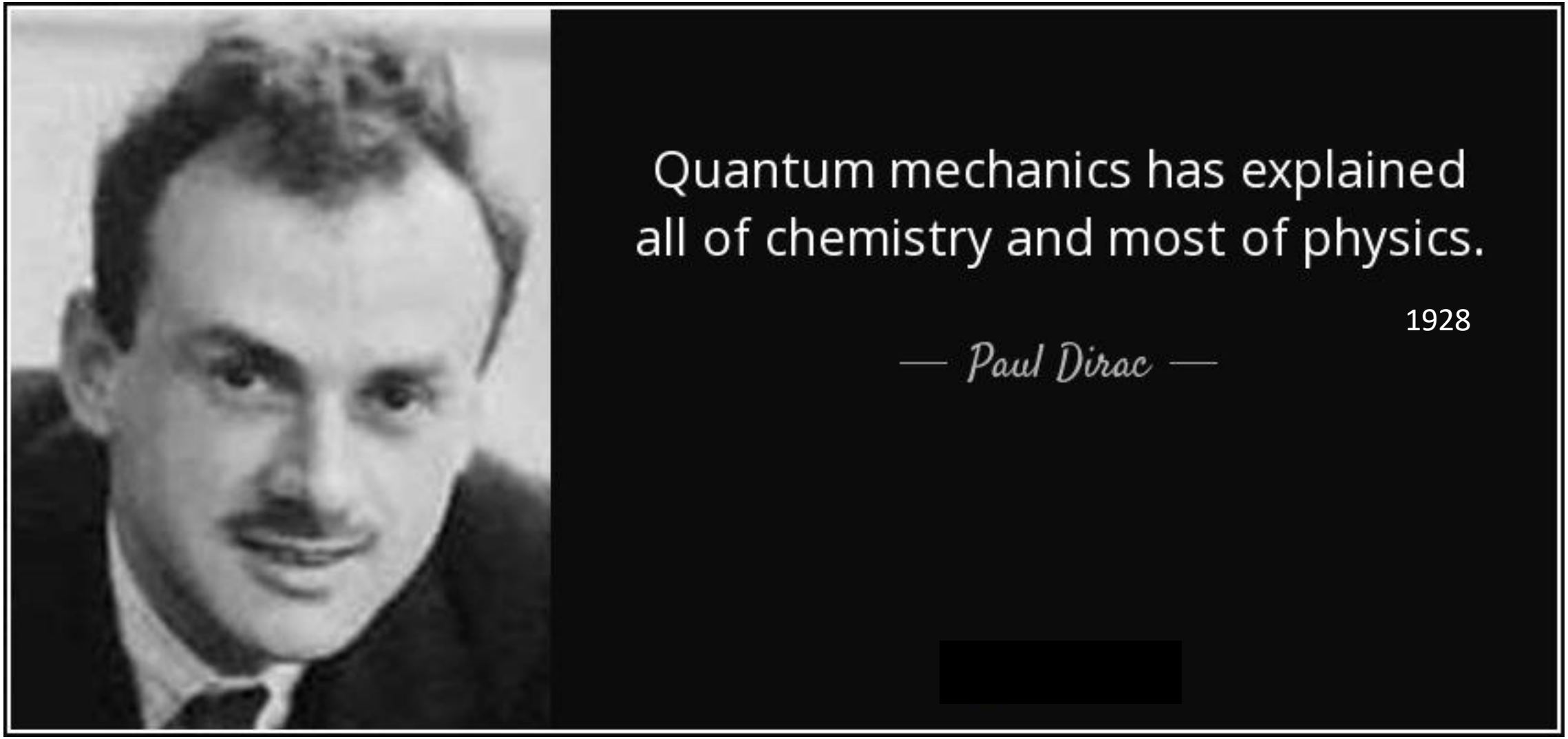
Kristian S. Thygesen

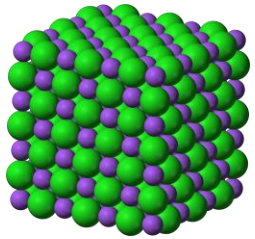
Technical University of Denmark

thygesen@fysik.dtu.dk

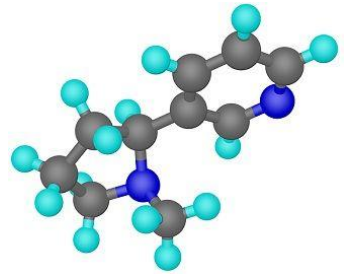
DTU

Deic conference 2023


Quantum mechanics has explained all of chemistry and most of physics.
1928
— Paul Dirac —



Electronic structure calculations



What?

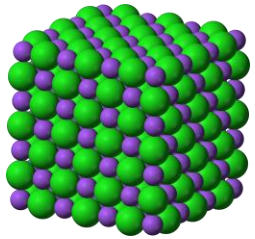
Solve Schrödinger's equation for the motion of electrons in a solid/molecule.

$$\left[\frac{\hbar^2}{2m}\nabla^2 + V(\{\boldsymbol{R}_i\};\boldsymbol{x})\right]\Psi_n(\boldsymbol{x}) = E_n\Psi_n(\boldsymbol{x})$$

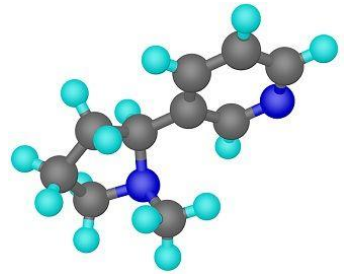
Why?

The electrons determine everything

- Chemistry (bonds, chemical reactions, ...)
- Thermodynamics (heat capacity, phase diagrams, ...)
- Structure and ion dynamics (phonons, heat transport, ...)
- Physical properties (magnetism, conductivity, ...)



Electronic structure calculations



What?

Solve Schrödinger's equation for the motion of electrons in a solid/molecule.

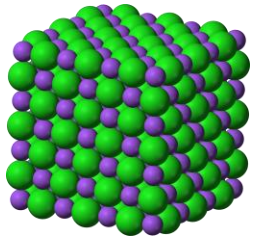
$$\left[\frac{\hbar^2}{2m}\nabla^2 + V(\{\boldsymbol{R}_i\};\boldsymbol{x})\right]\Psi_n(\boldsymbol{x}) = E_n\Psi_n(\boldsymbol{x})$$

No parameters.
Only laws of nature and fundamental constants.
Ab initio / first-principles

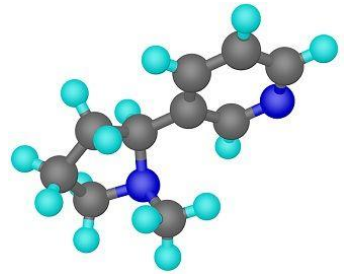
Why?

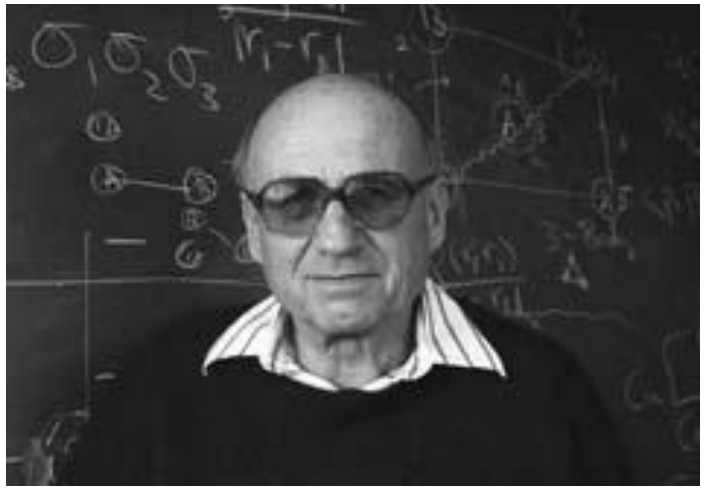
The electrons determine everything

- Chemistry (bonds, chemical reactions, ...)
- Thermodynamics (heat capacity, phase diagrams, ...)
- Structure and ion dynamics (phonons, heat transport, ...)
- Physical properties (magnetism, conductivity, ...)



