Machine Learning to Augment Material's Property Prediction

Hartwin Peelaers
Department of Physics & Astronomy
University of Kansas

First-principles calculations

No fitting parameters!

Start from the foundation of quantum mechanics

– many-body Schrödinger equation

$$\hat{\mathcal{H}}\Psi = E\Psi$$

All properties of the system

→ solve for many-body wavefunction

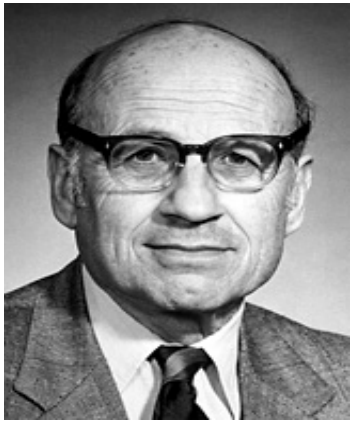
$\Psi(\vec{\mathbf{r}}_1, \vec{\mathbf{r}}_2, \cdots, \vec{\mathbf{r}}_N)$ → depends on 3*N* spatial coordinates

(3 Cartesian coordinates x N particles)

Example: bulk Si: discretize on 10x10x10 grid: need to store 10^{138} complex numbers

(there are approximately 10^{82} atoms in the universe...)

Density functional theory: introduction



Walter Kohn
1923 – 2016

Nobel prize in Chemistry
1998

Hohenberg and Kohn identified the electronic density as the fundamental quantity: $n(\vec{\mathbf{r}})$ → depends *only* on 3 spatial coordinates

All other quantities are functionals of the density:

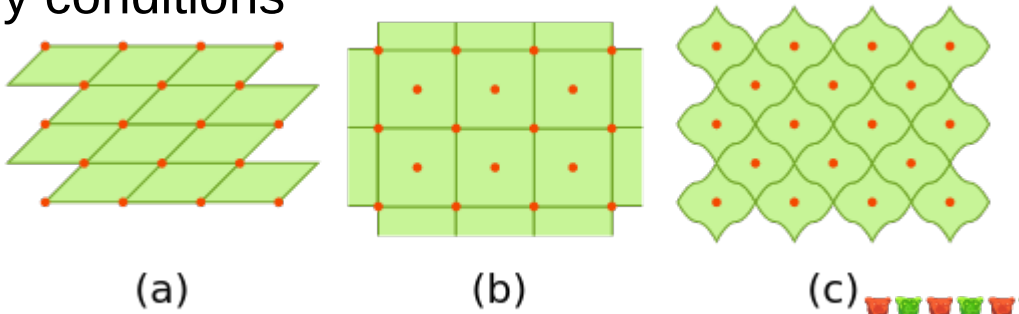
$$E[n(\vec{r})], \Psi[n(\vec{r})], O[n(\vec{r})], \cdots$$

Example: bulk Si: only need to store 10^5 complex numbers
→ only ~1.6 Mb

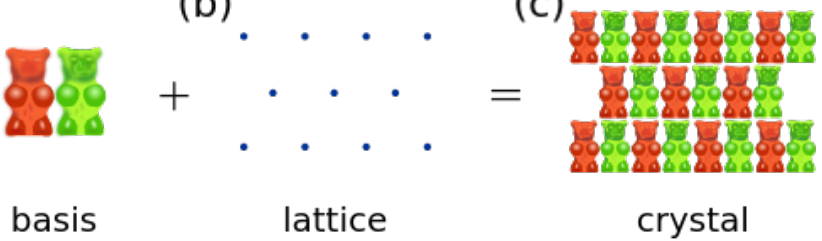
How to describe a crystal?

Infinite crystal: use periodic boundary conditions

Use smallest repeat unit: unit cell

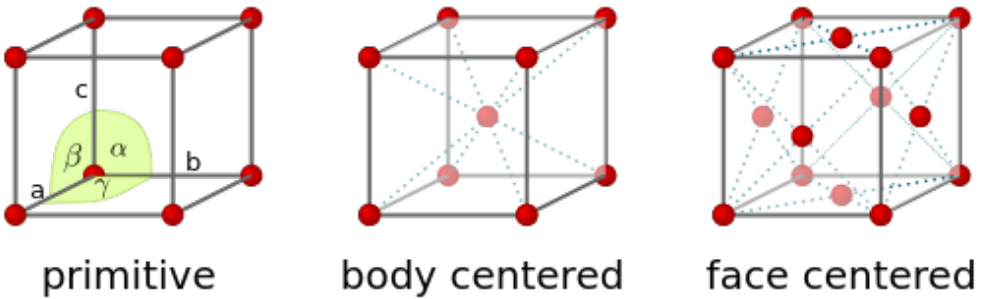


Specify vectors of unit cell: “lattice”
+ positions of atoms: “basis” → crystal is defined

