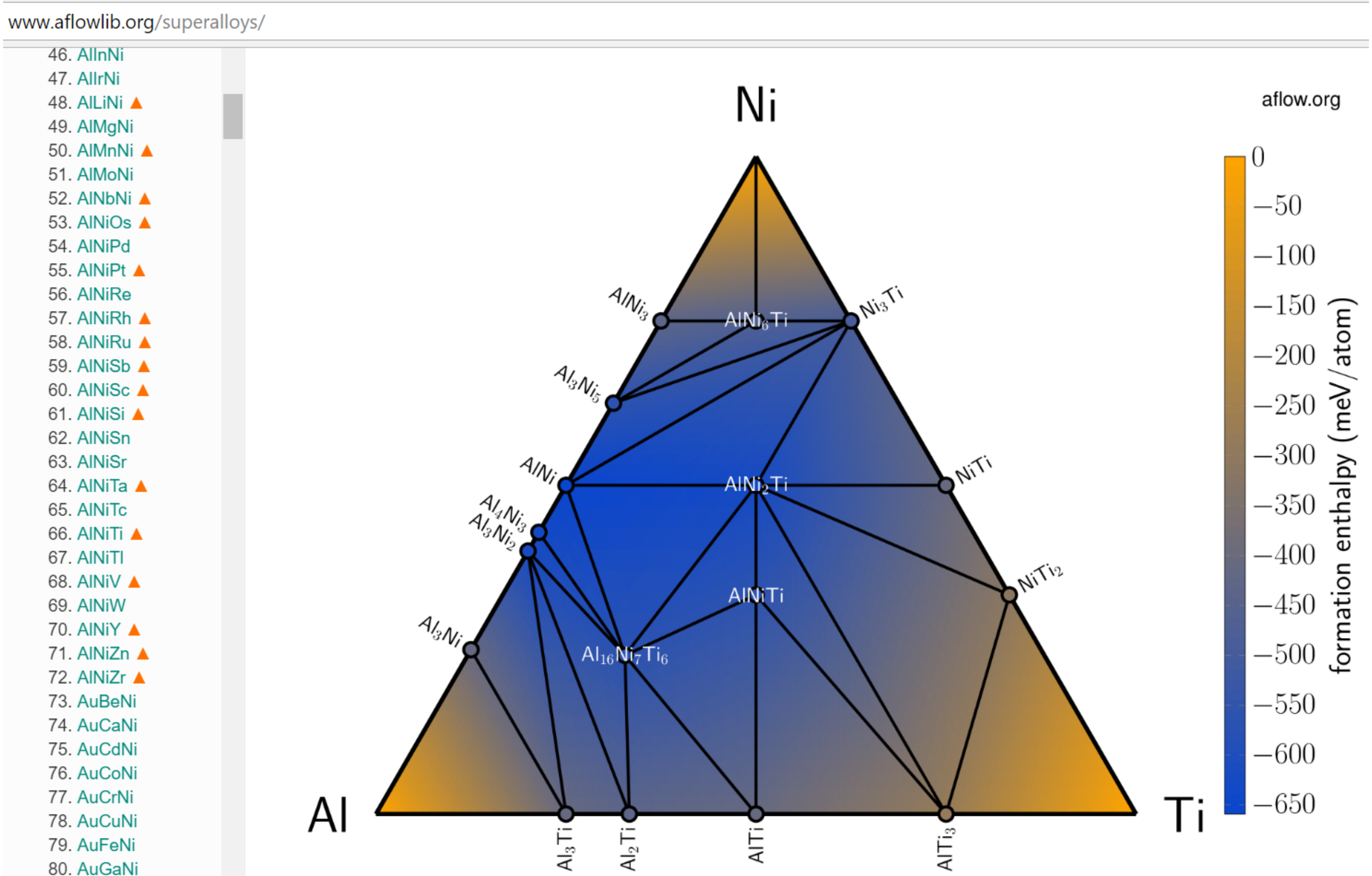
Machine learning of interatomic interaction

Alexander Shapeev[1],

1: Skoltech (Moscow)

Skoltech, AMM course

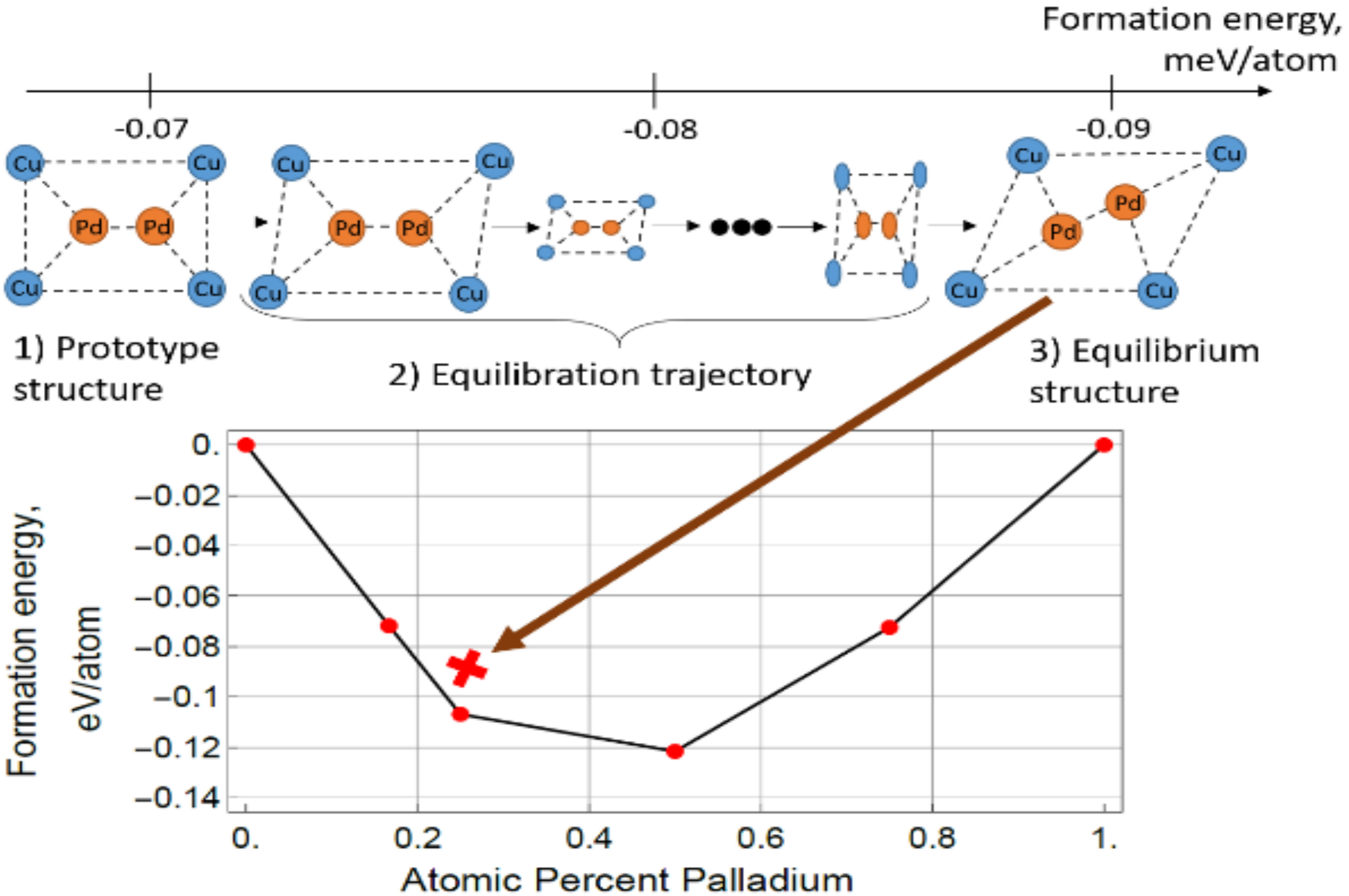
19 May 2020

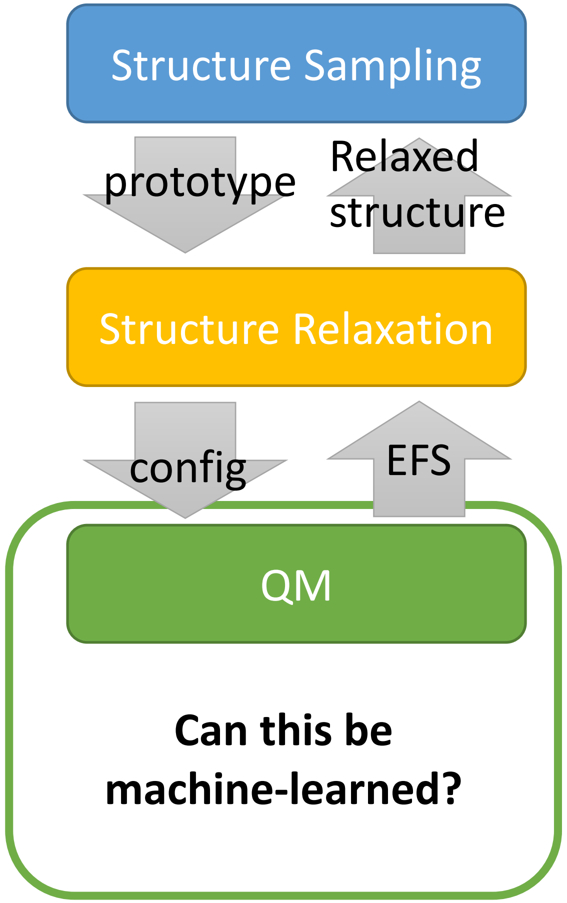


www.aflowlib.org/superalloys/

46. AlInNi
47. AlIrNi
48. AlLiNi ▲
49. AlMgNi
50. AlMnNi ▲
51. AlMoNi
52. AlNbNi ▲
53. AlNiOs ▲
54. AlNiPd
55. AlNiPt ▲
56. AlNiRe
57. AlNiRh ▲
58. AlNiRu ▲
59. AlNiSb ▲
60. AlNiSc ▲
61. AlNiSi ▲
62. AlNiSn
63. AlNiSr
64. AlNiTa ▲
65. AlNiTc
66. AlNiTi ▲
67. AlNiTl
68. AlNiV ▲
69. AlNiW
70. AlNiY ▲
71. AlNiZn ▲
72. AlNiZr ▲
73. AuBeNi
74. AuCaNi
75. AuCdNi
76. AuCoNi
77. AuCrNi
78. AuCuNi
79. AuFeNi
80. AuGaNi

Prediction of convex hull of stable alloys





Structure Sampling

prototype

Relaxed structure

Structure Relaxation

config

EFS

QM

Can this be machine-learned?