Density functional theory

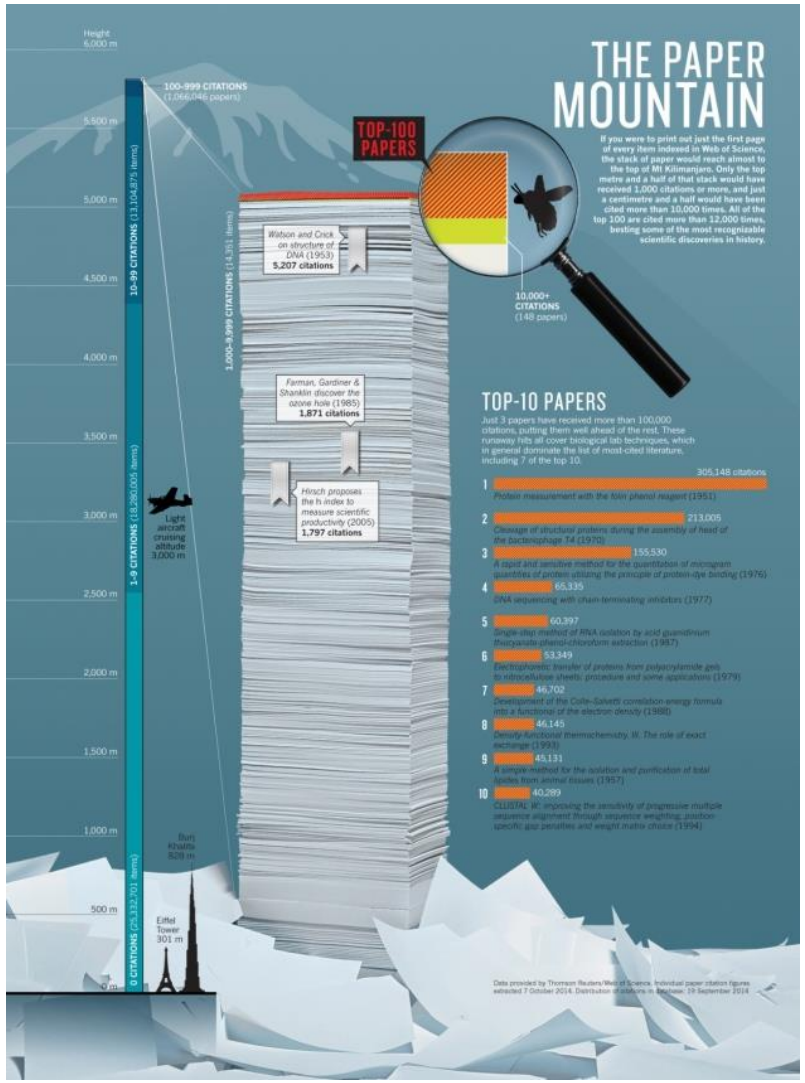




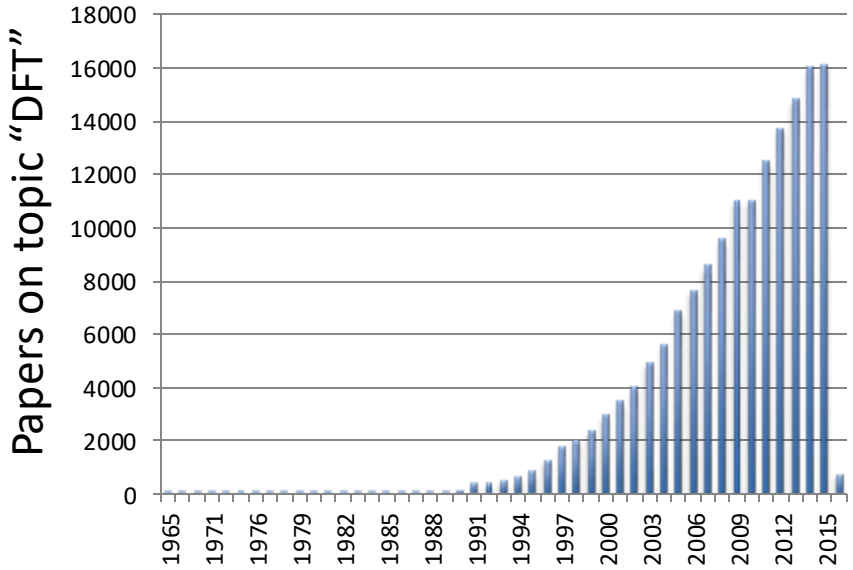
- DFT: The electron density $n(\mathbf{r})$ is the central variable (always only 3 degrees of freedom)
- The ground state energy and any other property is determined uniquely from $n(\mathbf{r})$

Walter Kohn
1998 Nobel Prize in Chemistry
"For his development of the density-functional theory"
(Papers made in 1964-65)

Importance of Density Functional Theory



In 2014 Nature identified the 100 most cited papers of all times



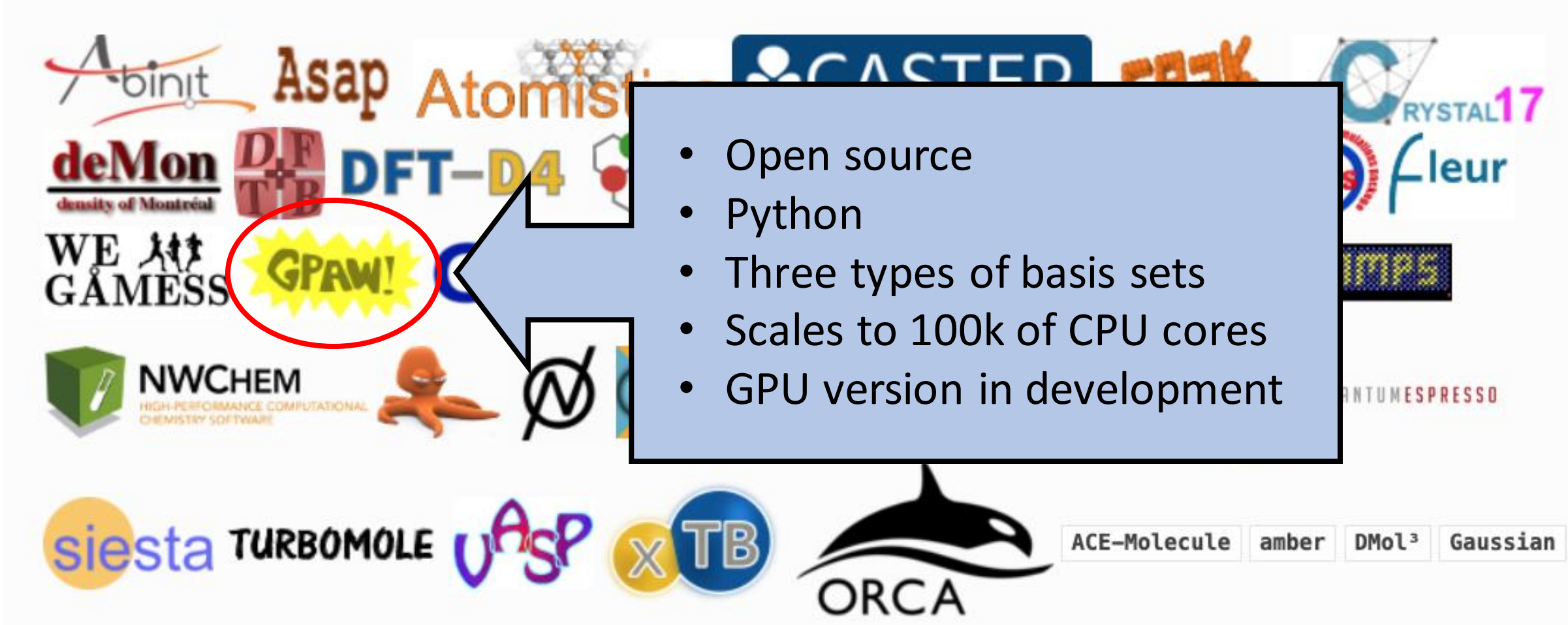
Density functional theory

[...]

Most of this software is built on density functional theory (DFT), easily the most heavily cited concept in the physical sciences. Twelve papers on the top-100 list relate to it, including 2 of the top 10. At its heart, DFT is an approximation that makes impossible

An eco-system of DFT codes

An eco-system of DFT codes



- Open source
- Python
- Three types of basis sets
- Scales to 100k of CPU cores
- GPU version in development

Materials in our society

Science

Materials

technology

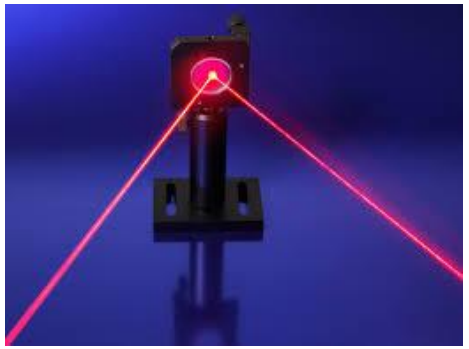


Photo-catalytic water splitting

$$2\,h\nu + H_2O_{(liq)} \rightarrow \tfrac{1}{2}\,O_{2(gas)} + H_{2(gas)}$$

Minimum energy required = 1.23 eV

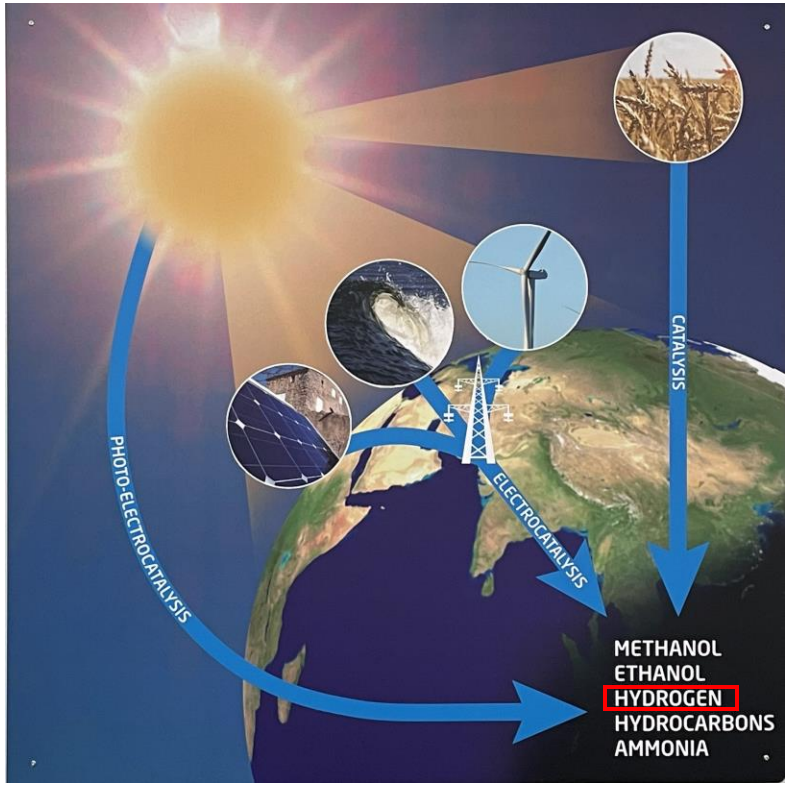
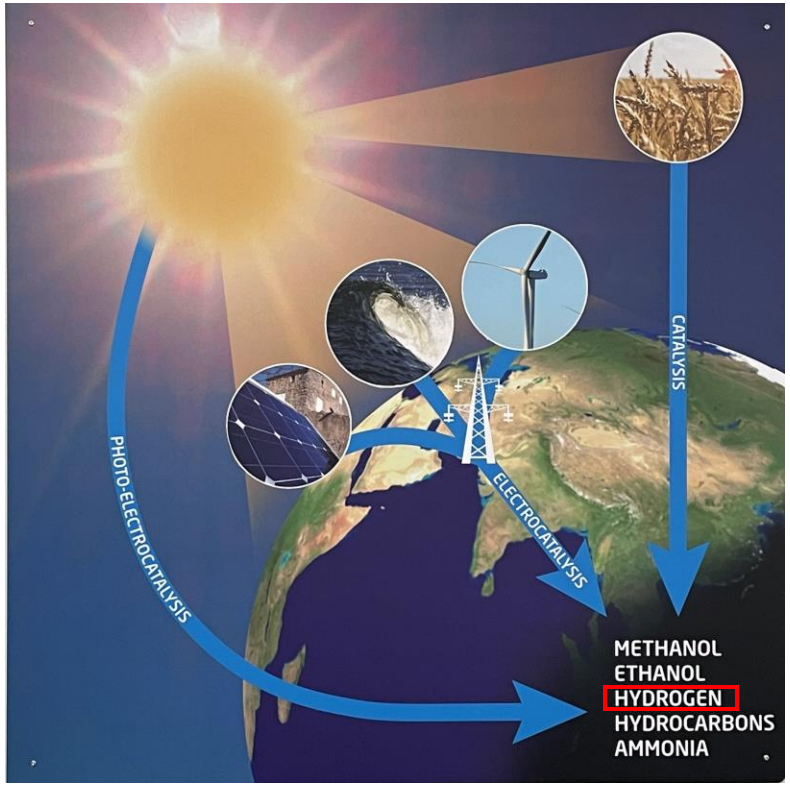