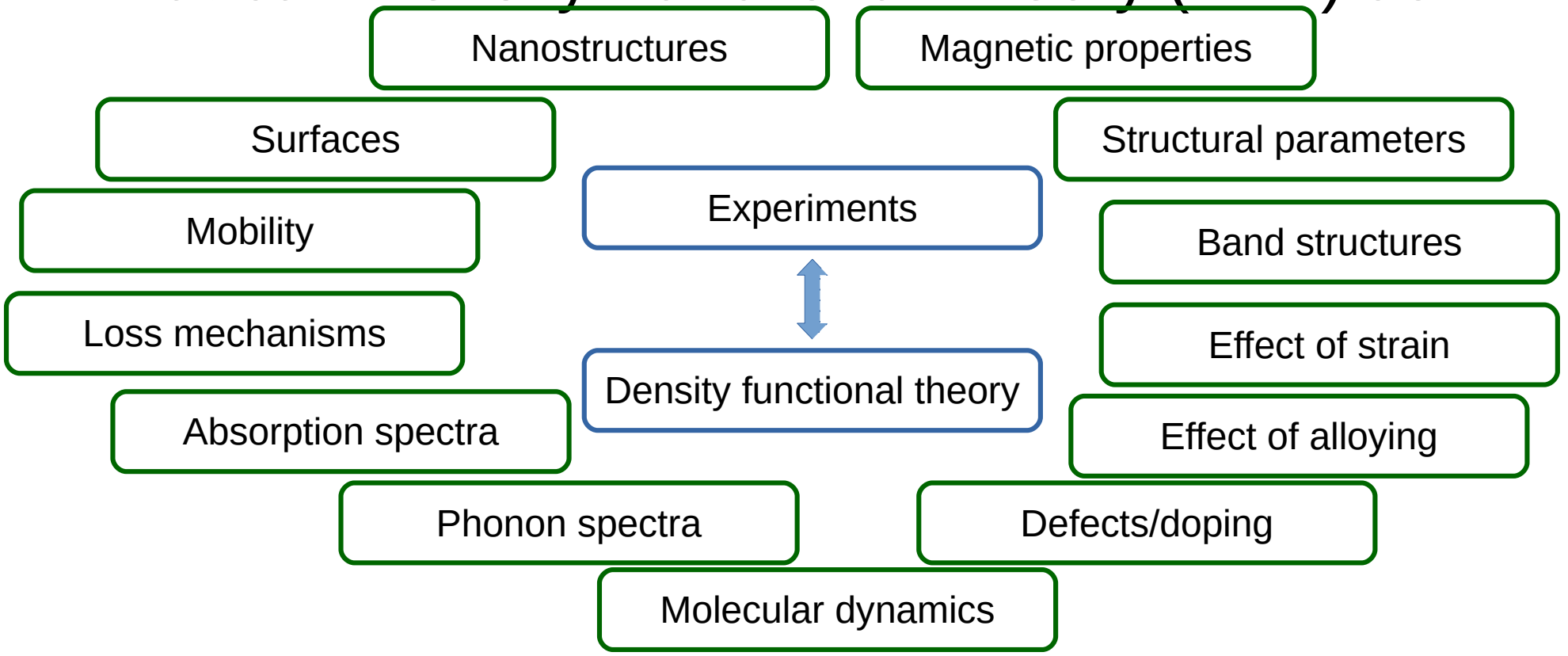


Different materials: different atoms, different unit cells (hexagonal, cubic,…), different positions of atoms within unit cell

Symmetry is important



What can Density Functional Theory (DFT) do?



Lots of options, but calculations are computationally expensive

Machine learning, big data, and DFT

Calculations are expensive → want to avoid them! → machine learning → requires a lot of calculations → ...

Step 1: get enough data!

Materials genome initiative: “discover, manufacture, and deploy advanced materials twice as fast, at a fraction of the cost”

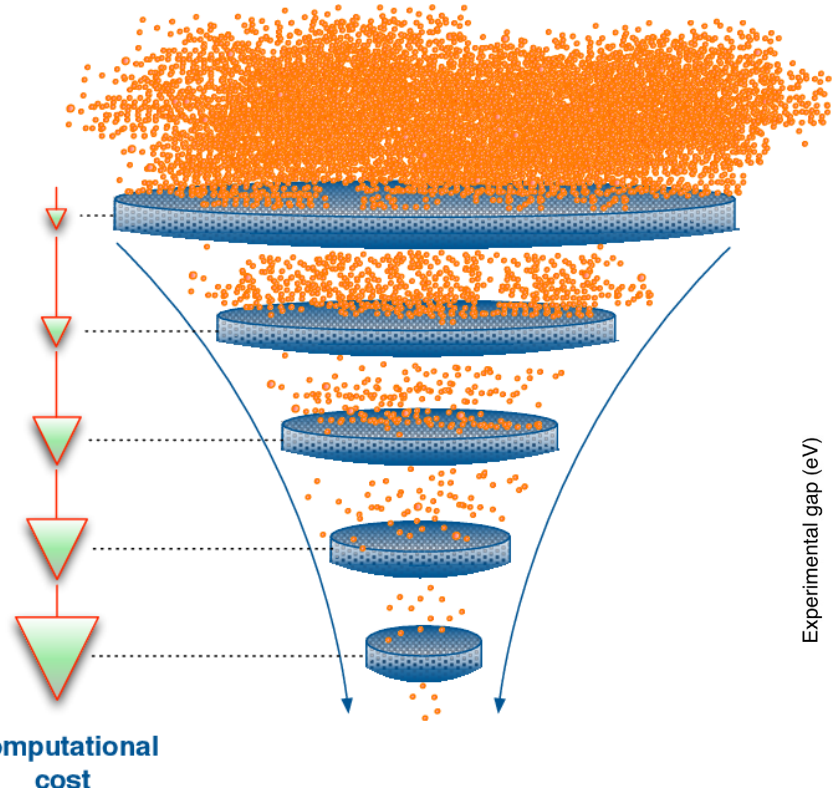
→ calculational aspect: Materials Project: “high-throughput calculations”: currently basic data for 124,515 inorganic compounds

But: most calculations calculated using methods that are fast (on a supercomputer), but not that accurate...