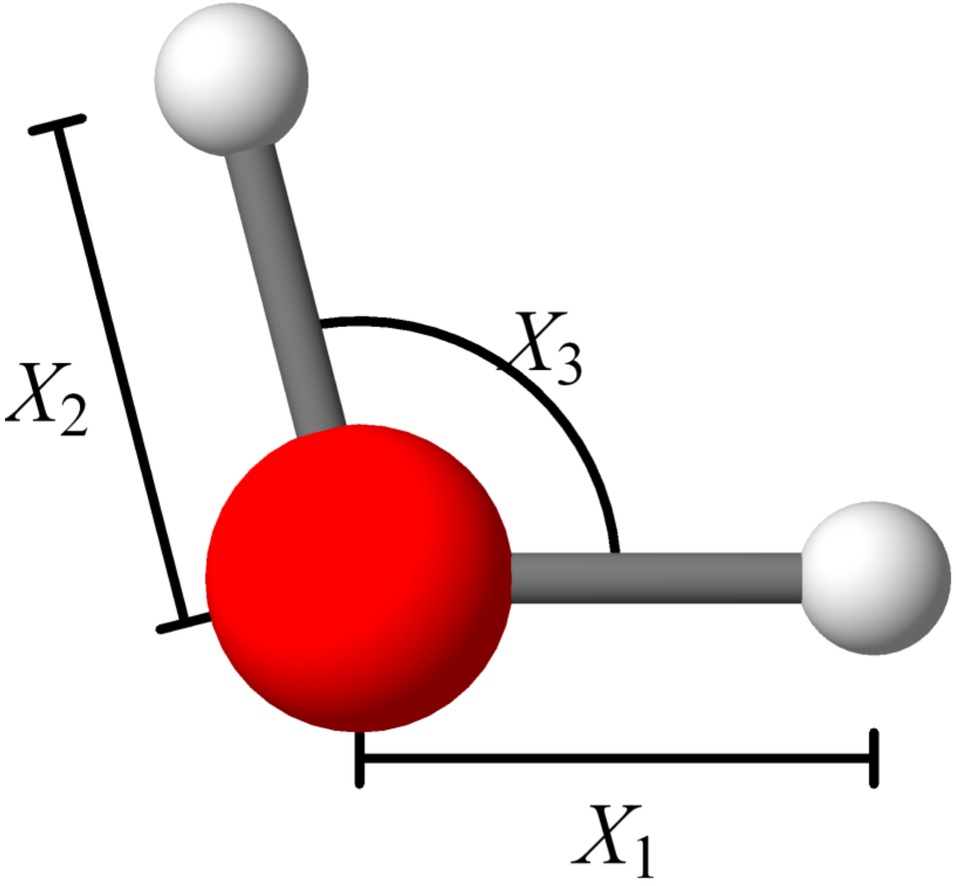
Machine-learning interatomic potentials

My perspective

Machine learning as interpolation,

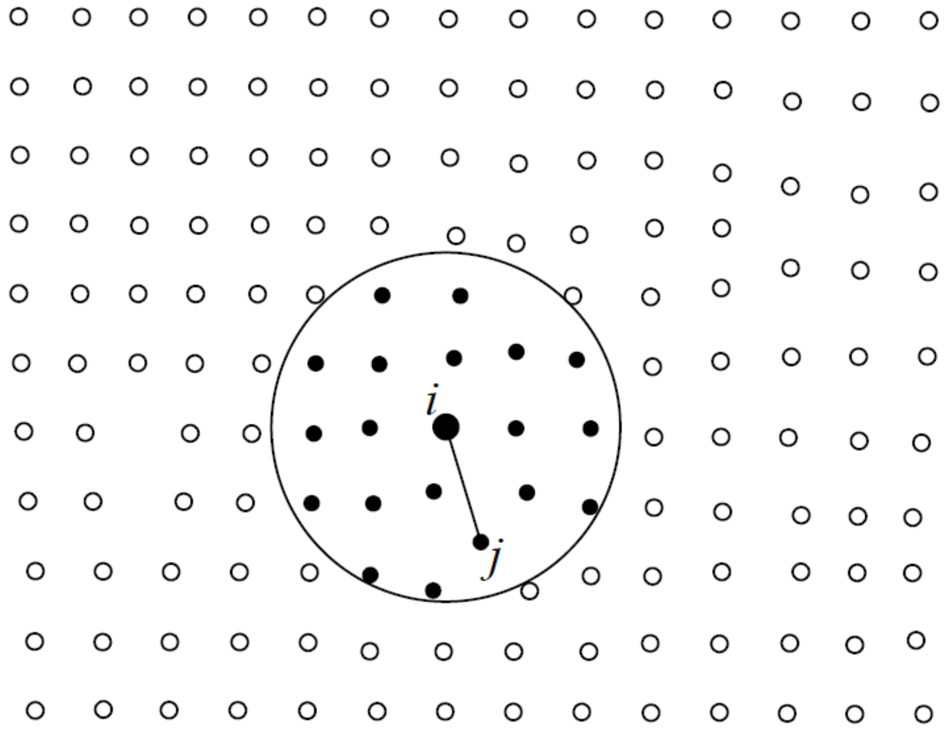
… data-driven and multidimensional.

- Problem: Given $E^{\mathrm{qm}}(\boldsymbol{X})$, interpolate it with $E(\boldsymbol{X})$
- Issue: no transferability w.r.t. the number of atoms
- Solution: <u>use locality</u>! (An atom interacts only with 10-100 neighboring atoms)



Traditional fitting

- Embedded atom model: $E = \sum_i V(r_{i1}, r_{i2}, \ldots)$,
- $V(\boldsymbol{r_i}) = \sum_j \varphi(r_{ij}) + F\left(\sum_j \rho(r_{ij})\right)$.
- Early interatomic potentials (=force fields) had few (three) parameters fitted from few experimental data (elastic constants, defect formation energy, etc.)
- Later potentials have tens of coefficients (e.g., spline coefficients) fitted from the QM data.
- What is different now: <u>there are lots of data</u>!
- So, the question is: *how to incorporate lots of data into the models*?



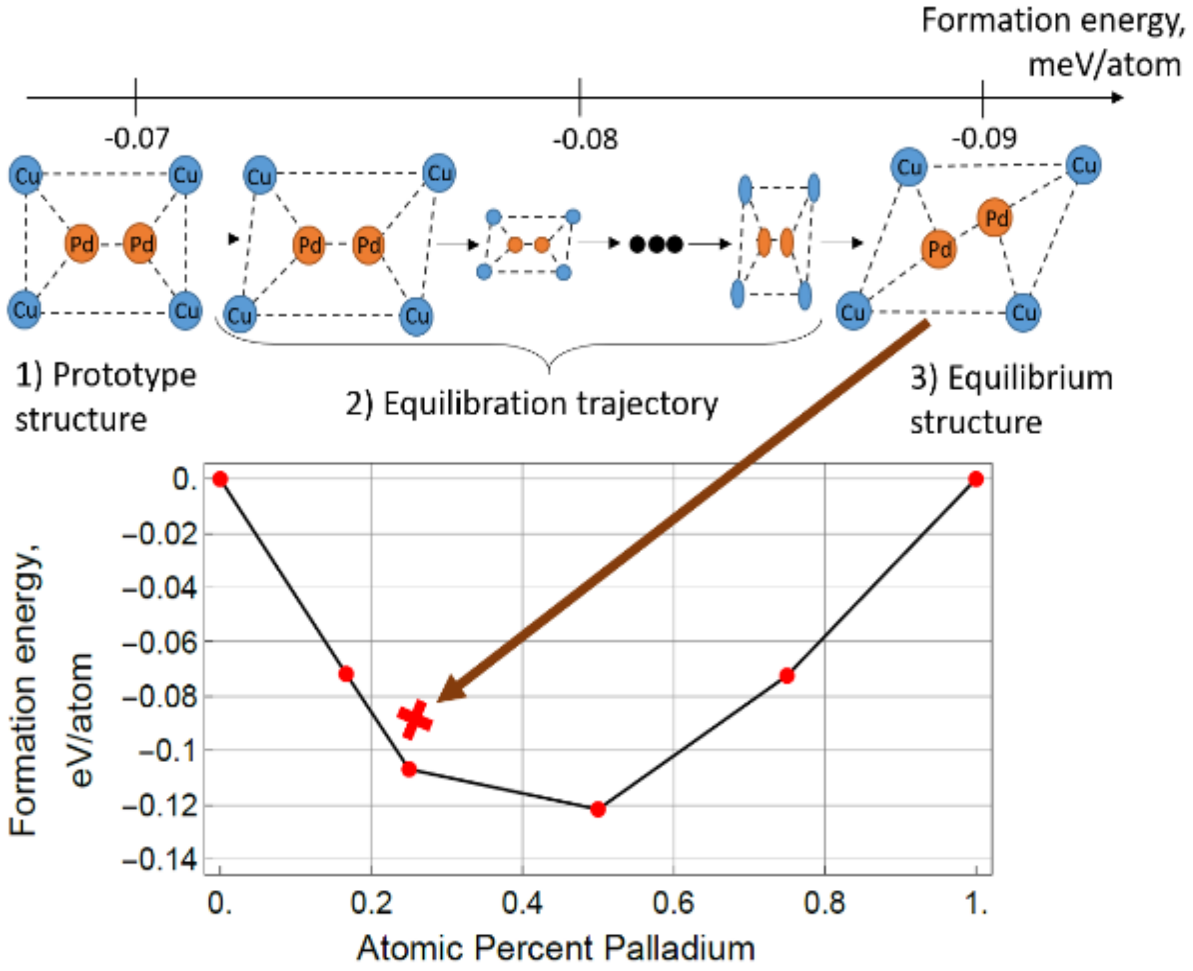
Machine-learning ideology:

1. Choose a (machine-learning) model $E = E(\boldsymbol{x})$
   ($\boldsymbol{x}$ is an atomic configuration)
2. We want to minimize $|E^{\mathrm{qm}} - E|$.
   So we:

- Generate data: $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots; E^{\mathrm{qm}}\left(\boldsymbol{x}^{(1)}\right), E^{\mathrm{qm}}\left(\boldsymbol{x}^{(2)}\right), \ldots, \boldsymbol{f}^{\mathrm{qm}}\left(\boldsymbol{x}^{(1)}\right), \ldots$
- Minimize on data: $\sum_i \left|E\left(x^{(i)}\right) - E^{\mathrm{qm}}\left(x^{(i)}\right)\right|^2 + (\text{forces})+\ldots$

But what if sampling the right $\boldsymbol{x}^{(i)}$
is a part of the problem?