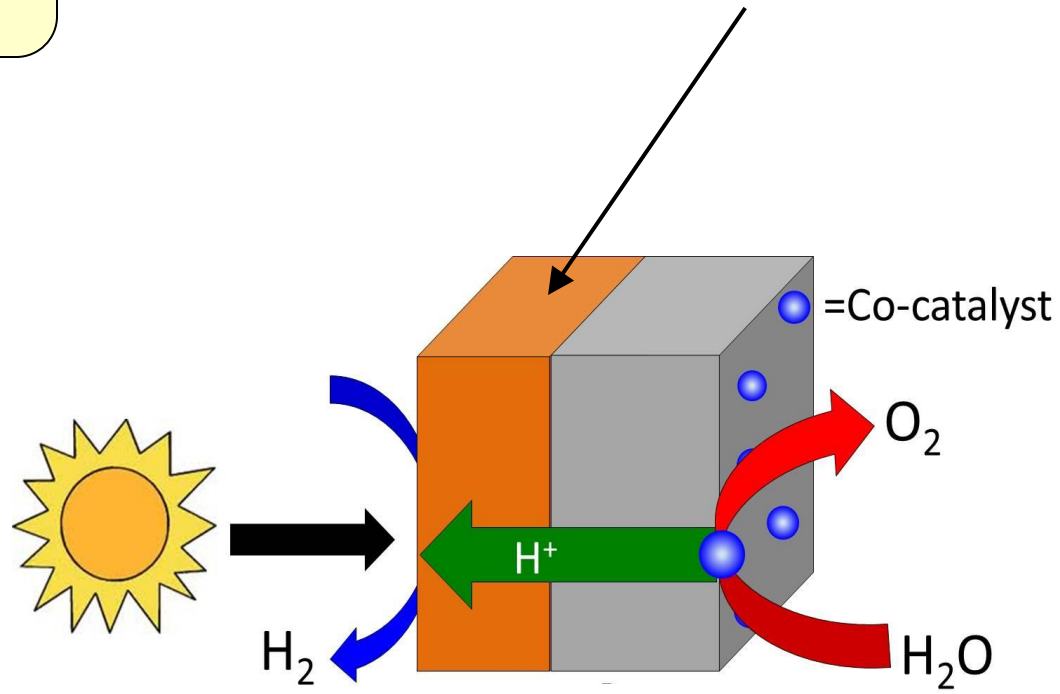
Photo-catalytic water splitting

$2\,h\nu + H_2O_{(liq)} \rightarrow \frac{1}{2}\,O_{2(gas)} + H_{2(gas)}$

Minimum energy required = 1.23 eV



Efficient water splitting requires an absorber material with a band gap ~2 eV



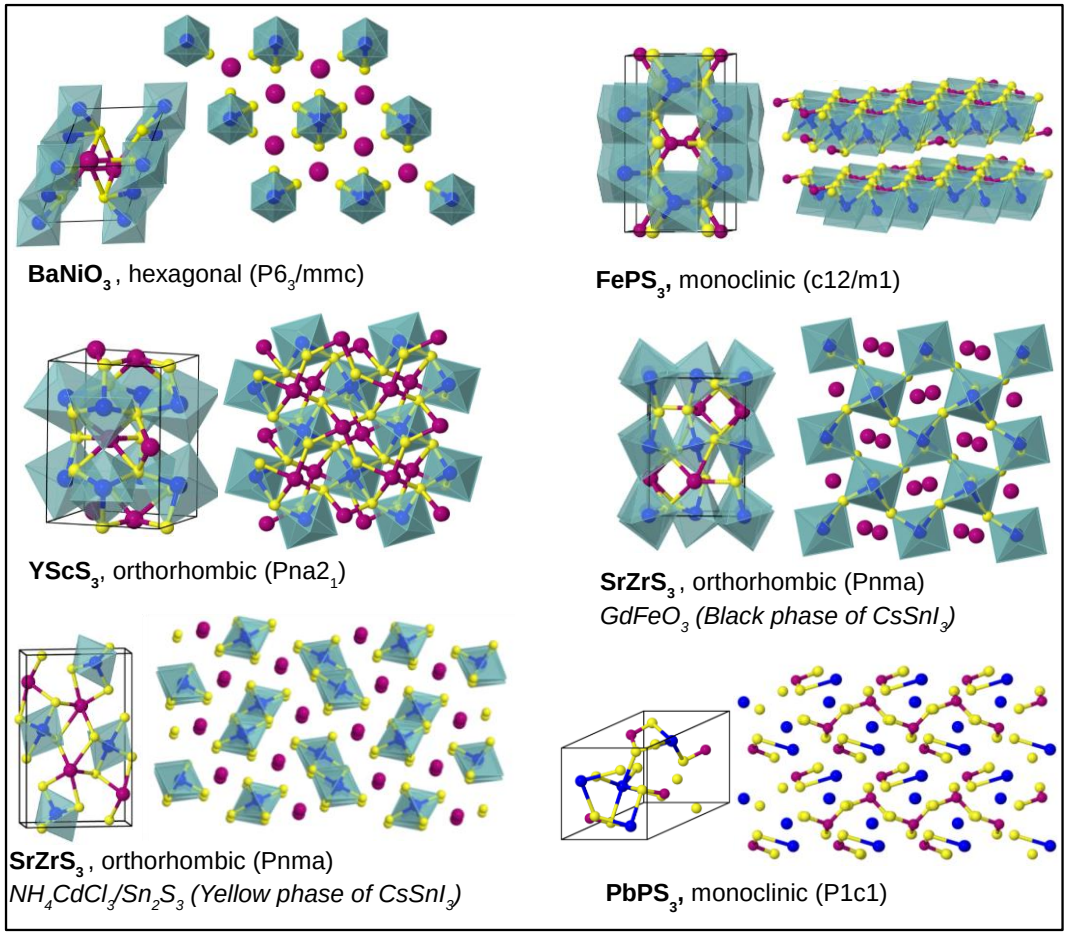
ABS_3 sulphide perovskites

2916 AB cation combinations

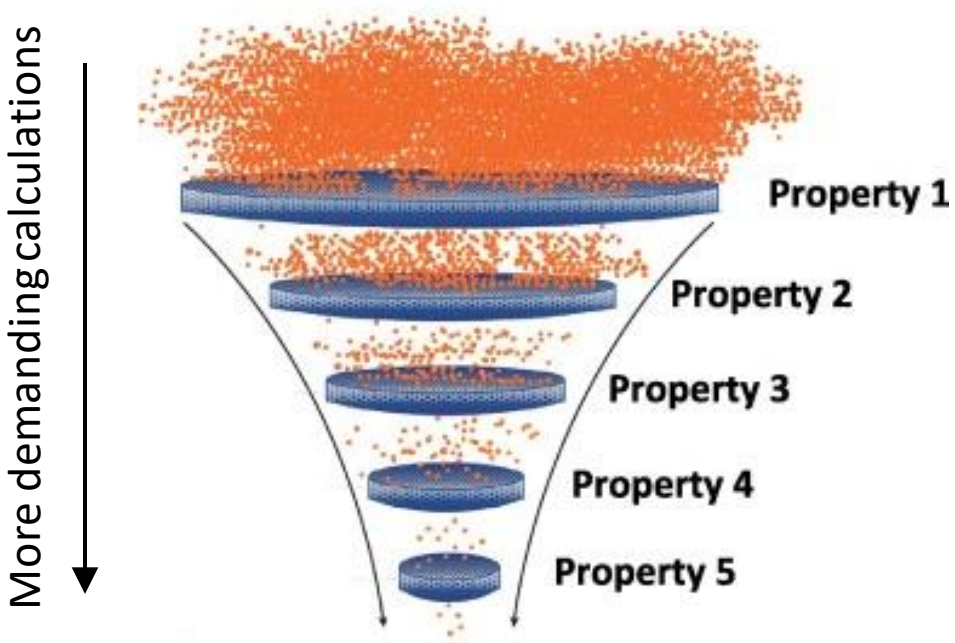
1 H																	2 He
3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne
11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
55 Cs	56 Ba *	71 Lu	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn

17.496 materials

Six most common ABS_3 crystal structures



Computational screening

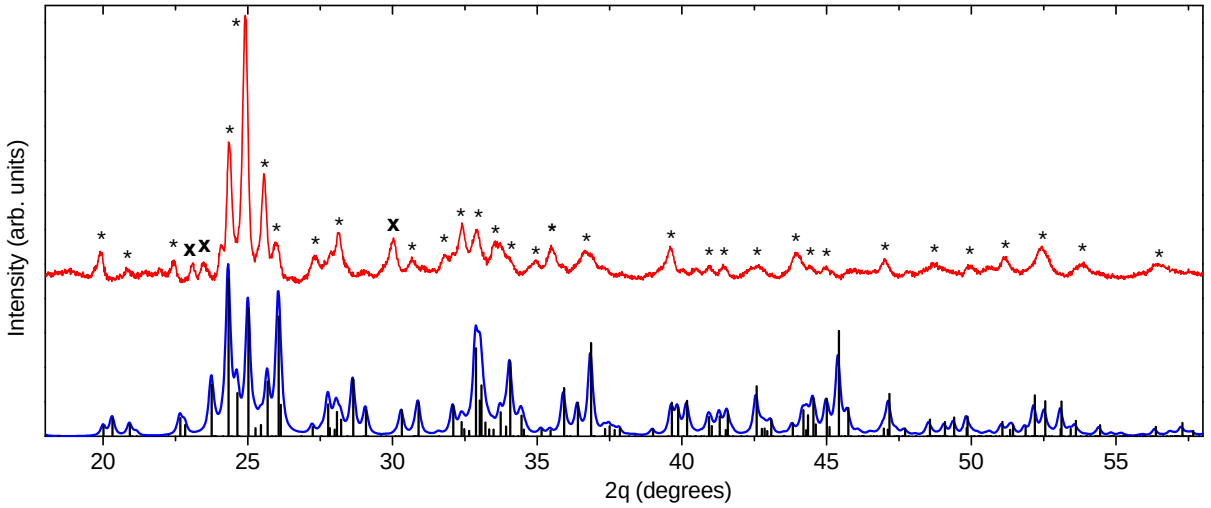


Best candidate materials

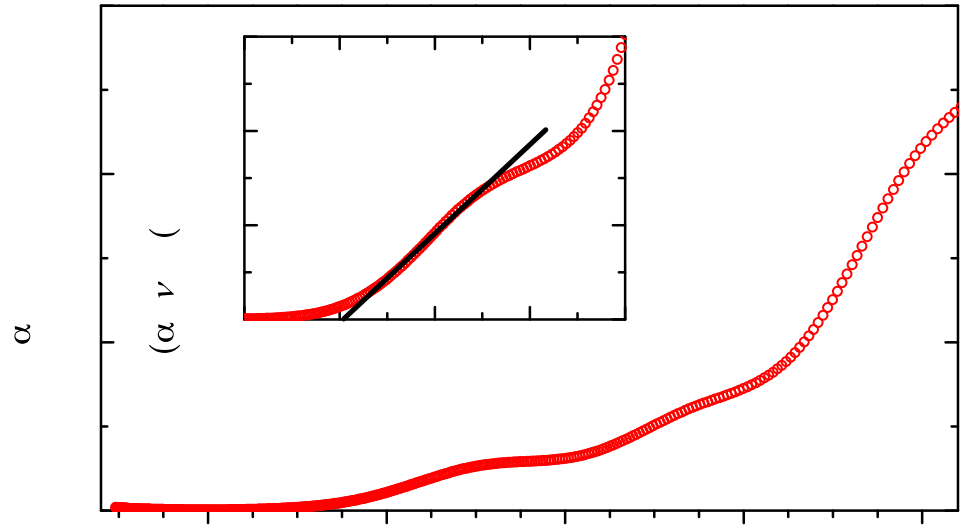
formula	$E_g^{GLLB-SC}$	$E_{g(direct)}^{GLLB-SC}$	E_g^{HSE06}	m^*_h	m^*_e	prototype
$AlLaS_3$	1.67	1.67	1.47	-0.337	0.489	$SrZrS_3$(Y)
$BaHfS_3$	2.32	2.32	2.20	-0.255	0.414	$SrZrS_3$(B)
$BaZrS_3$	2.25	2.25	2.08	-0.749	0.426	$SrZrS_3$(B)
$BiLiS_3$	1.13	1.43	1.08	-0.209	0.455	$FePS_3$
$BiScS_3$	2.45	2.64	2.62	-0.318	0.520	$SrZrS_3$(Y)
$BiTlS_3$	1.36	1.98	1.30	-0.636	0.309	$FePS_3$
$HfGeS_3$	1.70	1.73	1.68	-0.568	0.256	$SrZrS_3$(Y)
$HfPbS_3$	2.11	2.24	1.96	-0.396	0.538	$SrZrS_3$(Y)
$HfSnS_3$	1.53	1.57	1.53	-0.408	0.270	$SrZrS_3$(Y)
$HfZnS_3$	2.03	2.47	1.98	-0.173	0.431	$FePS_3$
$LaSbS_3$	1.23	1.23	0.99	-0.439	0.167	$SrZrS_3$(Y)
$MgZrS_3$	2.21	2.32	2.06	-0.718	0.779	distorted
$PbZrS_3$	1.68	1.91	1.66	-0.434	0.525	$SrZrS_3$(B)
$ScSbS_3$	2.35	2.43	1.99	-0.502	0.258	$SrZrS_3$(Y)
$SnZrS_3$	1.76	1.98	1.56	-0.488	0.802	$PbPS_3$
$SrZrS_3$	2.49	2.49	2.30	-0.768	0.496	$SrZrS_3$(B)
$TaLiS_3$	1.98	2.00	2.06	-0.755	0.985	$FePS_3$
$TlScS_3$	1.60	1.76	1.62	-0.377	0.685	$YScS_3$
$YBiS_3$	2.17	2.24	2.04	-0.428	0.488	$SrZrS_3$(Y)
$YLaS_3$	1.87	1.87	1.57	-0.509	0.438	$SrZrS_3$(Y)
$ZrBaS_3$	1.69	1.96	1.62	-0.453	0.279	distorted
$ZrBaS_3$	1.79	1.79	1.54	-0.402	0.413	$SrZrS_3$(Y)
$ZrZnS_3$	1.91	1.97	1.87	-0.616	0.427	$FePS_3$

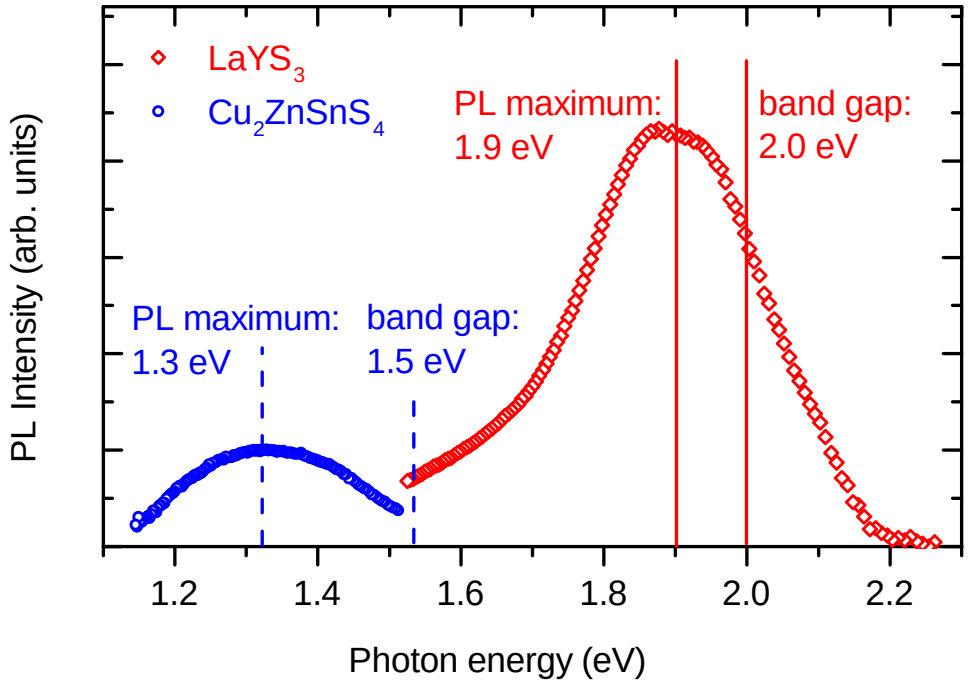
Kuhar, Thygesen, Jacobsen *et al.* Energy Env. Sci. **10**, 2579 (2018)

Synthesis of $LaYS_3$



- XRD confirms the structure
- Optical absorption confirms the band gap of 2.0 eV
- PL confirms direct gap and absence of defects