Machine learning, big data, and DFT

Step 2: how to use inaccurate data?

Funnel method:

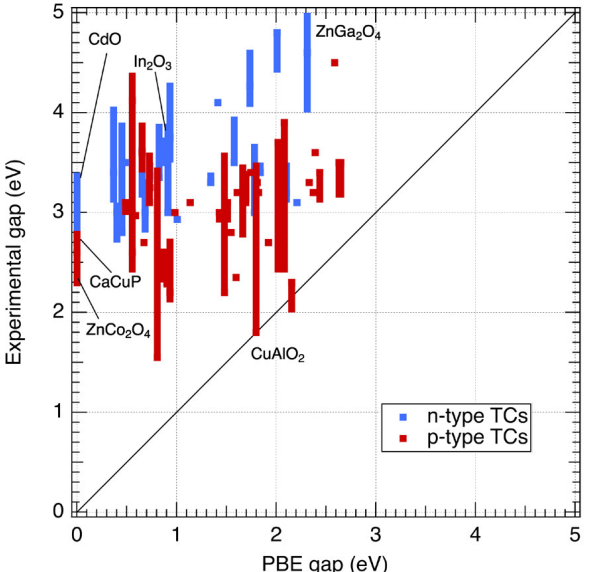


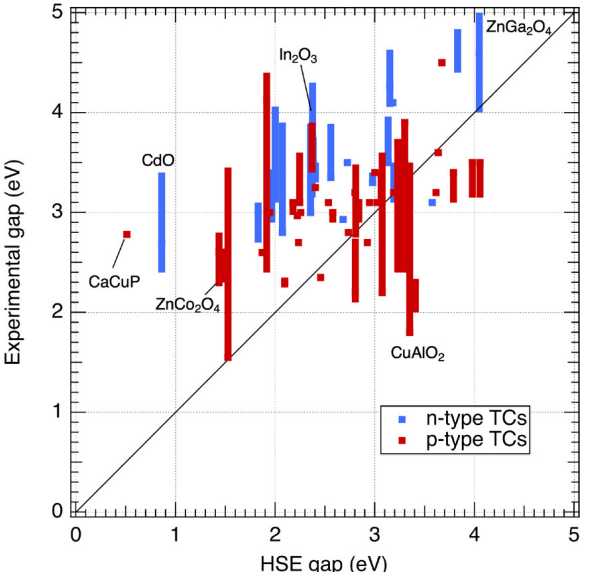
Woods-Robinson *et al.*, Chem. Mater. **30**, 8375 (2018)

Balance amount of calculations with accuracy (computational cost)

Obtain criteria to filter calculations → need descriptors

1) Use physical intuition+knowledge of accuracy of calculations: new transparent conducting oxide: band gap needs to be large enough