Illustration: calculating convex hull



Problem:

- accurate sampling of ground state structures

needs

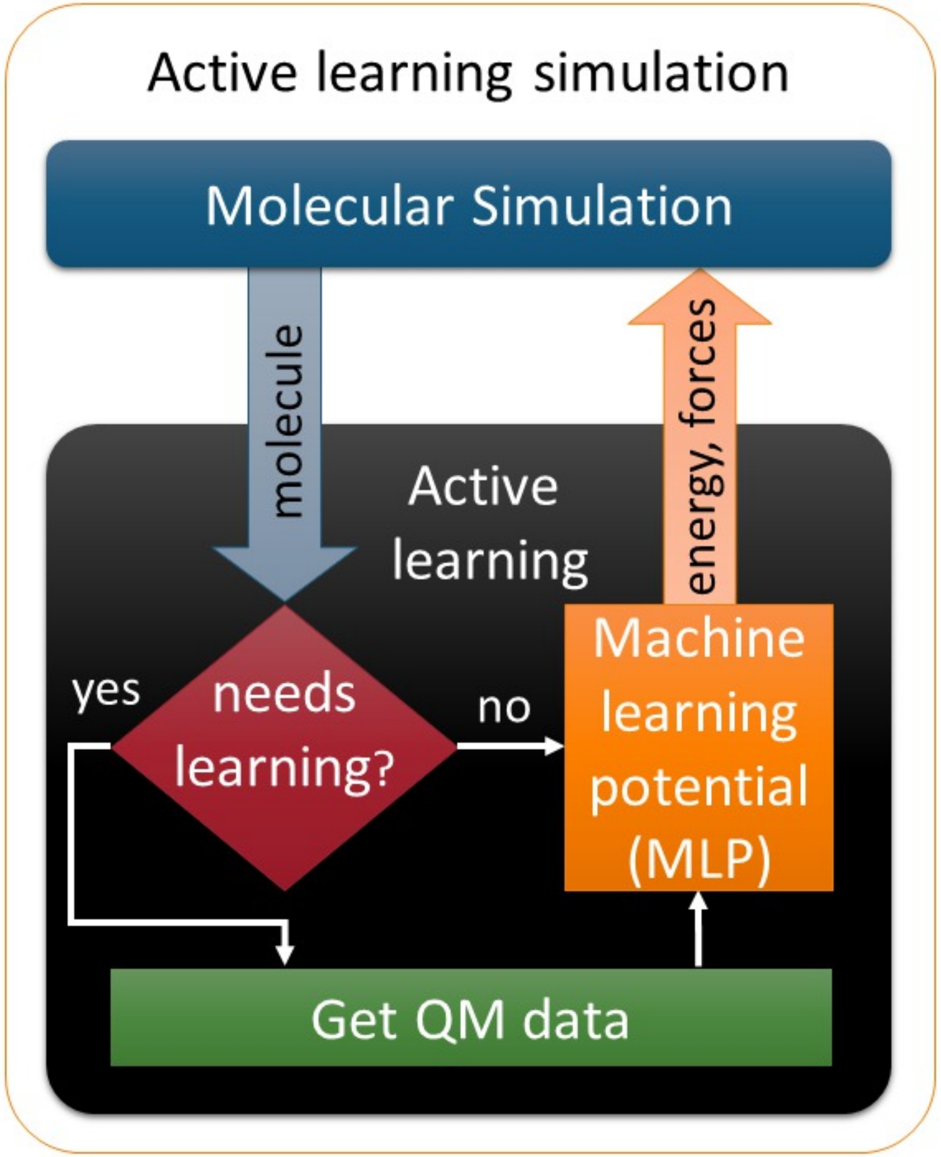
- accurate approximation of PES

which needs

- accurate sampling of ground state structures

which needs …

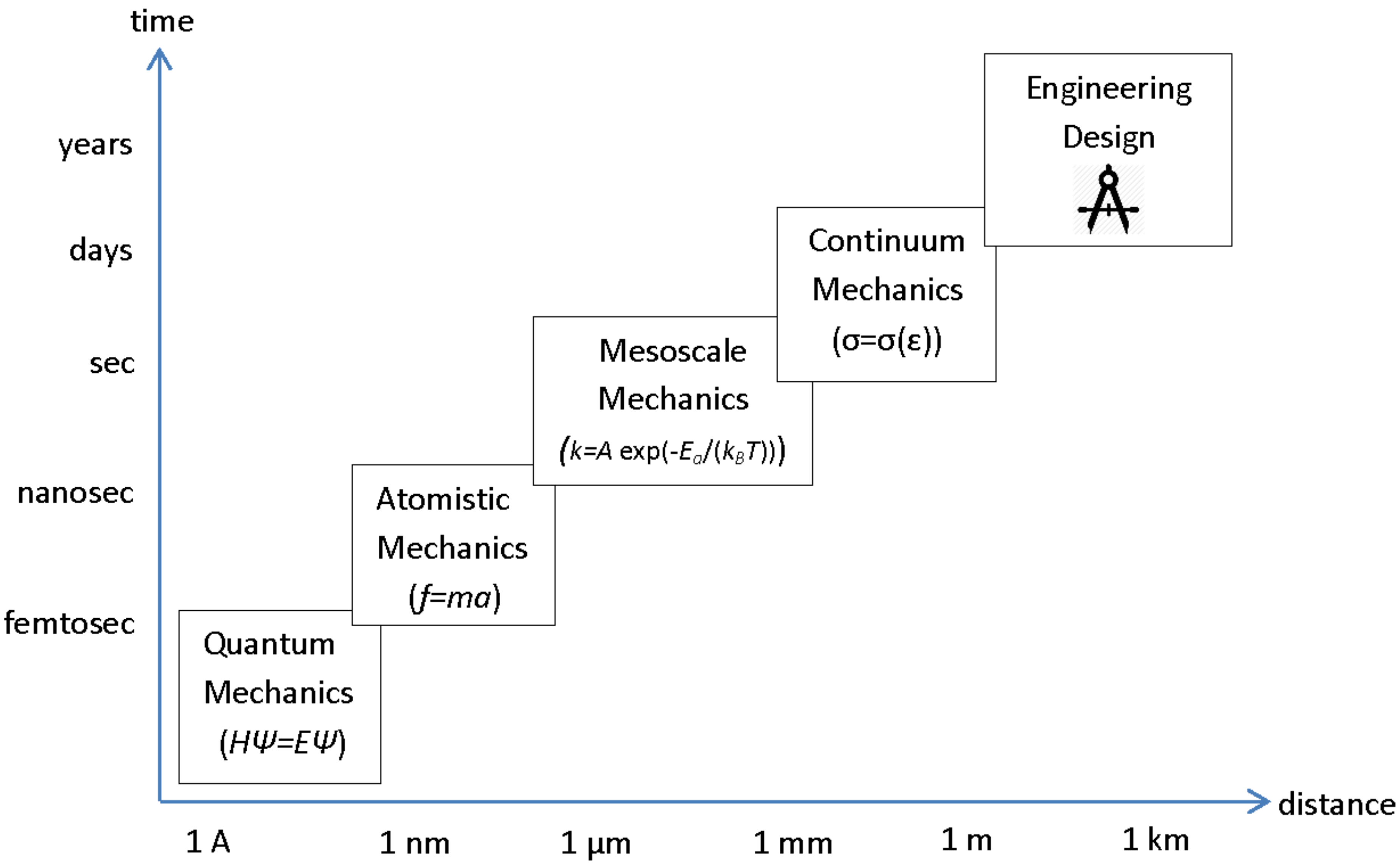
Solution: Active learning / Learning on-the-fly



Overview

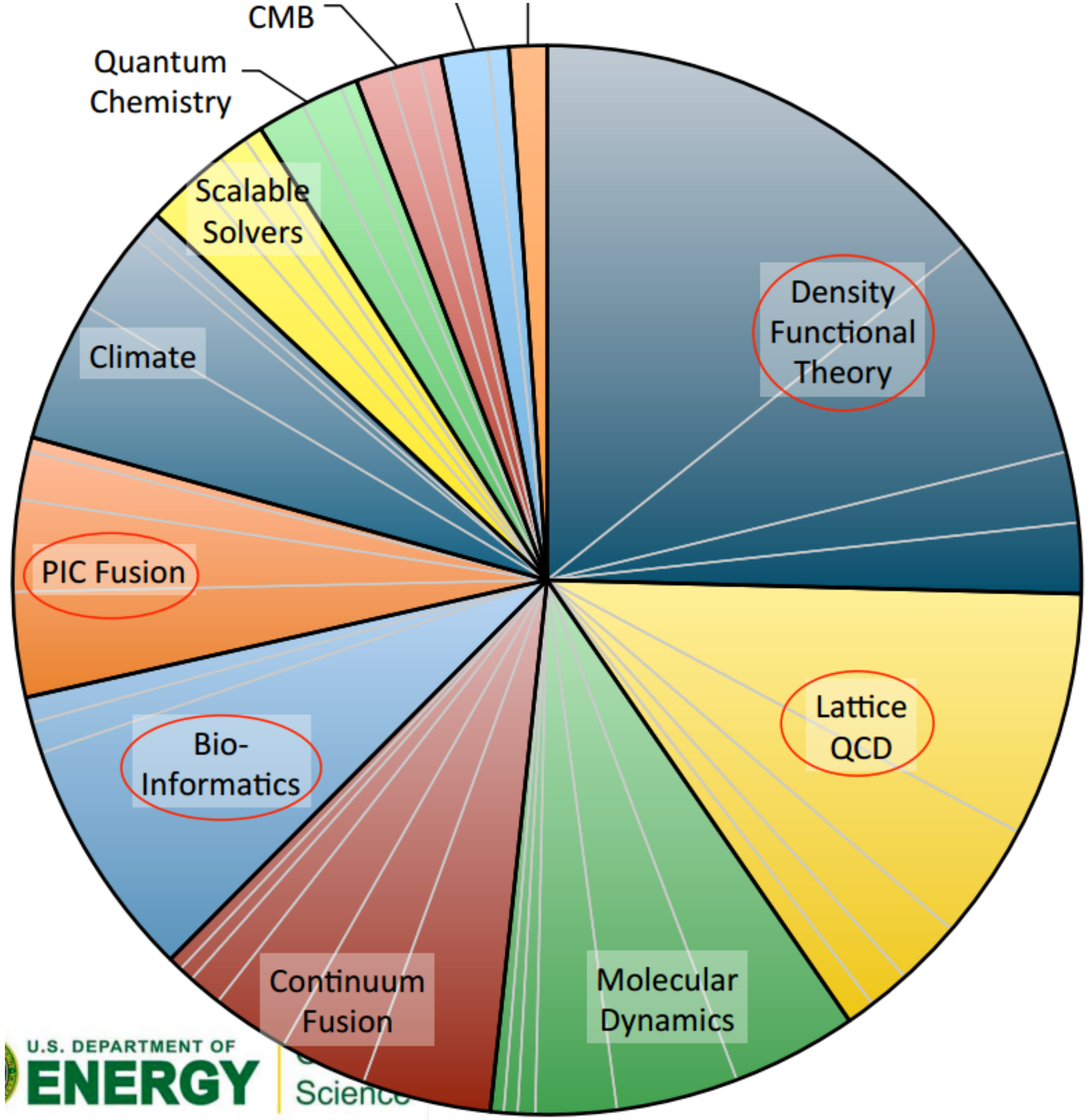
1. **Overview**
    - **Why is this important?**
2. Moment Tensor Potentials
3. Active learning (how to learn while sampling a PES)
4. Applications

A dream of comp.mater.sci


time
years
days
sec
nanosec
femtosec
Quantum Mechanics ($H\Psi=E\Psi$)
Atomistic Mechanics ($f=ma$)
Mesoscale Mechanics ($k=A\exp(-E_a/(k_BT))$)
Continuum Mechanics ($\sigma=\sigma(\varepsilon)$)
Engineering Design
distance
1 A
1 nm
1 µm
1 mm
1 m
1 km