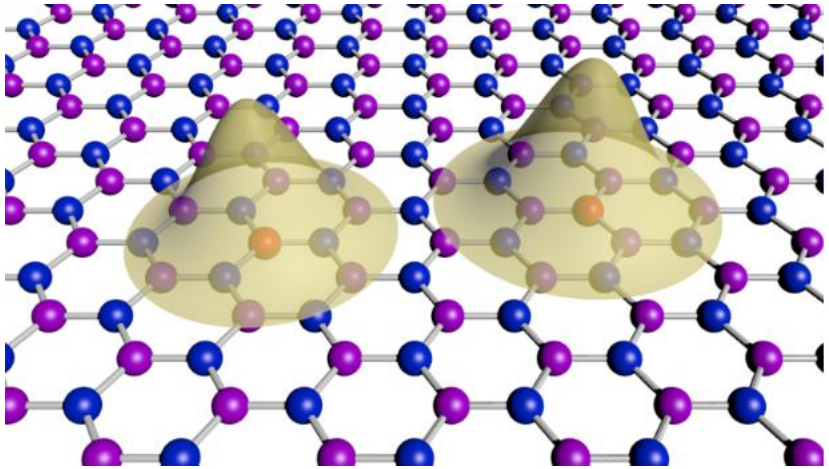




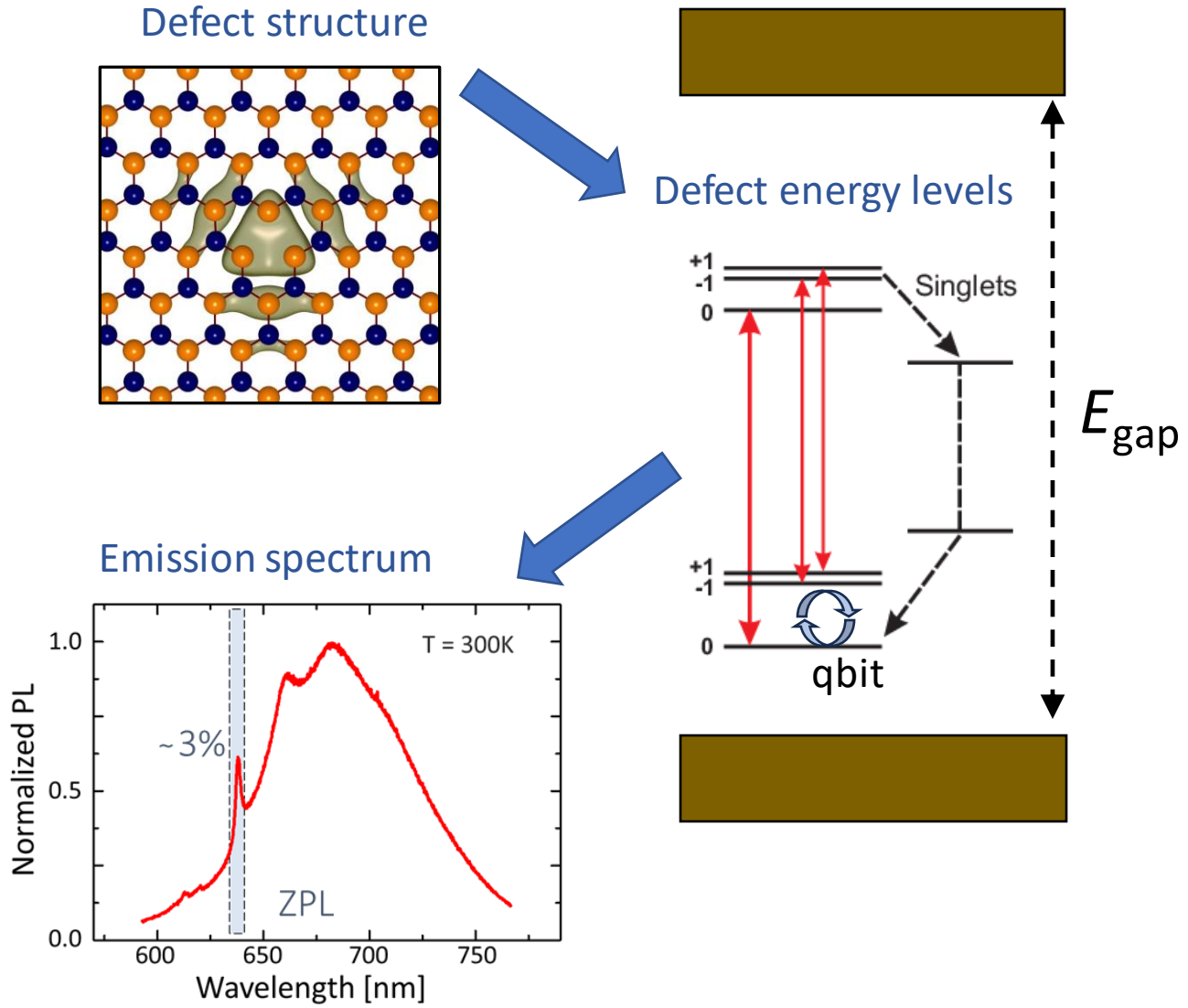
Crystal defects for quantum technology

Point defects in solid state systems have potential applications as

- Single photon sources
- Spin qubits
- Magnetic field sensors

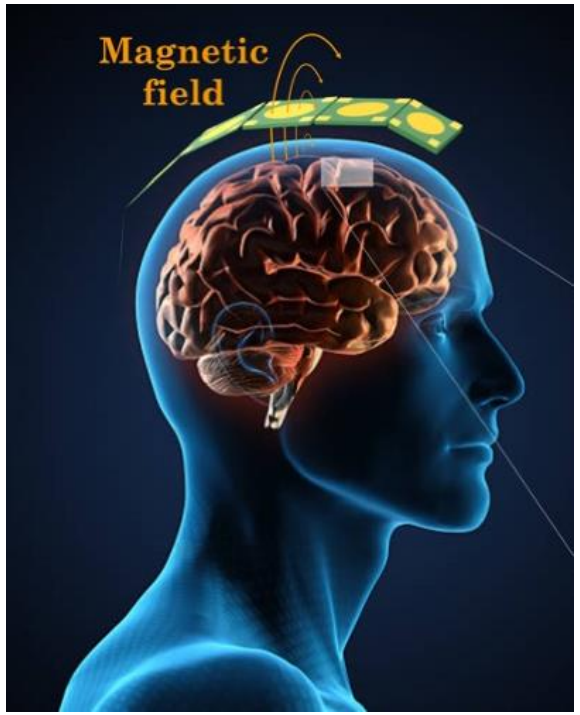


Spin qbits in a 2D material

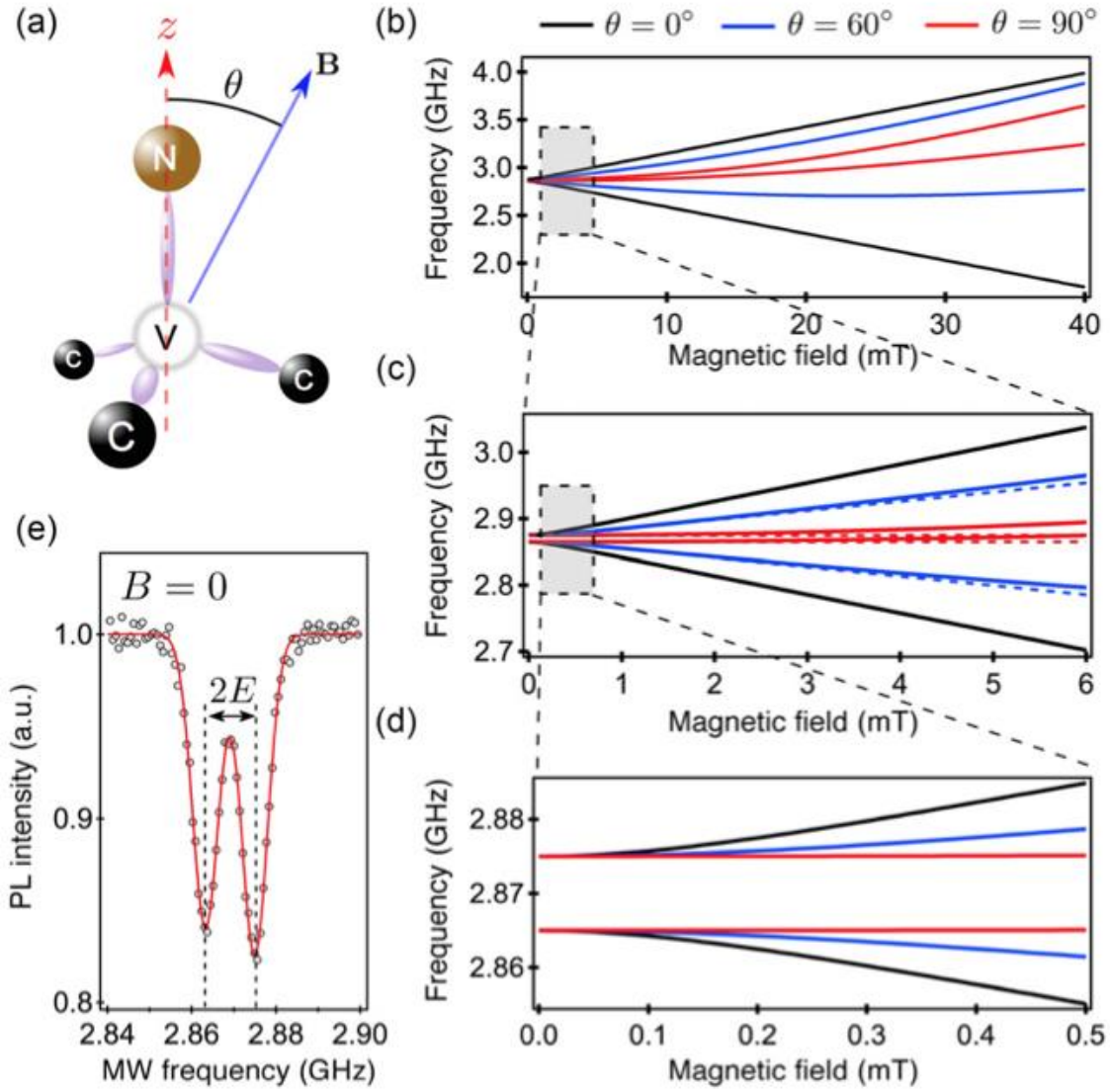


Magnetic field sensing

The energy of the spin states are sensitive to magnetic fields.