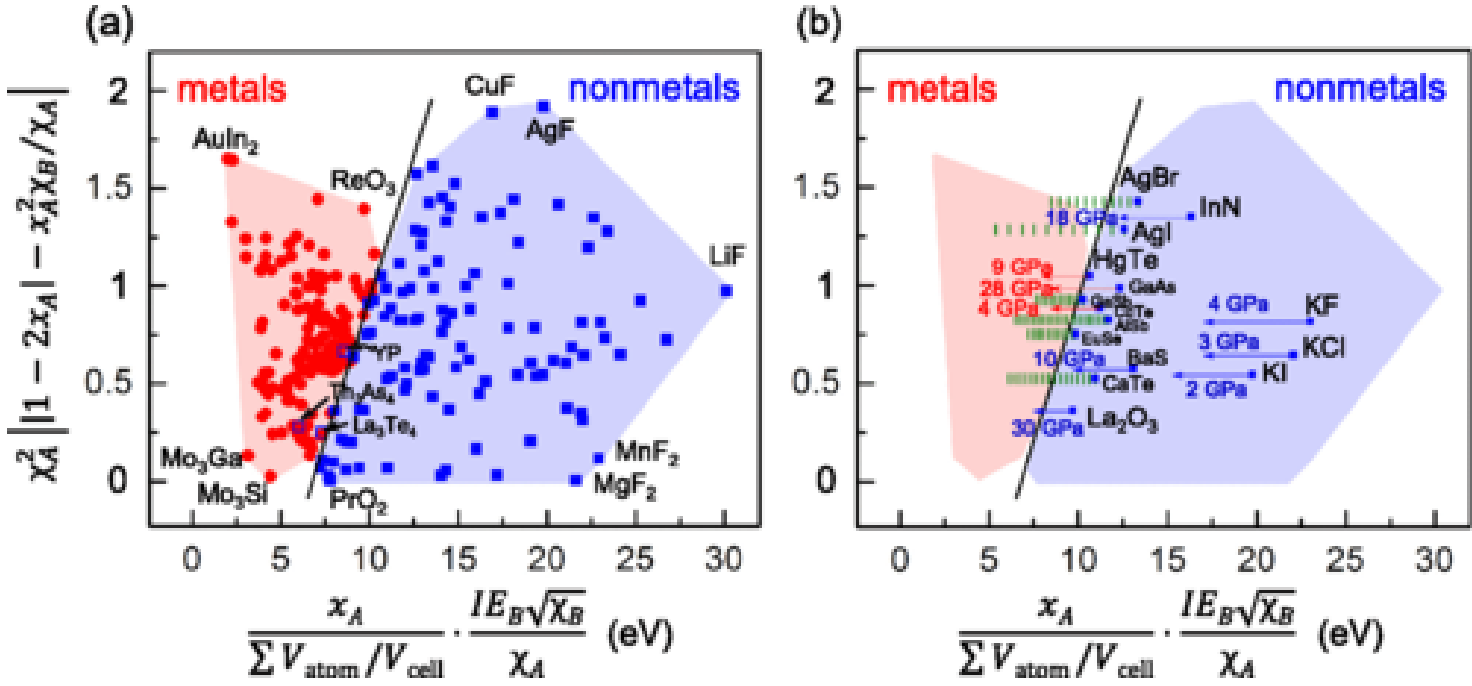




Machine learning, big data, and DFT

Alternative: Use small amount of accurate data + machine learning to find (unexpected) descriptors from small training sets

1) create a large pool of possible descriptors by combining elementary physical properties of atoms or easily calculated quantities
2) use **compressed sensing** to find best descriptor to distinguish metal/non-metal, crystal structure, topological properties,...
3) once you have a descriptor → use for other materials to make predictions

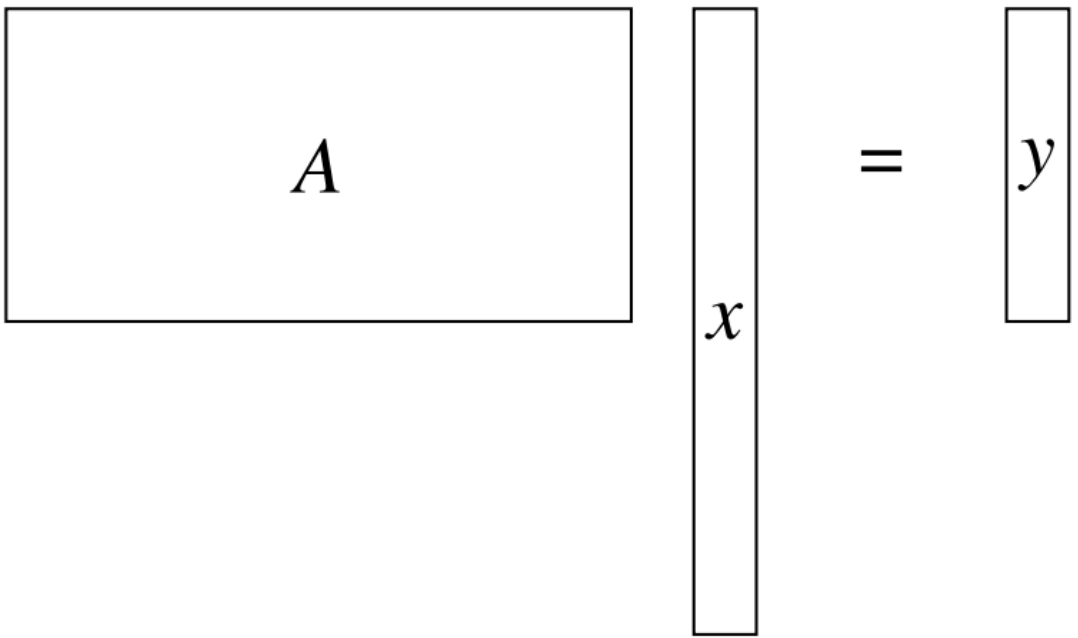


Ouyang *et al.*, Phys. Rev. Materials **2**, 083802 (2018)

Compressed sensing

Signal processing technique: find solutions to underdetermined linear systems

Problem:



Matrices:

A: (m by N) → measurement matrix
x: (N by 1) → actual signal
y: (m by 1) → measurement vector

find *x* such that *y=Ax* with *m<<N*

→ no unique solution
→ not possible to reconstruct x from the *m* measurements *y*?