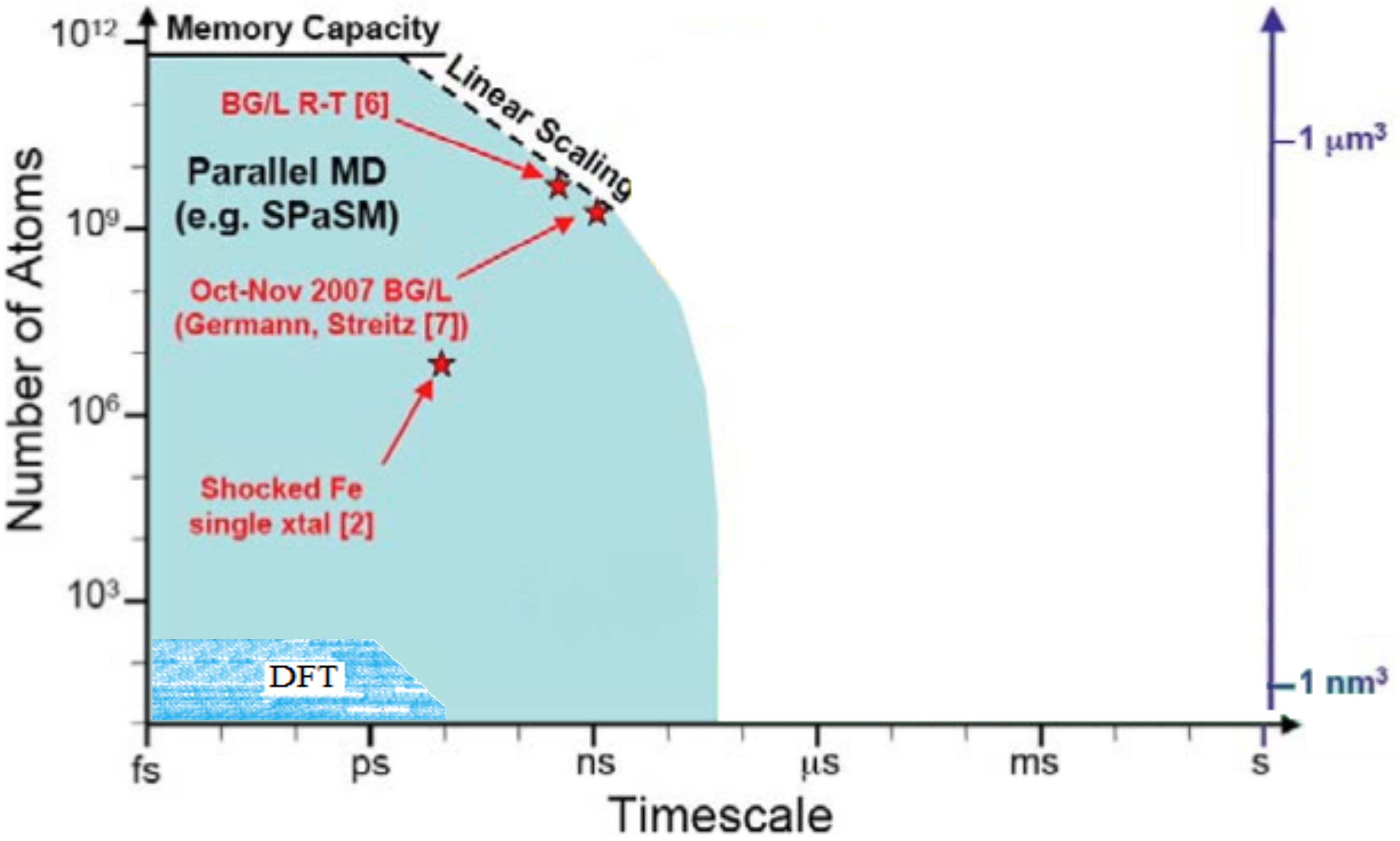
Molecular modeling

- ~40% of supercomputing time is spent on Molecular Modeling



[Adopted from nersc.gov]

Molecular dynamics scales



[D. Perez, LANL]

Overview

1. **Overview**
2. **Machine-learning Potentials**
3. Active learning (how to learn while sampling a PES)
4. Applications

Regression of Atomistic Properties

Basic problem:

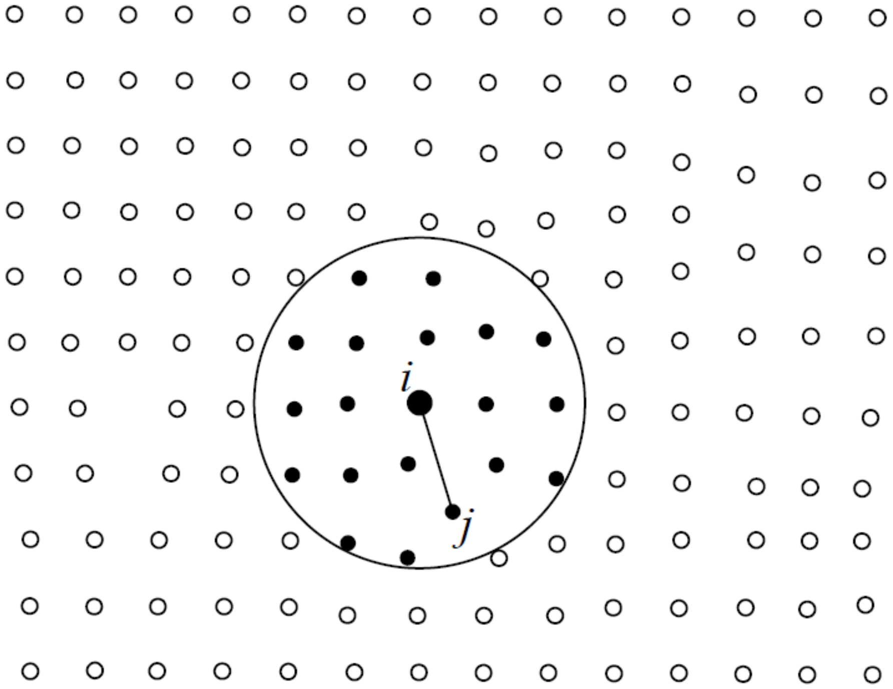
- Given a molecule/atomic system $\boldsymbol{r}_i$ predict its property $\boldsymbol{F}(\boldsymbol{r}_i)$.
- Often, one does want to learn the physical symmetries (or, more generally physical properties), instead embed into the model.

- For interatomic potentials Step 1 is to use locality

Locality: Energy

$$E = \sum_i V(r_{i1}, r_{i2}, \dots)$$

- Most interatomic potentials are covered. (Coulomb should be added explicitly.)

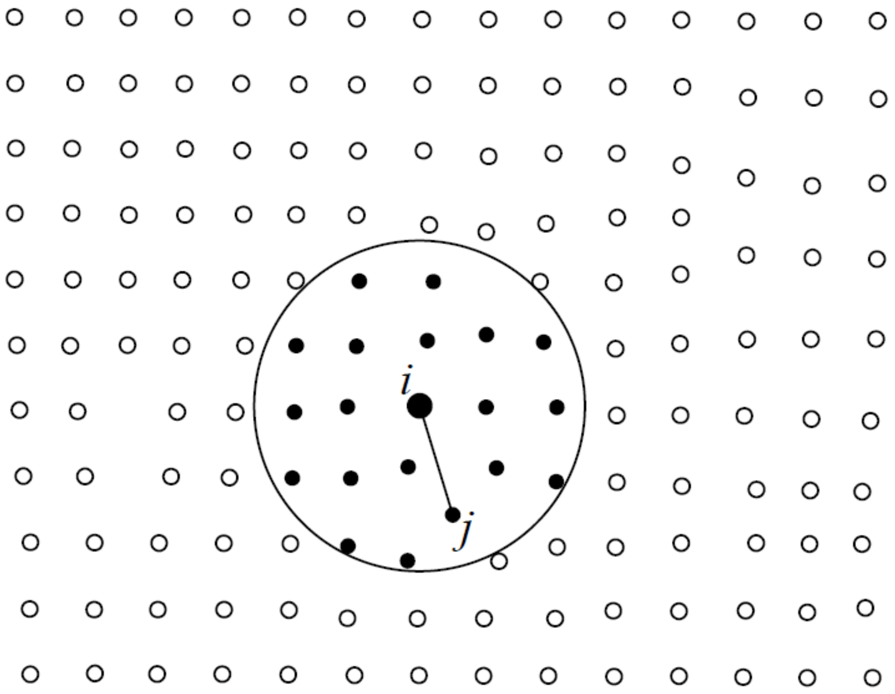


- Problem: find a good *V.*
- This step gives us:
  - Transferability wrt number of atoms (can apply to systems with millions of atoms)
  - translation invariance

Locality: Energy

$$E = \sum_i V(r_{i1}, r_{i2}, \dots)$$

- Most interatomic potentials are covered. (Coulomb should be added explicitly.)



- Problem: find a good *V.*
- This step gives us:
  - Transferability wrt number of atoms (can apply to systems with millions of atoms)
  - translation invariance
- To do next: rotation and permutation invariance

Rotation and permutation invariance:

By far most popular solution is to design descriptors of atomic environments

- Behler-Parrinello descriptors:

$D^{(2)}(\boldsymbol{r}_{i\cdot}) = \sum_j f\left(r_{ij}\right)$ for some scalar function f

$D^{(3)}(\boldsymbol{r}_{i\cdot}) = \sum_j \sum_k f\left(r_{ij}\right) f(r_{ik})\, \varphi\left(r_{ij} \cdot r_{ik}\right)$ for some scalar f and φ

For long time it was considered that they could generate a complete description of atomic environment, but recently it was proved that this is false: https://arxiv.org/pdf/2001.11696.pdf

Regression: Neural networks

- Problem: given a vector of descriptors $v_1, \dots, v_M$, find the mapping $F = F(v_1, \dots, v_M)$
- Machine-learning approach: find $F = F(v_1, \dots, v_M)$ from data by fitting some parameters
- Two-level Neural network:

$$F(\boldsymbol{v}) = A_2 f(A_1\, \boldsymbol{v} + \boldsymbol{b}_1) + \boldsymbol{b}_2,$$

Where matrices A_1, A_2 and vectors $\boldsymbol{b}_1, \boldsymbol{b}_2$ are found from data

Alternative: Gaussian process regression

- $F(\boldsymbol{r}_{i\cdot}) = \sum_l k\left(\boldsymbol{r}_{i\cdot}, \boldsymbol{r}_{i\cdot}^{(l)}\right)$, where *k* is a kernel giving a similarity measure between the given atomic environment $\boldsymbol{r}_{i\cdot}$ and those from the training(=fitting) set $\boldsymbol{r}_{i\cdot}^{(l)}$.
- The problem reduces to designing a kernel satisfying physical symmetries