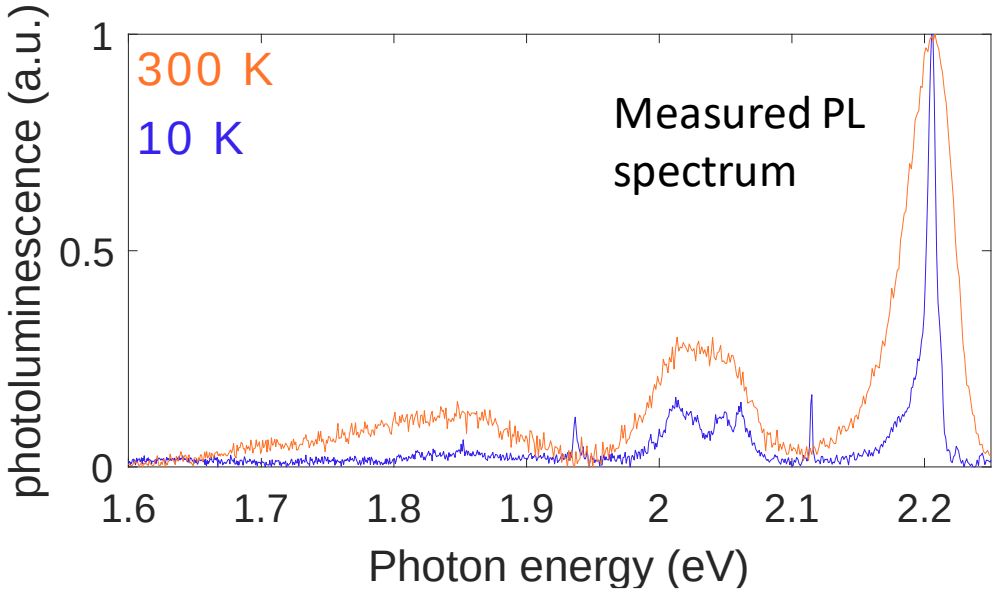


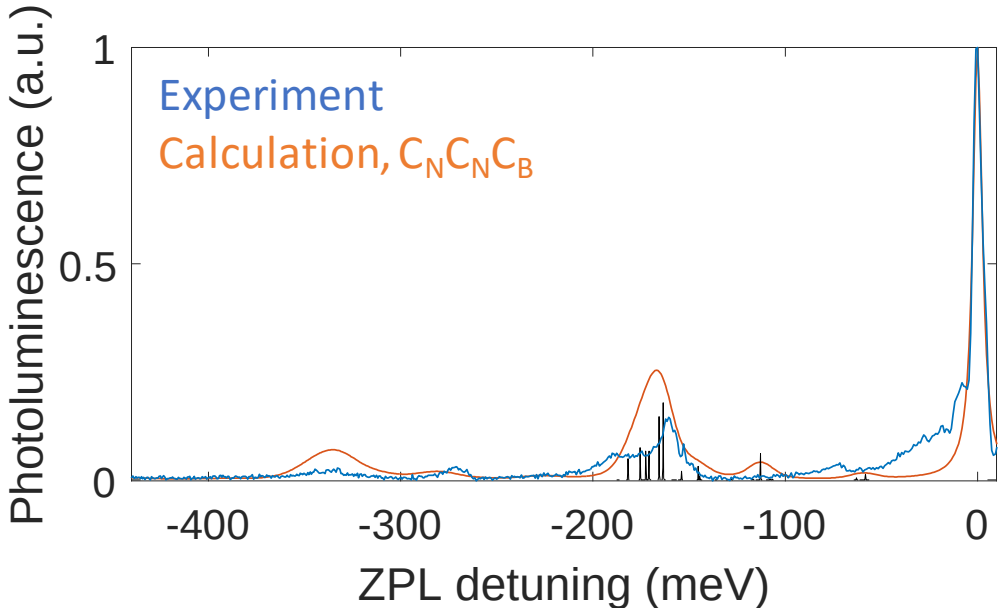
BIOMAG, an NNF Challenge Project

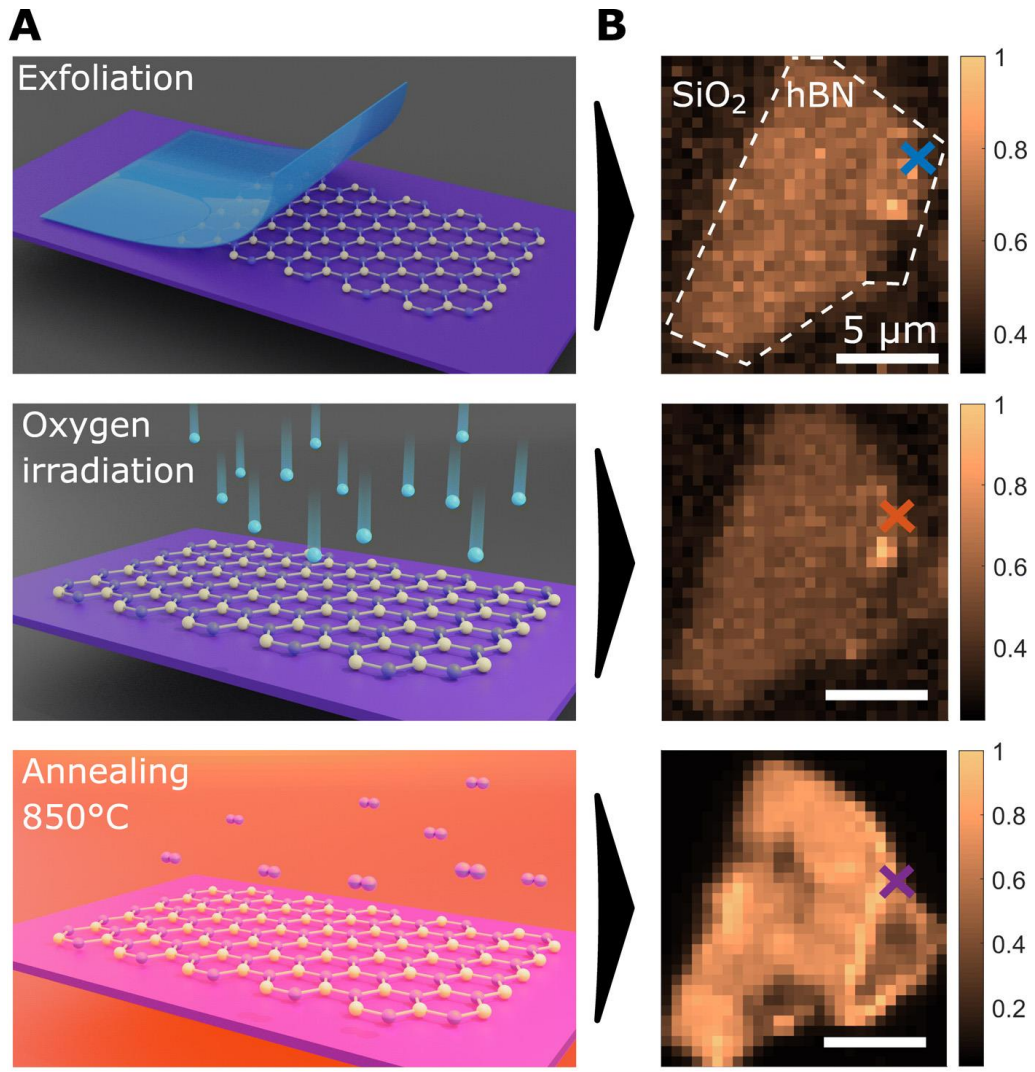


King et al. Nature Com. **6**, 8965 (2016)