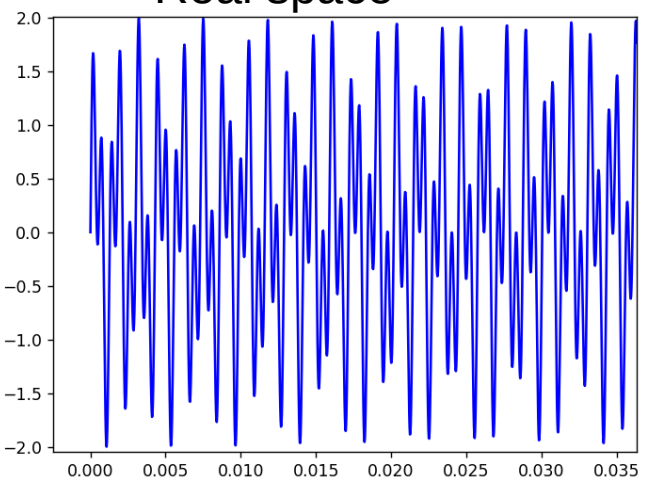
Compressed sensing

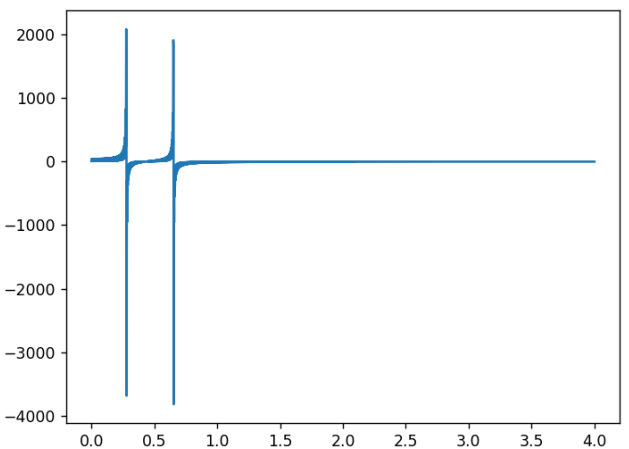
$$A \; x = y$$

Many signals have redundancy → sparse when represented in some domain
E.g.: in reciprocal space

→ use this to look for sparsest solution of system

Real space → Reciprocal space





http://www.pyrunner.com/weblog/2016/05/26/compressed-sensing-python

Nelson *et al.*, Phys. Rev. B **88**, 155105 (2013)

Compressed sensing

$$A \quad x = y$$

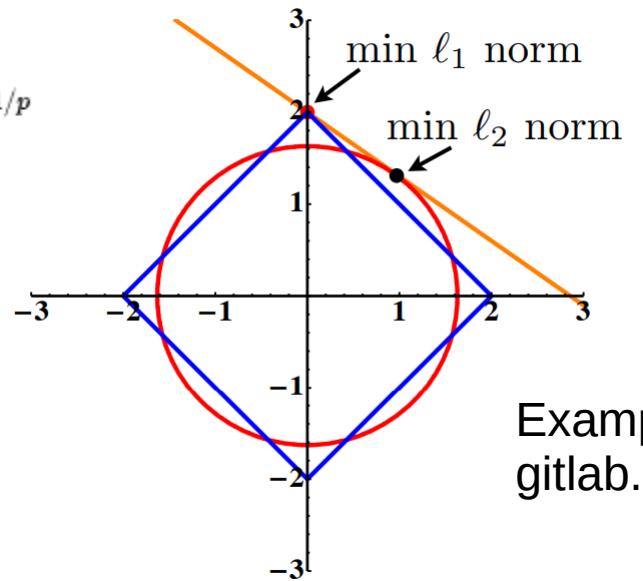
Many signals have redundancy → sparse when represented in some domain
E.g.: in reciprocal space

→ use this to look for sparsest solution of system

Minimize L_1 norm

$$\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}$$

L_2 norm: vector length
L_1 norm: Manhattan distance



Why L_1 norm? → Find sparse solutions

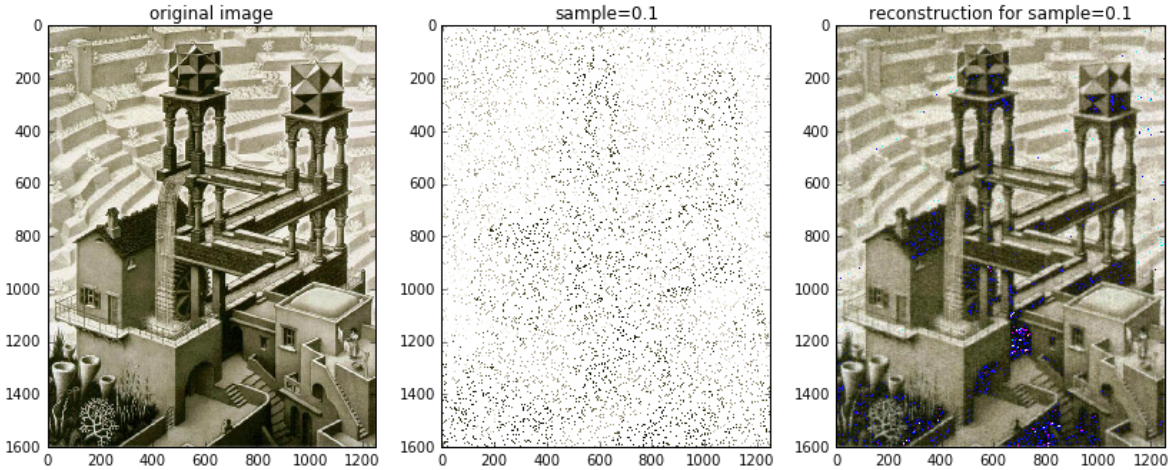
Example: 10y + 7x = 20, find solution with smallest norm