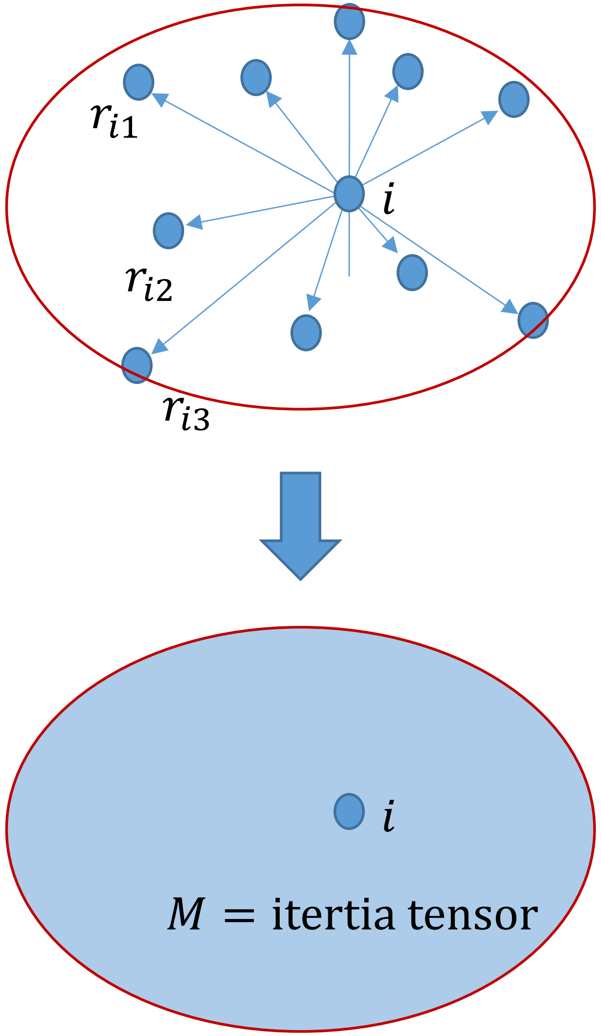
Alternative: Moment Tensor Potentials

Descriptors of atomic environments:

- Moments of inertia of surrounding atoms
- They satisfy the needed symmetries (rotation, permutation, translation, …);
- **Math:**
- $M_{n,m}(\boldsymbol{r}_{\boldsymbol{i}\cdot}) = \sum_j \boxed{f_n(|r_{ij}|)}\,\boxed{\underbrace{r_{ij} \otimes \cdots \otimes r_{ij}}_{m \text{ times}}}$

Radial term: extracting shells of neighboring atoms

Angular term: shell orientations



Moment Tensor Potentials, basis functions

- $V(\boldsymbol{u};\theta) = \sum_\alpha \theta_\alpha B_\alpha(\boldsymbol{u})$
- $B_\alpha(\mathbf{u})$ are (all) different multiplications (contractions) of inertia tensors $M_{m,n}(\mathbf{u})$ yielding a scalar.

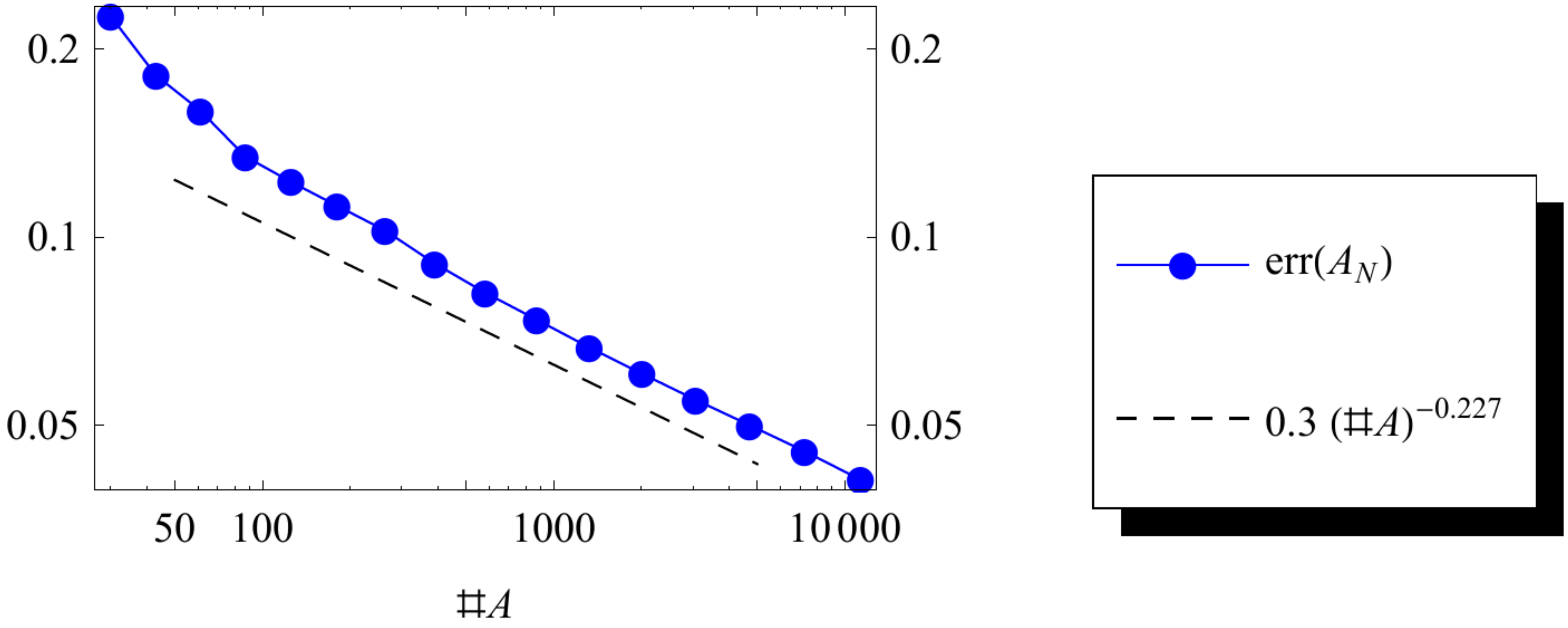
Theorem:

- $B_\alpha(\mathbf{u})$ is a complete basis

Learning curves

Database (Csanyi, Bartok, Szlachta, 2014)

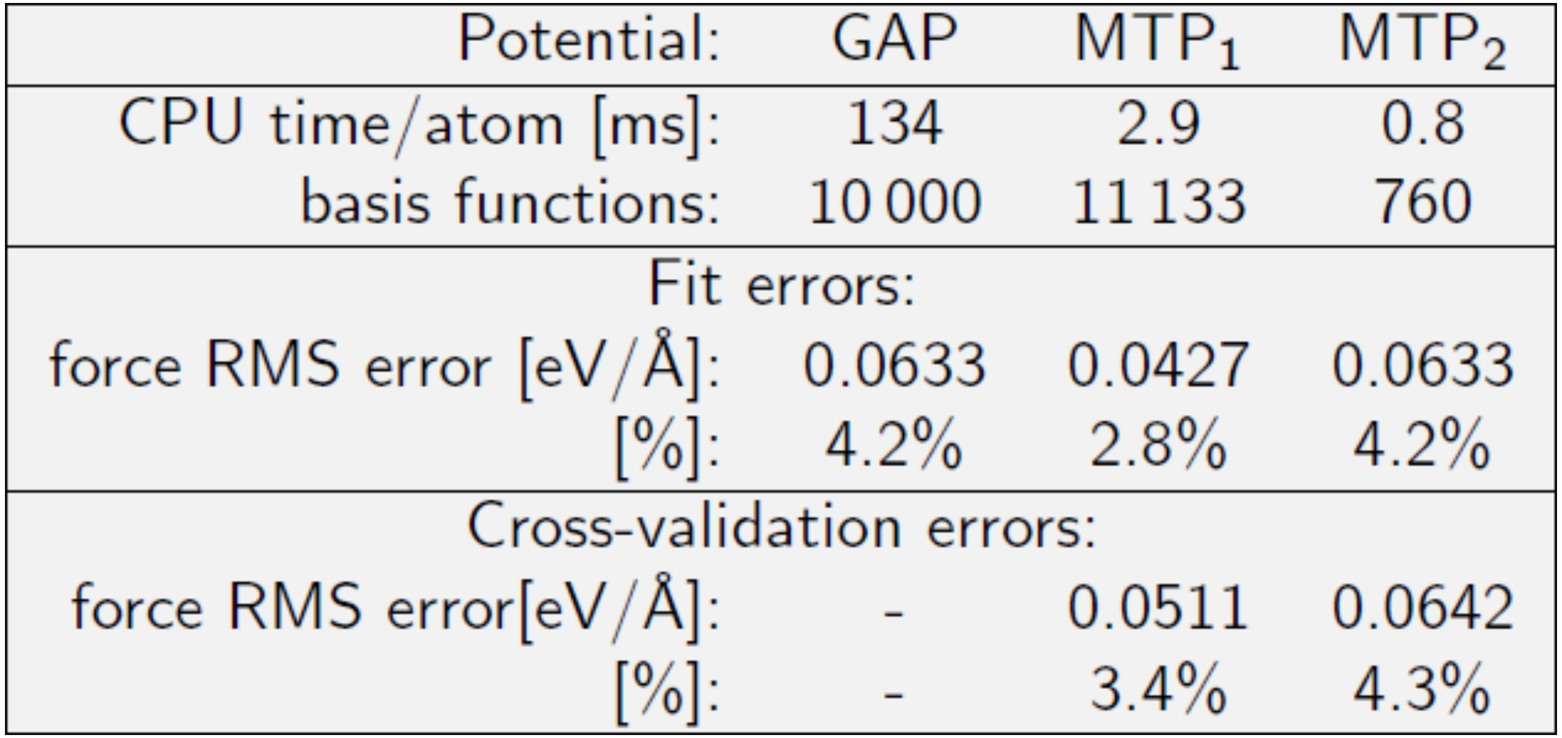
- Tungsten: uniform and perturbed lattices, vacancies, dislocations



Performance tests

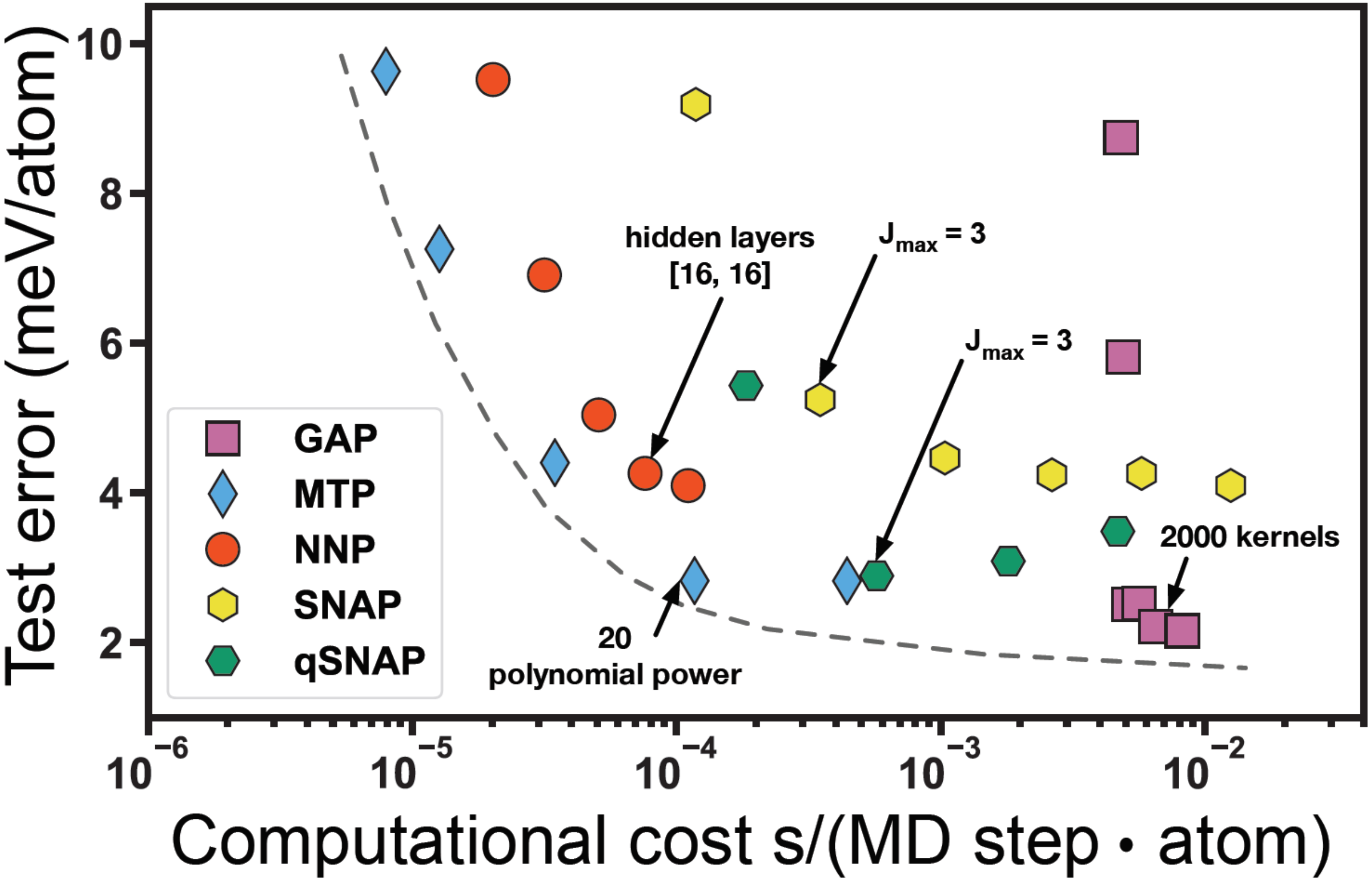
Database (Csanyi, Bartok, Szlachta, 2014)