Luminescent centers in hexagonal boron-nitride

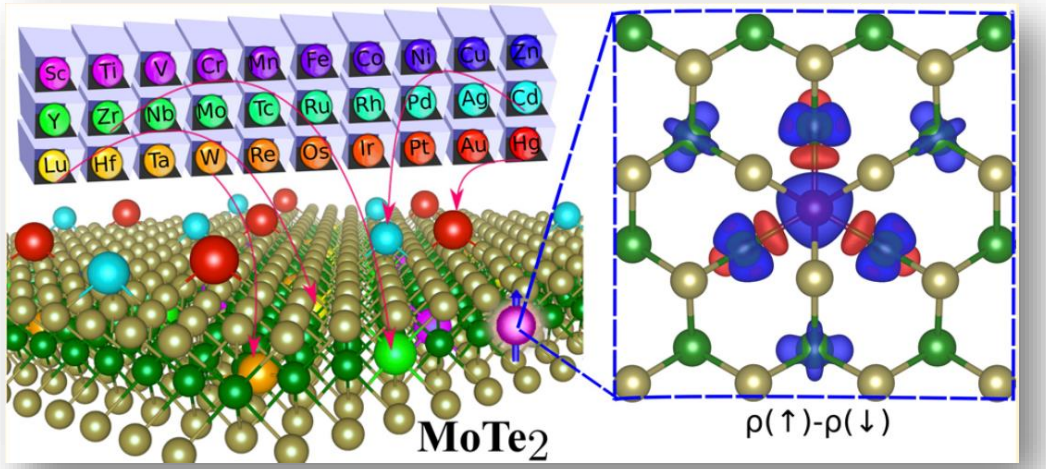




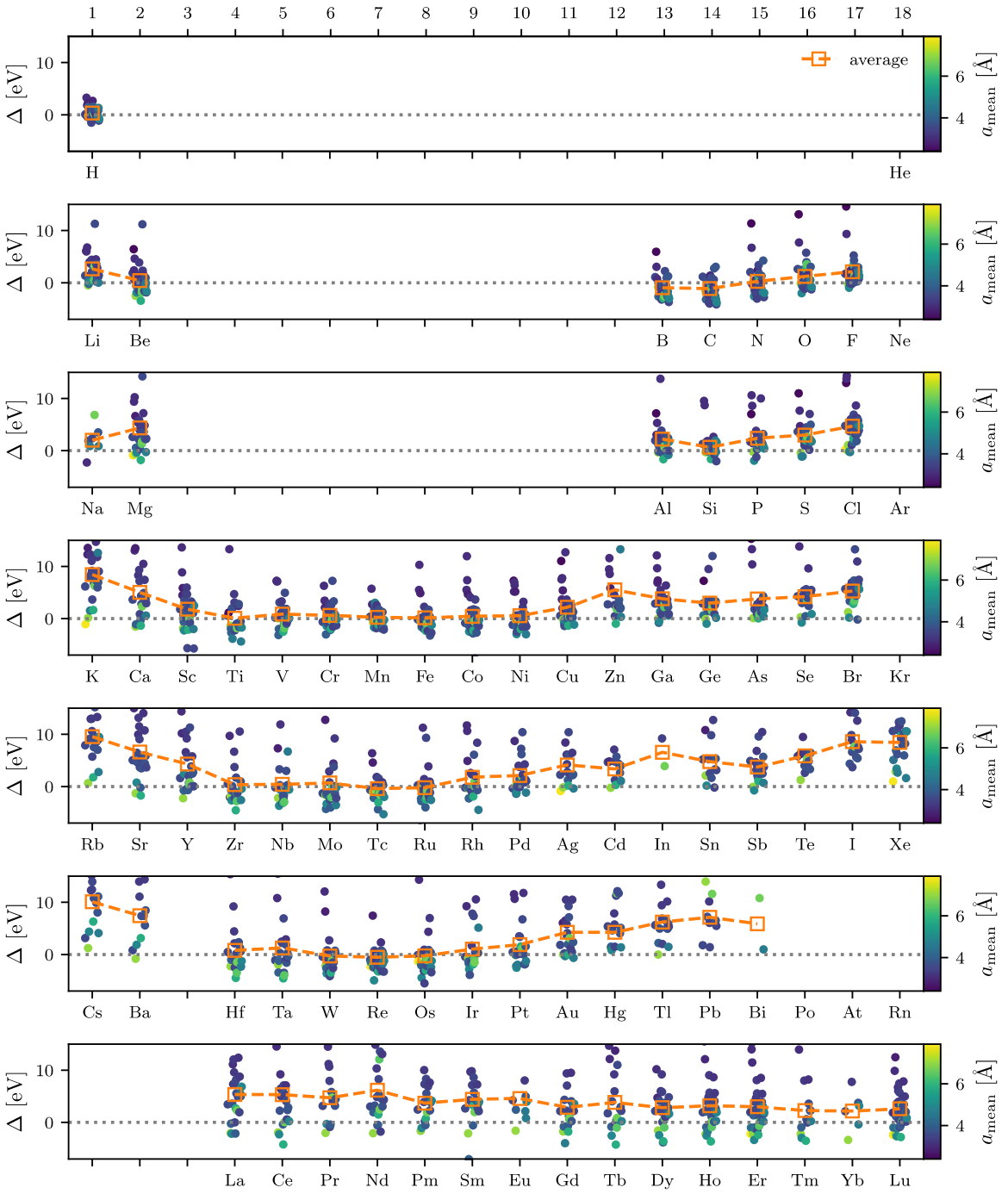


M. Fischer *et al.* Science Adv. 7, 8 (2021)
M. Fischer *et al.* arXiv: 2209.08910

Introducing atomic impurities in 2D crystals

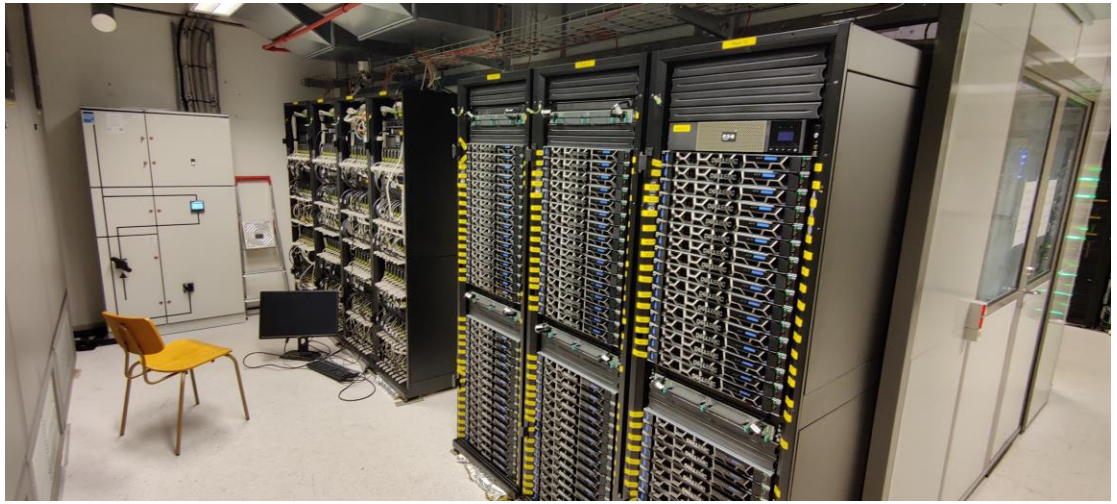


Which impurity atoms can get incorporated into the material and which will adsorb on the surface?
(53 materials x 65 impurities)



J. Davidsson *et al.* npj 2D Materials and Applications **7**, 26 (2023)

The DTU *Niflheim* supercomputer



- CPU cores: 24.000
- Peak performance: 1.8 PetaFlops
- Capacity/1 year: 215 mio core hrs
- Investment: DKK 40M over 8 years
- Manpower: 1 person (Ole H. Nielsen)

Danmarks Grundforskningsfond
Danish National Research Foundation

erc European Research Council

VILLUM FONDEN

novo nordisk fonden

LUMI Grand Challenge

- **When**: 6 hours on March 22, 2023
- **Allocation** : 60.000 GPU hrs
- **Team**: Mikael Kuisma, Jens Jørgen Mortensen, Ask Hjorth Larsen, Tara Boland
- **Challenge**: Calculate the electronic structure of all known inorganic crystals

LUMI Grand Challenge

- **When**: 6 hours on March 22, 2023
- **Allocation** : 60.000 GPU hrs
- **Team**: Mikael Kuisma, Jens Jørgen Mortensen, Ask Hjorth Larsen, Tara Boland
- **Challenge**: Calculate the electronic structure of all known inorganic crystals

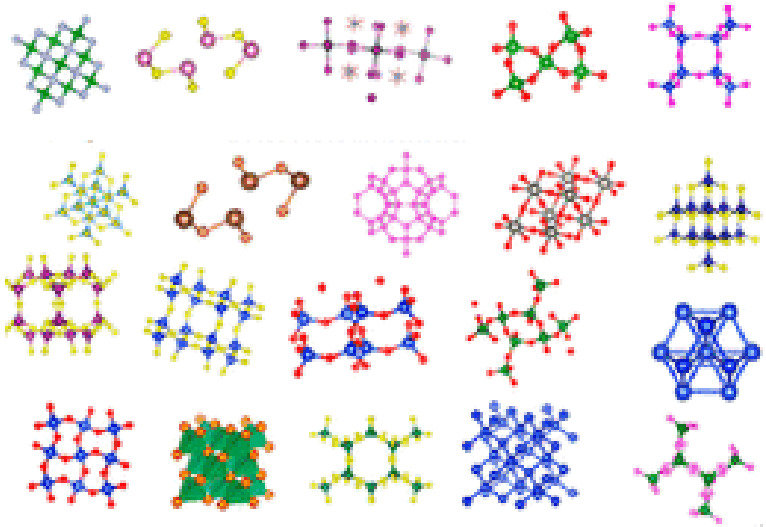
NEWS ICSD

31.10.2023

ICSD now contains 291,382 crystal structures

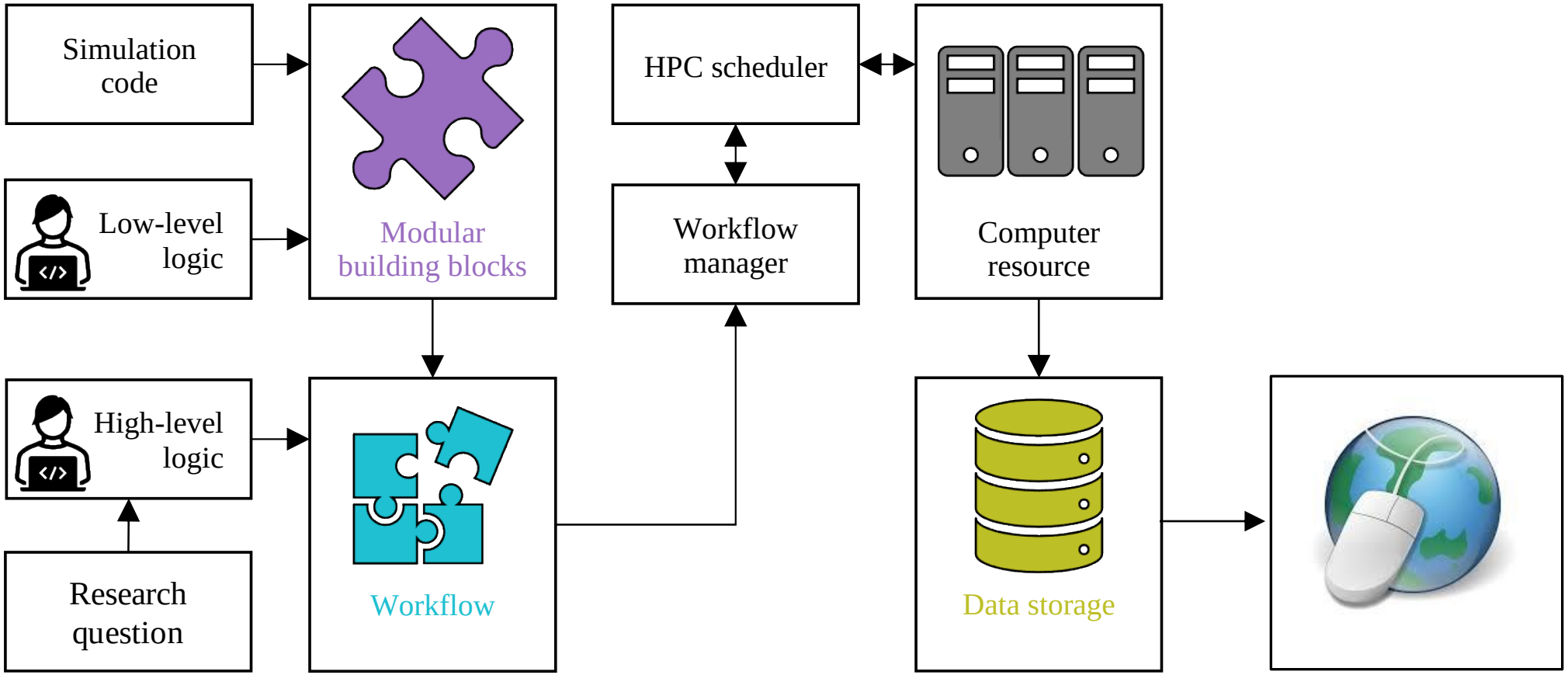
The ICSD web version 5.1.0 is now online.

read more



Import structures and compute properties!

TaskBlaster: Python framework for workflows



Scaling of TaskBlaster

- Demonstrated scaling of TaskBlaster to the entire LUMI supercomputer (g-partition).
- Running ca. 10.000 materials simulations concurrently.
- Not trivial! We used the experience to develop a LUMI-compatible version of TaskBlaster

- Approx. 10k GPUs
- 375 petaflops (or 1.5 mio. laptops)
- #3 in the world
- Operated by CSC, Finland

Porting GPAW to GPU: Status

- GPAW now has full GPU support for ground state calculations
- Still, a lot of work is required to port the entire code
- The same code runs on several platforms (CPU, NVIDIA, AMD)
- Parallelised with optional GPU aware MPI. Strong scaling and exascale performance is yet to be optimised.
- > 10 x node to node speedup over CPU nodes

Porting GPAW to GPU: How it was done

- cupy provides numpy-like support of GPU arrays. In a nutshell:

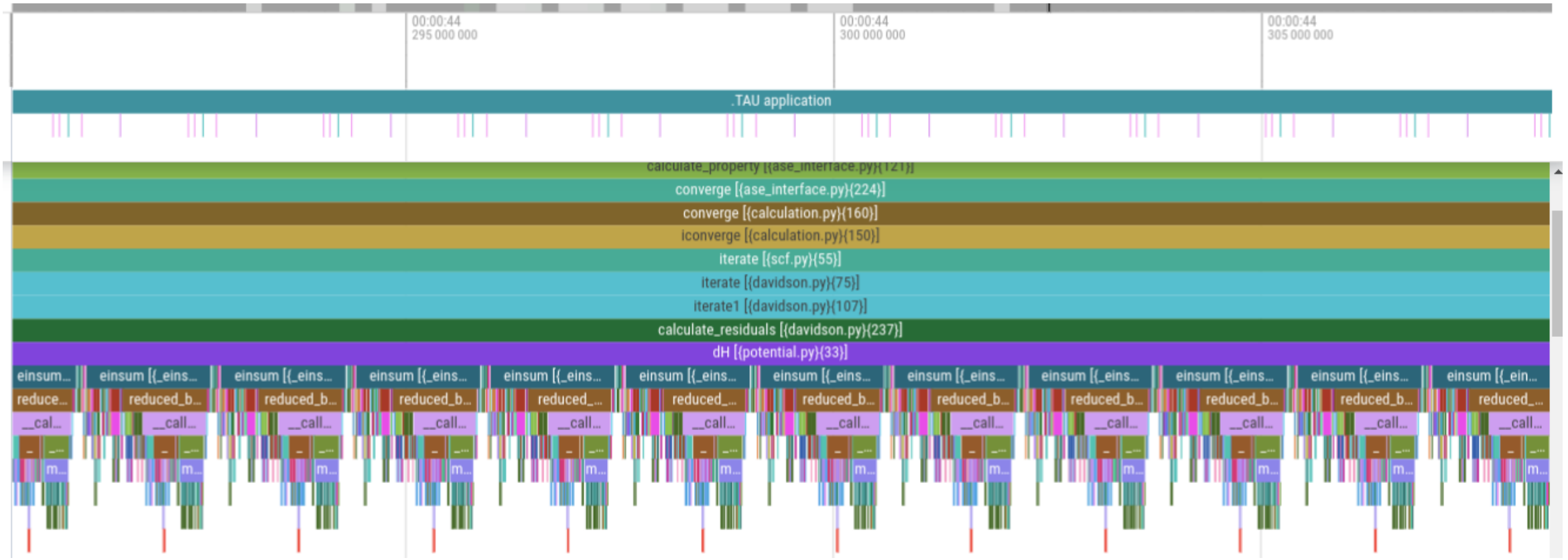
```
if CPU:
  import numpy as xp
if GPU:
  import cupy as xp
```