Example notebook:
gitlab.com/peelaers/machine-learning-talk

http://www.pyrunner.com/weblog/2016/05/26/compressed-sensing-python

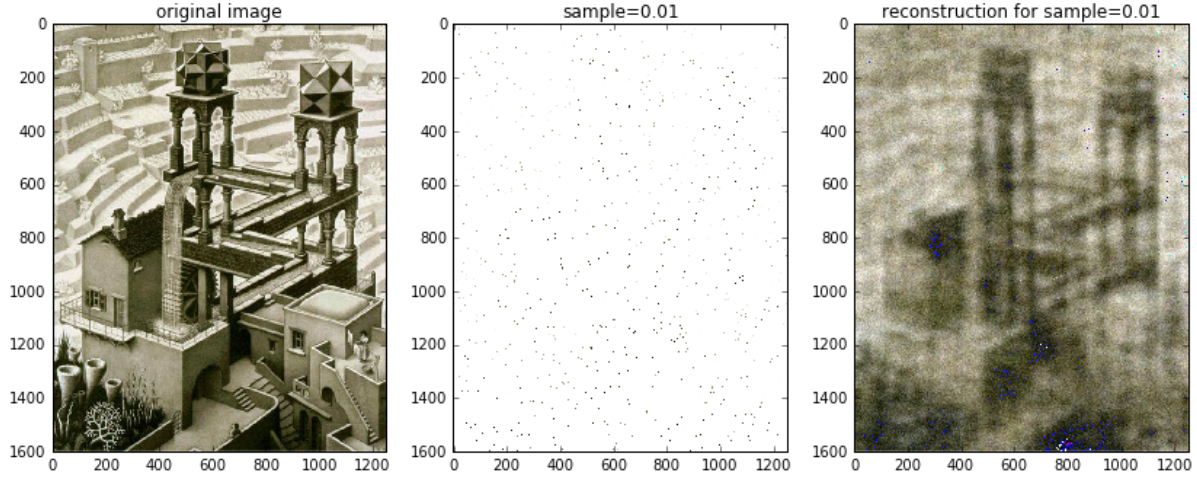
Nelson *et al.*, Phys. Rev. B **88**, 155105 (2013)

Compressed sensing



Reconstruct image with only 10% of data

Reconstruct image with only 1% of data!



http://www.pyrunner.com/weblog/2016/05/26/compressed-sensing-python

Finding stable alloys

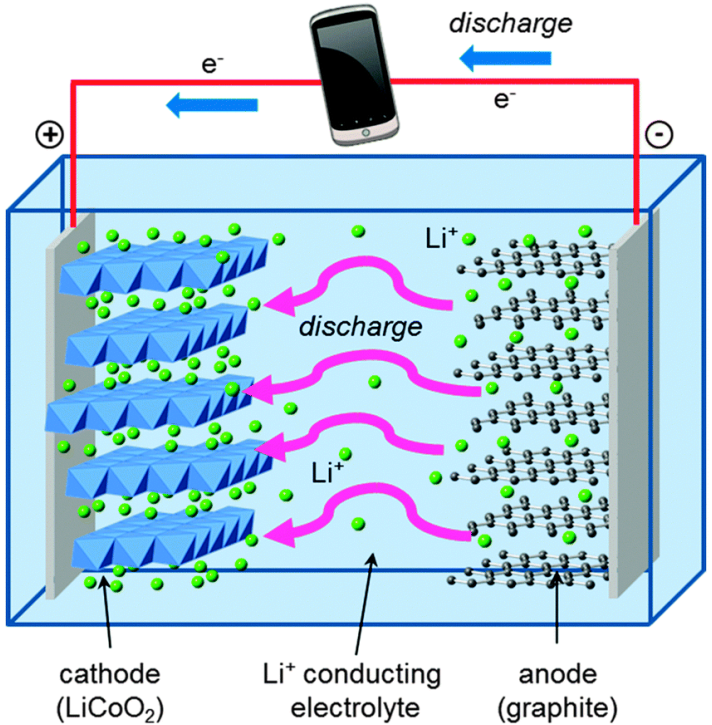
Alloying is a great tool to modify the properties of a material

Examples: stainless steel: iron, 11% chromium, max 1.2% carbon, some molybdenum,...

Semiconductors: change band gap, lattice constants, band alignment

Battery electrodes:

How do ions intercalate in cathode/anode?
Are there low energy ordered structures?
What are the thermodynamic properties?



Calculational issue:

Need to be able to describe disordered systems and small alloy concentrations → requires large simulation cells

Cluster expansion

Possible solution: Use first-principles calculations in small simulation cells to construct a model of the alloy, and use that!