- Tungsten: uniform and perturbed lattices, vacancies, dislocations



Potential:	GAP	MTP_1	MTP_2
CPU time/atom [ms]:	134	2.9	0.8
basis functions:	10 000	11 133	760
Fit errors:			
force RMS error [eV/Å]:	0.0633	0.0427	0.0633
[%]:	4.2%	2.8%	4.2%
Cross-validation errors:			
force RMS error[eV/Å]:	-	0.0511	0.0642
[%]:	-	3.4%	4.3%

Comparison with more methods



Yunxing Zuo,
Chi Chen,
Xiangguo Li,
Zhi Deng,
Yiming Chen,
Jörg Behler,
Gábor Csányi,
A.S.,
Aidan P. Thompson,
Mitchell A. Wood,
Shyue Ping Ong.
arXiv:1906.08888

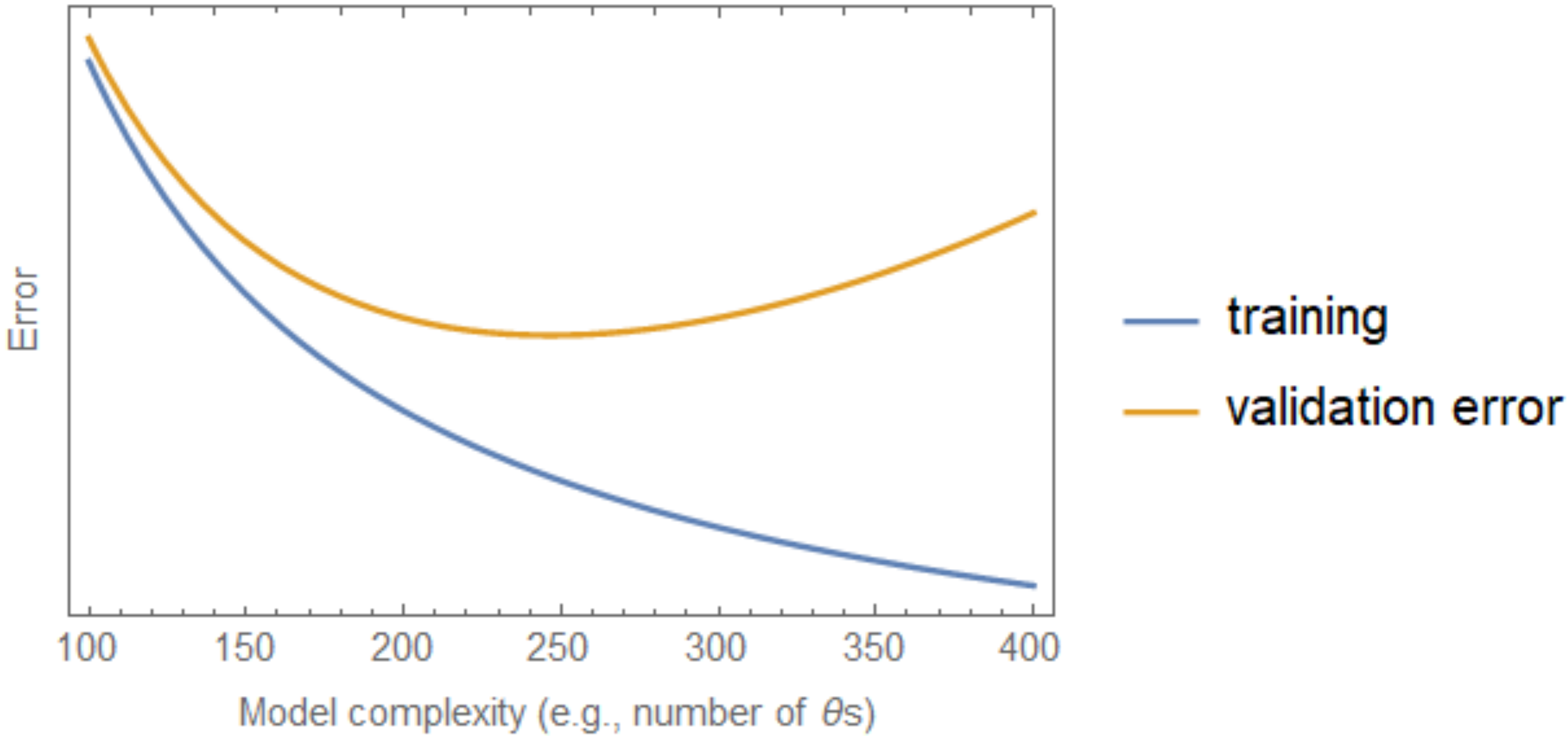
Training and Validation

Often, to test the quality of the potential we split training and validation:

- Machine-learning model: $E = E(\boldsymbol{\theta}, \boldsymbol{x})$
- Training set: $\boldsymbol{x}_{\mathrm{tr}}^{(1)}, \dots, \boldsymbol{x}_{\mathrm{tr}}^{(N)}$. Training: $\min_{\boldsymbol{\theta}} \sum_k \left(E\left(\boldsymbol{\theta}, \boldsymbol{x}_{\mathrm{tr}}^{(k)}\right) - E_{\mathrm{tr}}^{(k)}\right)^2$
- Validation set: $\boldsymbol{x}_{\mathrm{vld}}^{(1)}, \dots, \boldsymbol{x}_{\mathrm{vld}}^{(N)}$. Validation error:

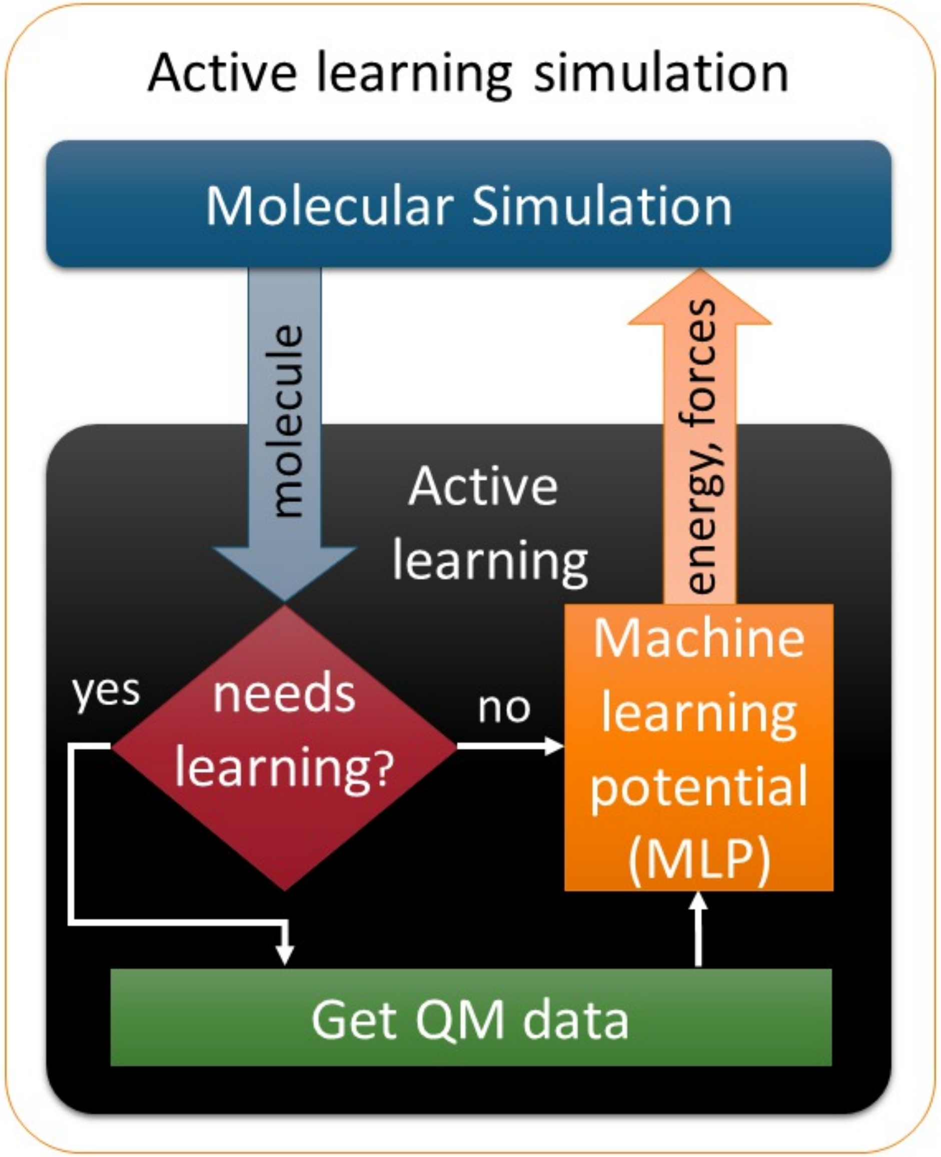
$$\left(\frac{1}{K} \sum_k \left(E\left(\boldsymbol{\theta}, \boldsymbol{x}_{\mathrm{vld}}^{(k)}\right) - E_{\mathrm{vld}}^{(k)}\right)^2\right)^{\frac{1}{2}}$$