- Heavy use of profiling tools (Tau Python, omnitrace) to pinpoint bottlenecks
- Algorithmic improvements of bottlenecks
  - Batching (pass larger chunks of data to GPU per Python call, e.g. applylocalpot)
  - Write custom kernels (CUDA/HIP kernel library with about 10 kernels written)
  - Improvement of existing Python code

Porting GPAW to GPU: Profiling

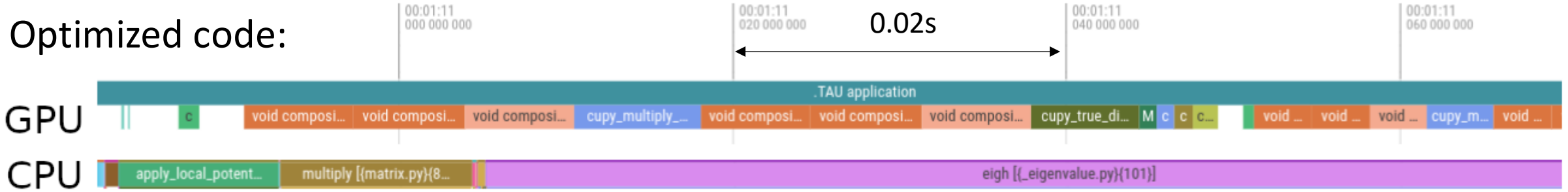
In the initial code (replacing NumPy by CuPy), the GPU was not utilised sufficiently

GPU:

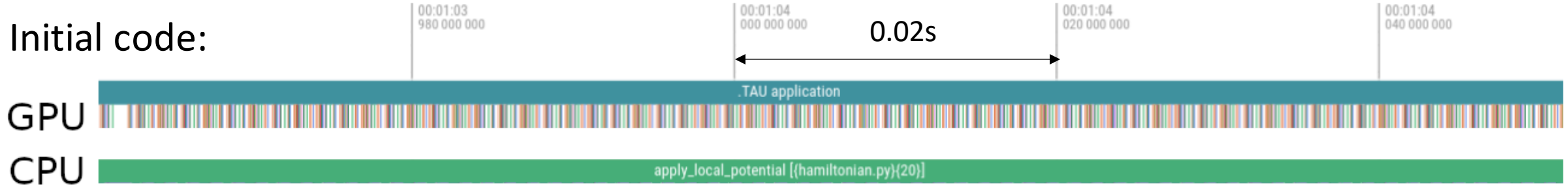


Porting GPAW to GPU: Profiling

Optimized code:



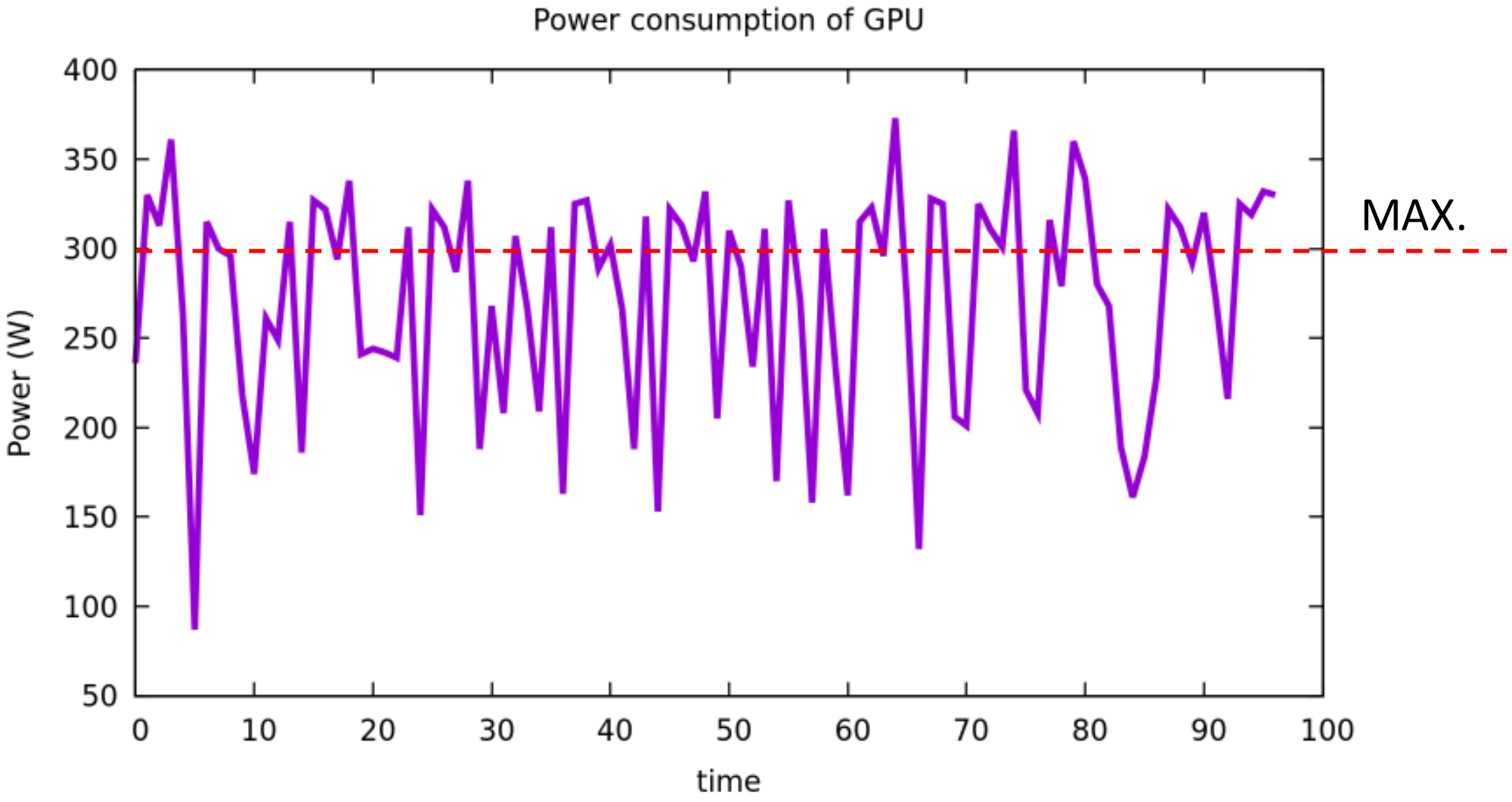
Initial code:



Initial code: GPUs not fully utilised. CPU busy submitting to GPU.

Optimized code: Batching of kernel calls in Python → CPU submits large work chunks → GPUs work all the time and is behind the CPU.

Porting GPAW to GPU: Power consumption



New GPAW: A multi-platform code

- **Supported architectures:**
  - NVIDIA
    - NVIDIA support via CuPy and CUDA
    - Tested on 4xNVIDIA A100 nodes on Mahti, CSC
  - AMD
    - AMD support via CuPy and HIP
    - Tested on AMD node 4xMI250X on LUMI
  - CPU
    - The GPU code shares the main code base with the CPU code
    - Includes a fake CuPy library to run GPU tests without GPU
- **Multi-platform testing**
  - Gitlab runner on LUMI (for signed commits only)
  - Local gitlab runner at DTU-Physics for CPU
  - Local gitlab runner at DTU-Physics for GPU (NVIDIA A4000)

GPU versus CPU performance

GPU cost / CPU cost

- **Hardware cost** — 7
  - Cost of NVIDIA A100 node: kr. 500k
  - Cost of Xeon 56 CPU node: kr. 75k
- **Running costs** — 2.5
  - GPU node: 4x300 W + 600 W (cooling) = 1.8 kW
  - Cost per 5 years: 5*1.8 kW * 2 kr./kWh = kr. 160k
  - CPU node (xeon56): 750 W
  - Cost per 5 year: 5*0.75 kW * 2 kr./kWh = kr. 65k
- **Total cost** — 5
  - GPU: kr. 660k
  - CPU: kr. 140k

Conclusions

- Atomic-scale computations can accelerate the understanding and discovery of novel materials for energy and quantum technologies.
- The danish DFT community needs easy access to large CPU computing resources (10^9 core hrs/year). GPUs cannot replace this need the next many years.
- TaskBlaster: Simple Python framework for automated workflows.
- GPAW-GPU multiplatform code made possible by several factors:
  - Python libraries (cupy)
  - Access to LUMI (thanks to DeiC)
  - Support from CSC and AMD, incl. setup of gitlab runner for testing
  - Excellent team members

Acknowledgements

People

Mikael Kuisma, DTU

Jens Jørgen Mortensen, DTU

Ask Hjorth Larsen, DTU

Tara Boland, DTU

Ole Holm Nielsen, DTU

Tuomas Rossi, CSC

Martti Louhivuori, CSC

Kjetil Haugen, AMD

Center for Nanostructured Graphene

Danmarks Grundforskningsfond
Danish National Research Foundation

erc European Research Council

VILLUM FONDEN

NOMAD

Sponsored by the European Union H2020