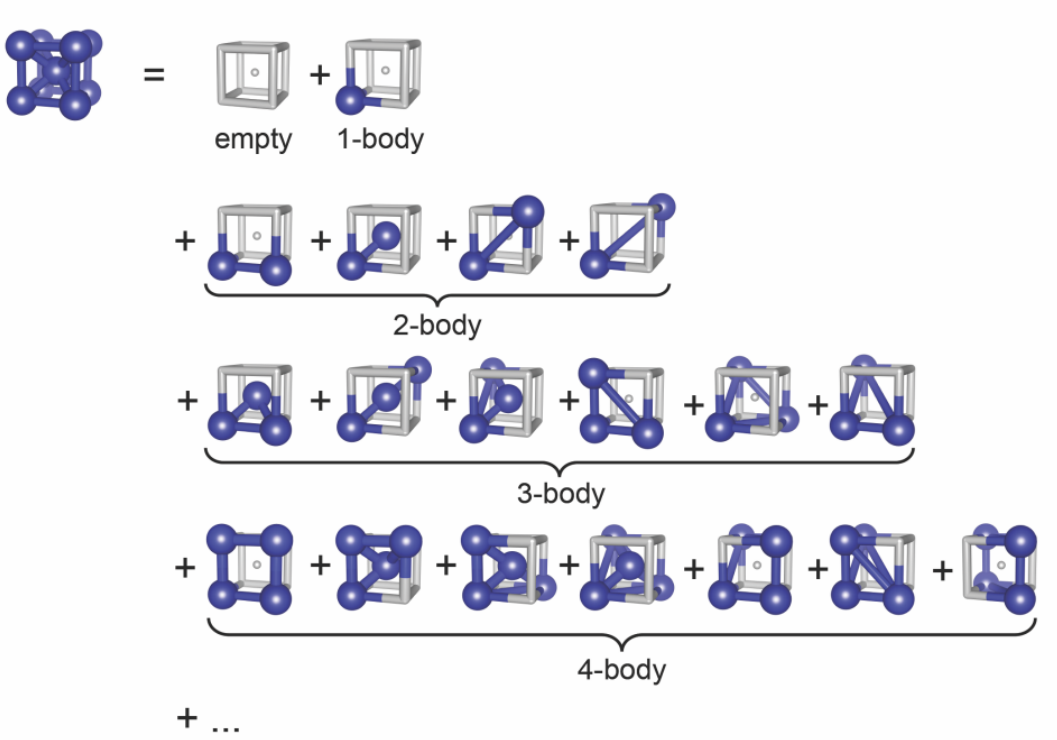
But how to do so?

Use an expansion (similar to Taylor series)

$$E(\sigma) = E_0 + \sum_f \bar{\Pi}_f(\sigma) J_f$$

E: property of interest
σ: vector identifying alloy: decode all information as pseudo-spins:
if a position is occupied by atom A: +1, else 0
→ vector uniquely identifies alloy

f: sum over clusters
J: coefficients (unknown, but will fit from calculations)
Π: cluster basis function (accounting and symmetry)



Chang *et al.*, J. Phys.: Condens. Matter **31**, 325901 (2019)

Cluster expansion

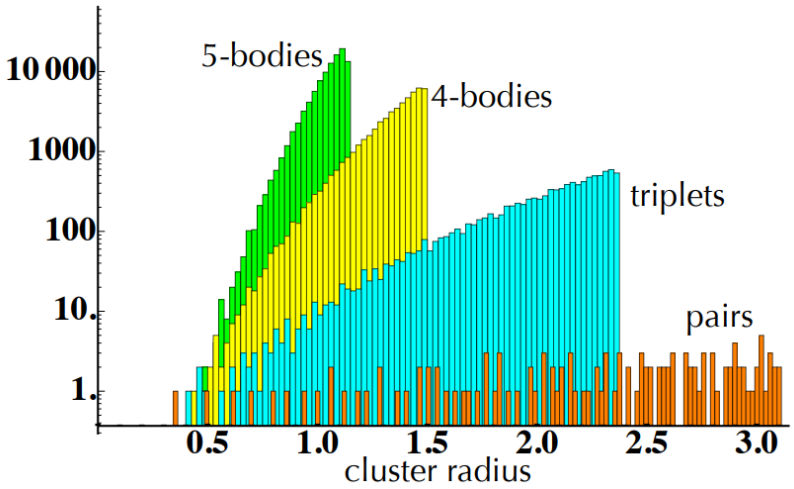
$$E(\sigma) = E_0 + \sum_f \bar{\Pi}_f(\sigma) J_f$$

Big question: how to pick the clusters:
*distance between elements (cluster radius)
*number of elements in cluster: when to truncate?
→ combinatorics explode
And remember: each calculation we do is very time-consuming

So how to select as few calculations as possible, while obtaining a good expansion?