Training and Validation



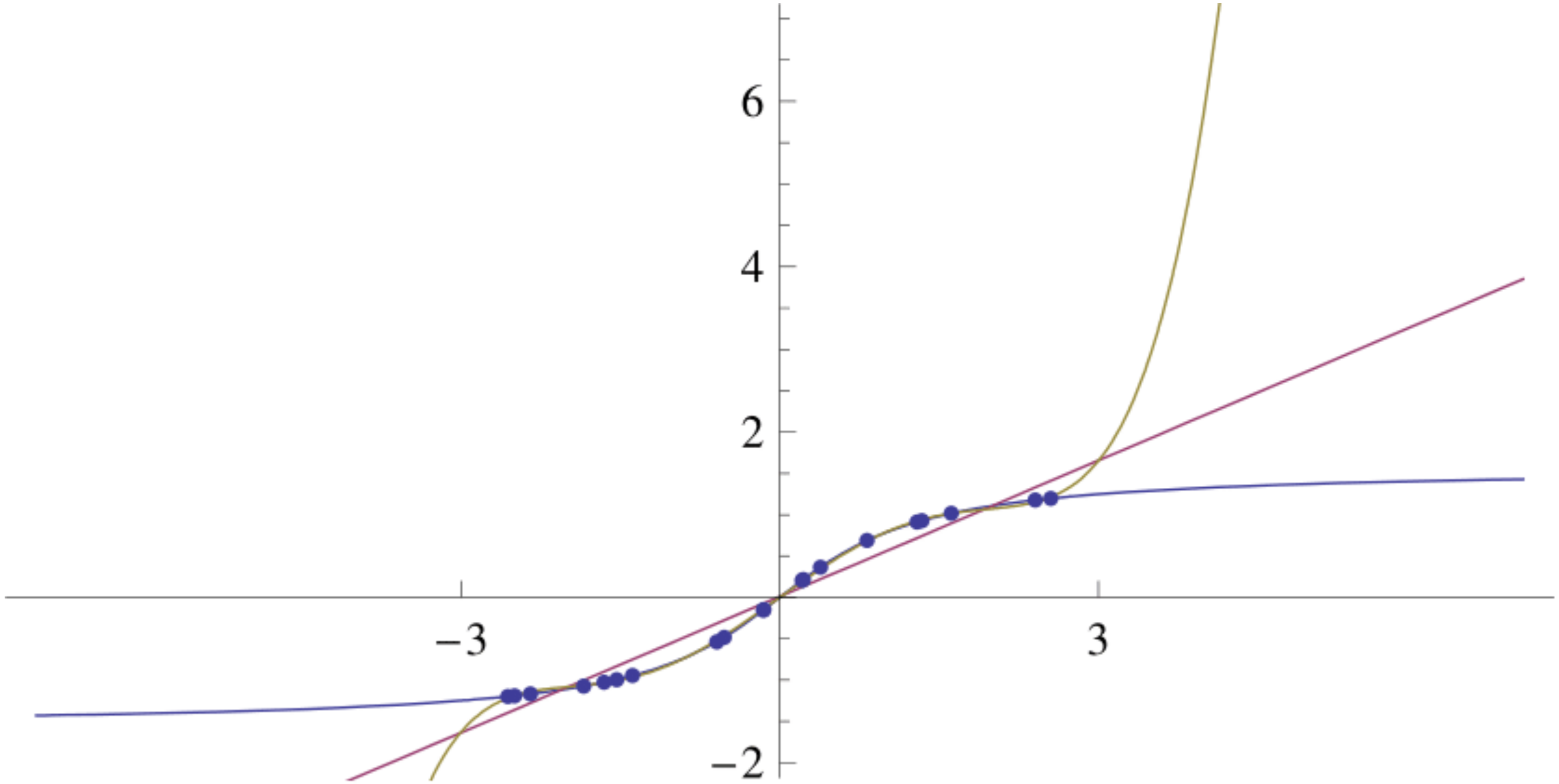
Active Learning of Interatomic Potentials