Possibilities include: lucky guesses, genetic algorithms,…

But also compressive sensing! $\bar{\mathbf{\Pi}}\vec{J} = \vec{E}$



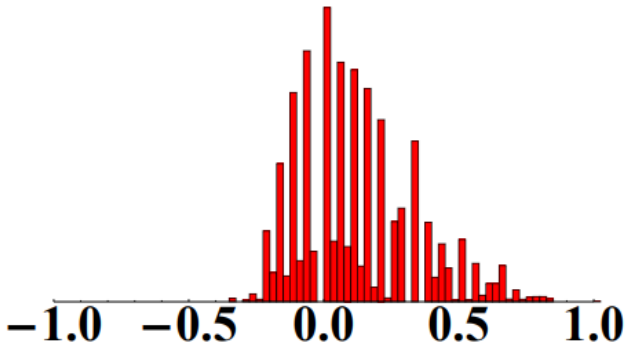
Structure selection procedure

1. Generate a random vector π on the unit hypersphere.
2. Orthogonalize π to all rows of the current sensing matrix $\bar{\mathbf{\Pi}}$.
3. Normalize π
4. Find the nearest crystal structure to the orthonormalized π.
5. Add the structure to the training set.
6. Update the matrix $\bar{\mathbf{\Pi}}$. Go back to step 1.

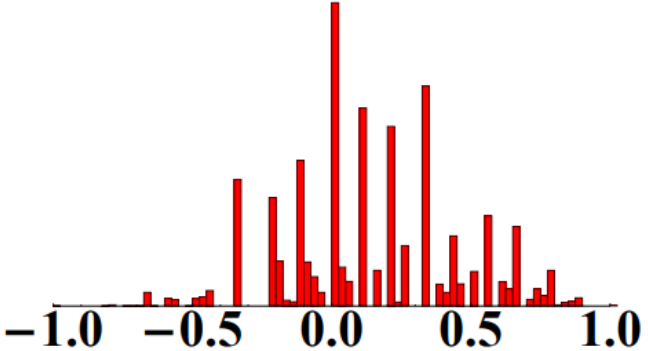
Nelson *et al.*, Phys. Rev. B **88**, 155105 (2013)

Cluster expansion

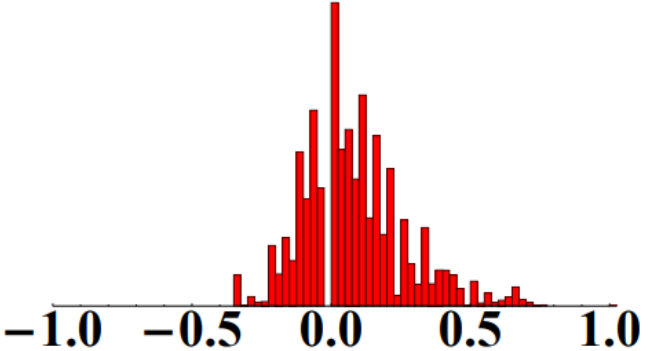
Random structures do not exploit sparseness and shape of cluster functions



Pairs: nearest-neighbor



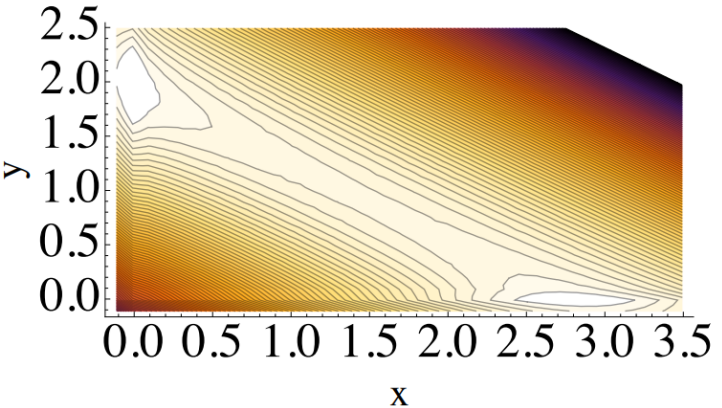
second-nearest



third-nearest neighbors

Combine selection of new test calculations with Bayesian inference

But maintain sparsity by selecting the shape of the priors



Nelson *et al.*, Phys. Rev. B **88**, 155105 (2013)

Finding ground state structure of a material

Usually: we already know a lot of information: symmetry, stoichiometry, structure of chemical similar structures (e.g., if we want to find the crystal structure of Ge, the known structure of Si would be a good guess)

→ we can usually find ground state

But: if all this information is absent, the parameter space is huge! (and calculations are time-consuming) (e.g., what structures are found at high pressure)

Lots of methods to avoid brute force: simulated annealing, genetic algorithms, particle swarm optimalisations, Bayesian optimization,...

→ machine learning can play an important role

Simulating kinetic processes

First-principles calculations: all at 0K

How to introduce temperature: perturbation theory

But not sufficient to simulate a lot of processes, that require large number of atoms:

- how do Li-ions move between the layers of the electrode?
- how do liquids behave?
- how do materials melt?
- how are crystals grown?
- protein folding

→ need to be able to do molecular dynamics: extremely expensive using first-principles

Possible solution: use limited first-principle calculations to obtain classical potentials, and use these potentials to do the simulations

→ use machine-learning to obtain these potentials

Conclusions

- Incomplete overview of machine learning to augment first-principles calculations
- Focused on compressive sensing
  - Difference between L^1 and L^2 norm
  - Example: reconstructing a wave from rough sampling
- Use of compressive sensing to find insights from limited amount of data
- Cluster expansion as method to describe alloys/intercalation/…
  - Role of compressive sensing
- Structure minimization and dynamics

Machine learning has lots of potential to augment first-principles calculations!

Questions?