Active learning



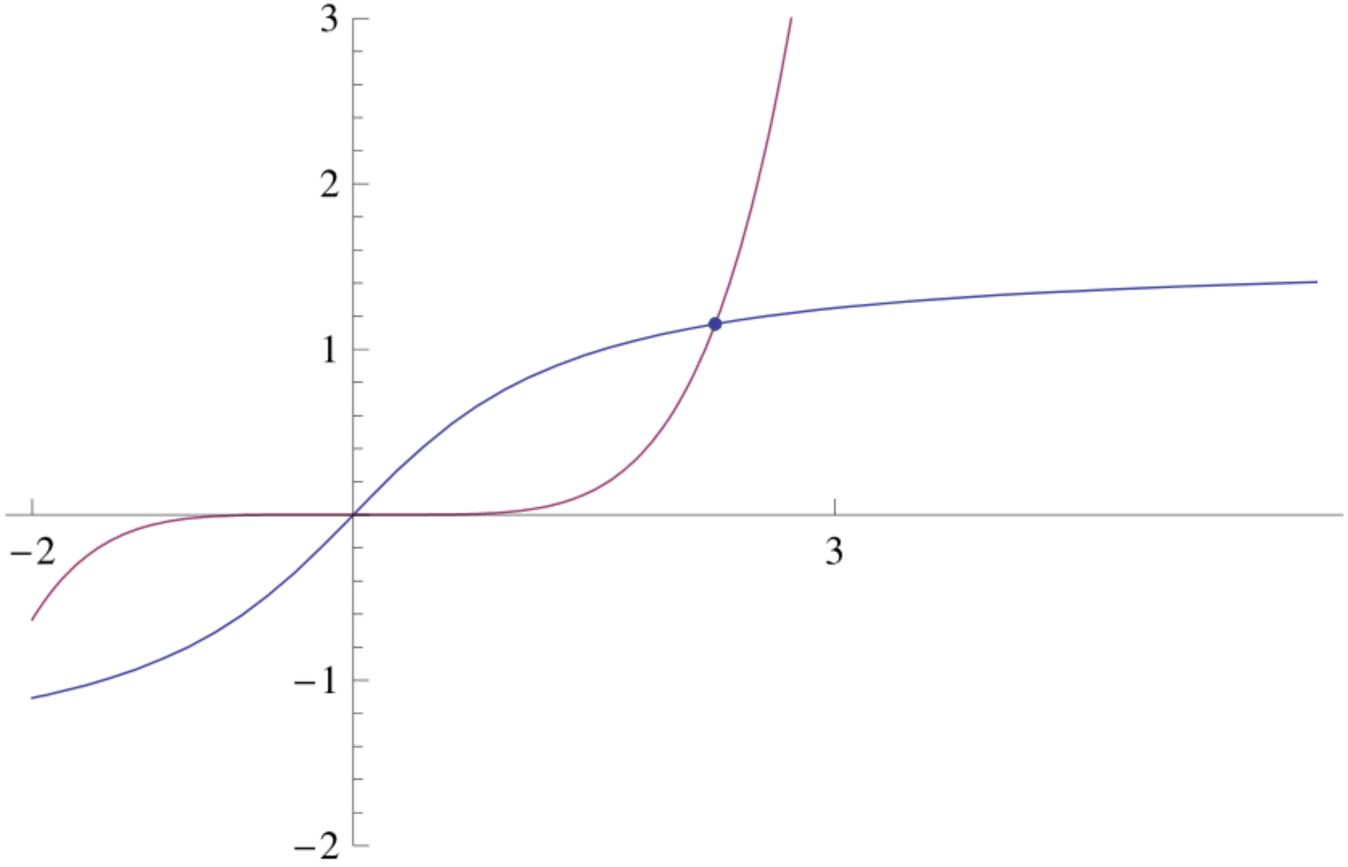
Active Learning of MLIP: Motivation

Higher accuracy => More parameters to fit => Lower transferability



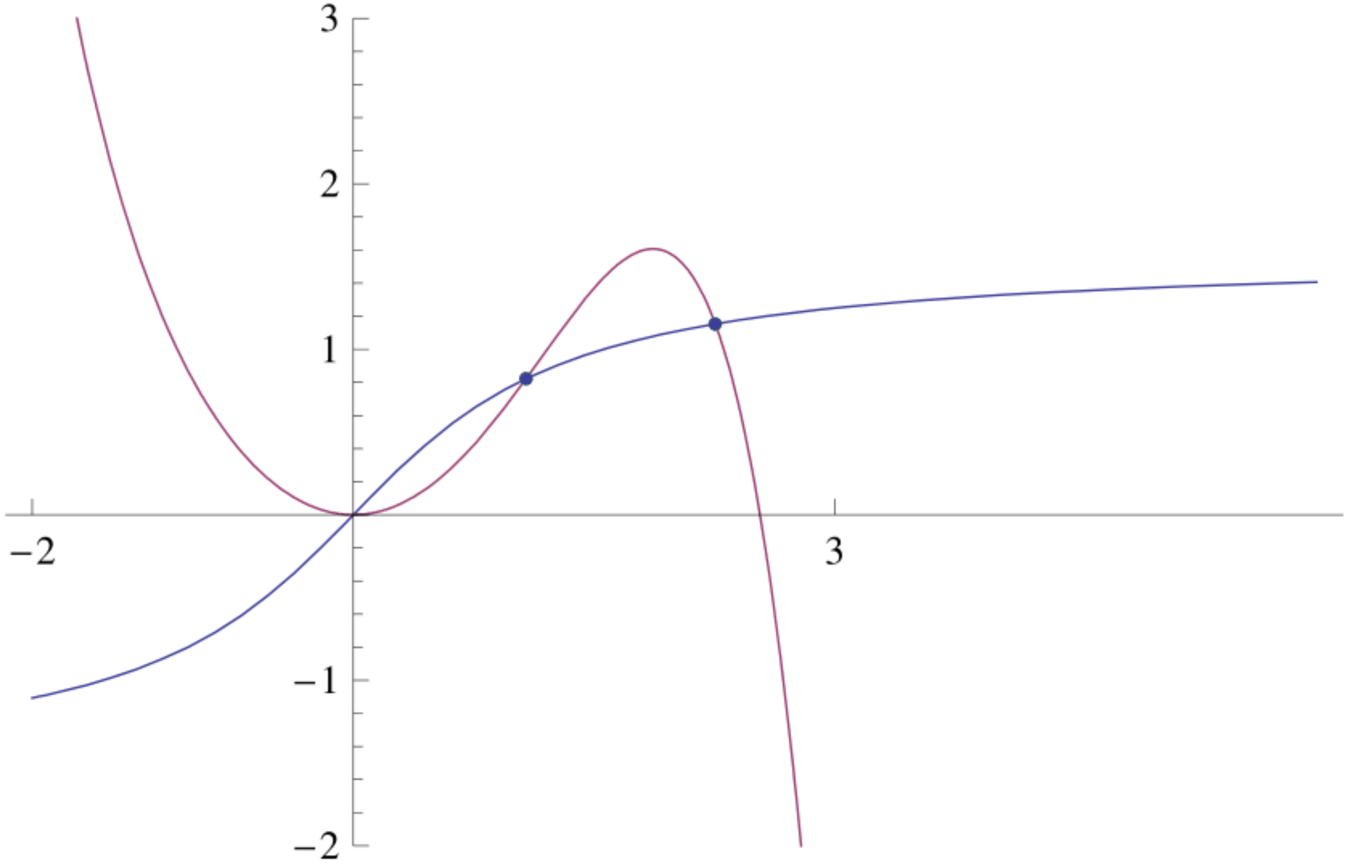
Active learning

Solution: detect when we are extrapolating and switch on learning



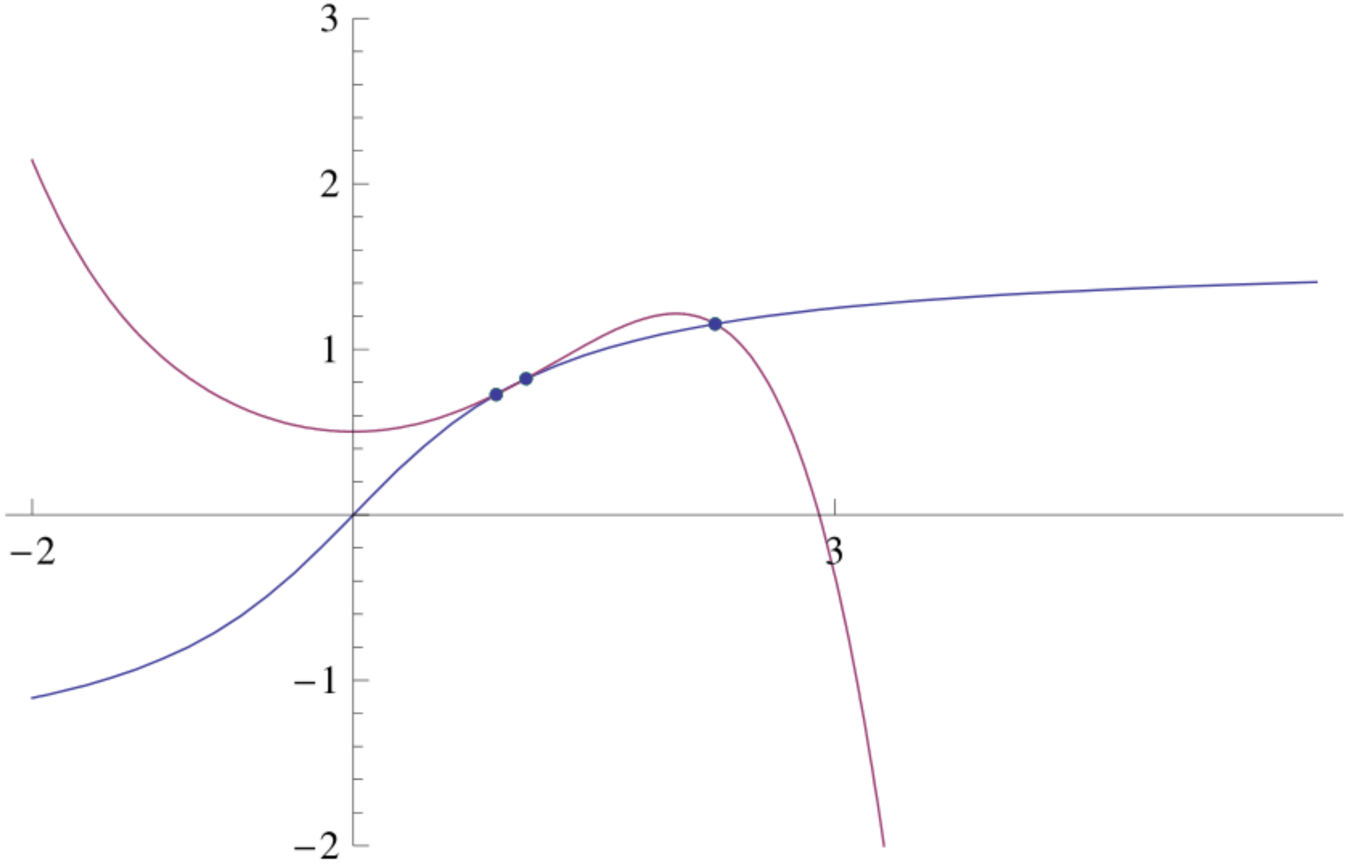
Active learning

Solution: detect when we are extrapolating and switch on learning



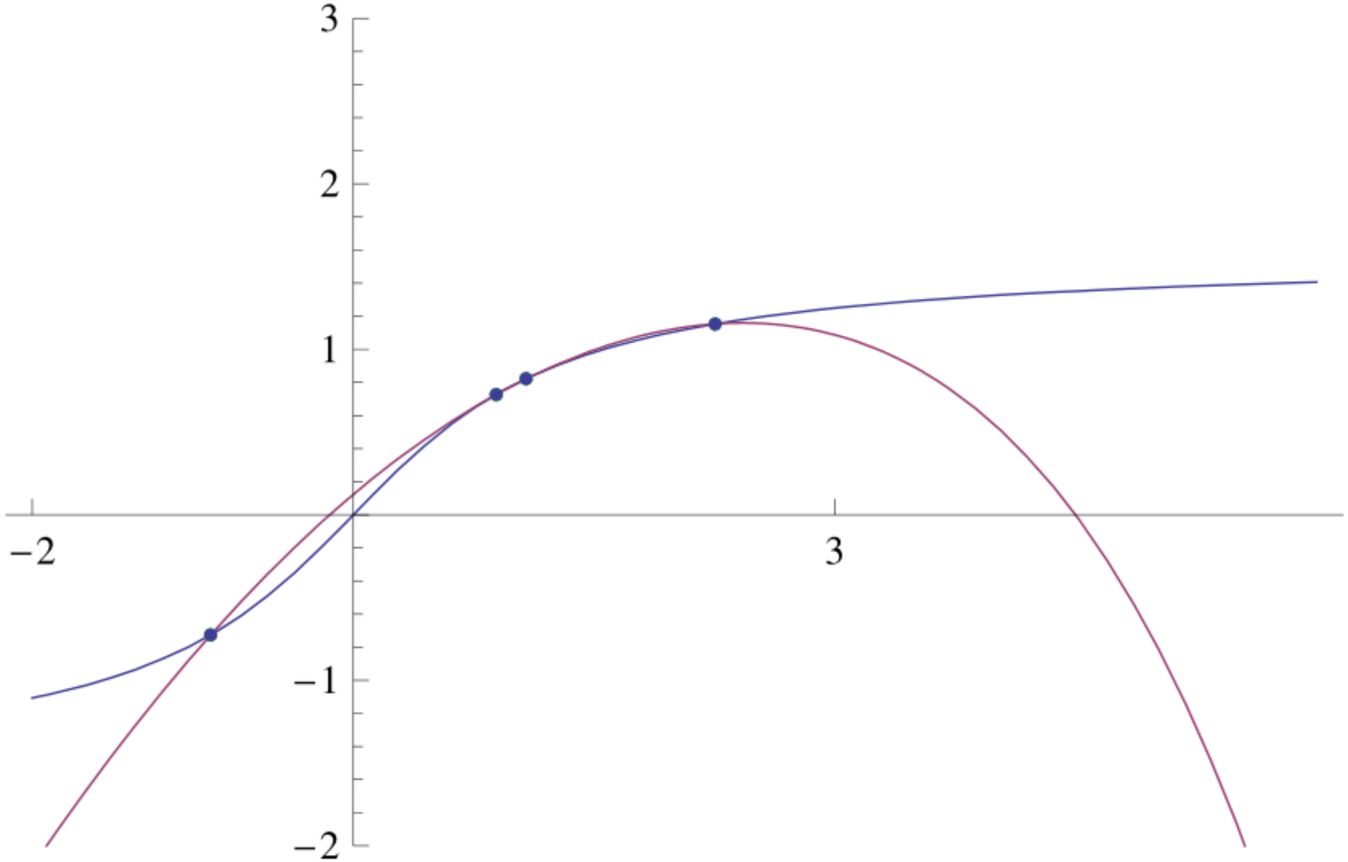
Active learning

Solution: detect when we are extrapolating and switch on learning



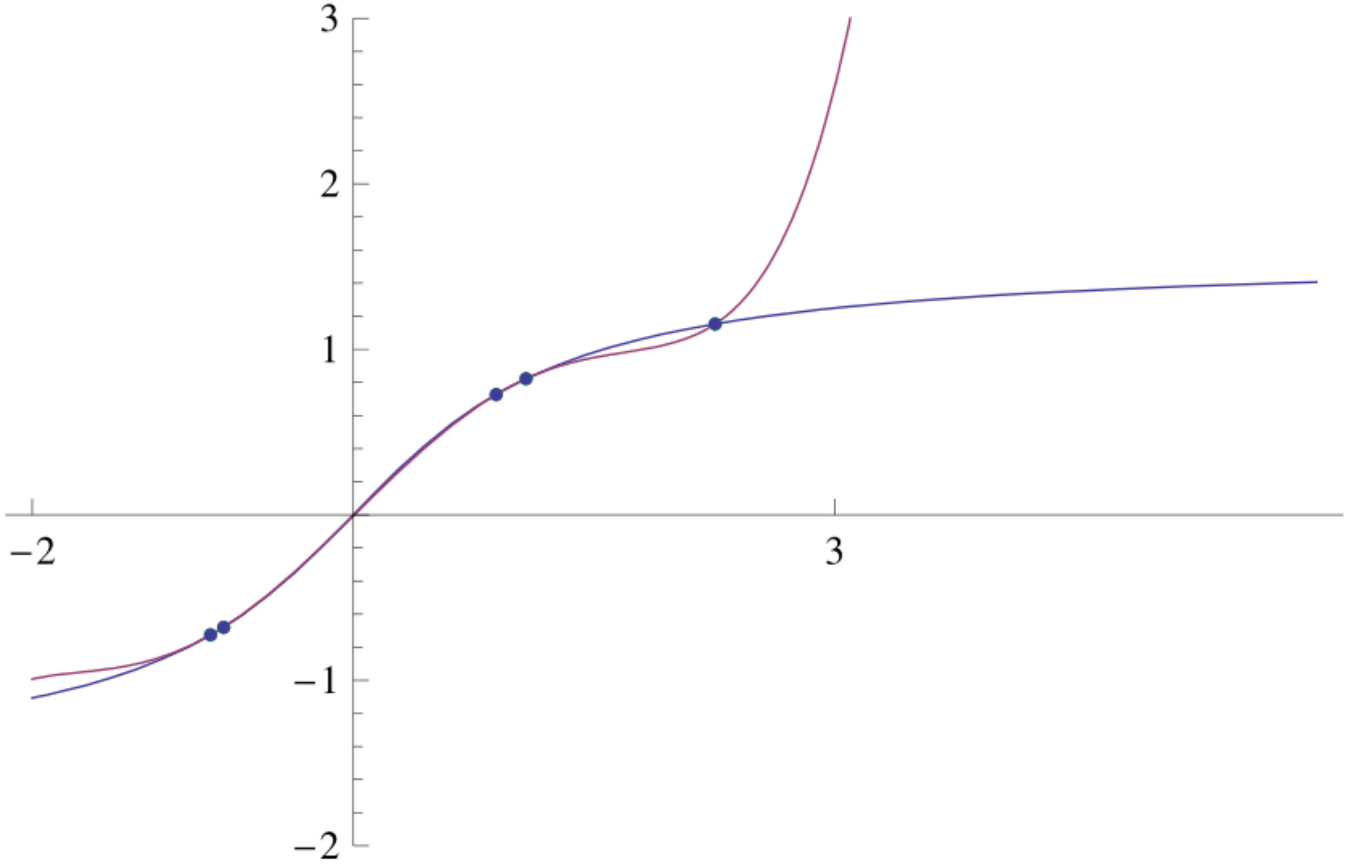
Active learning

Solution: detect when we are extrapolating and switch on learning



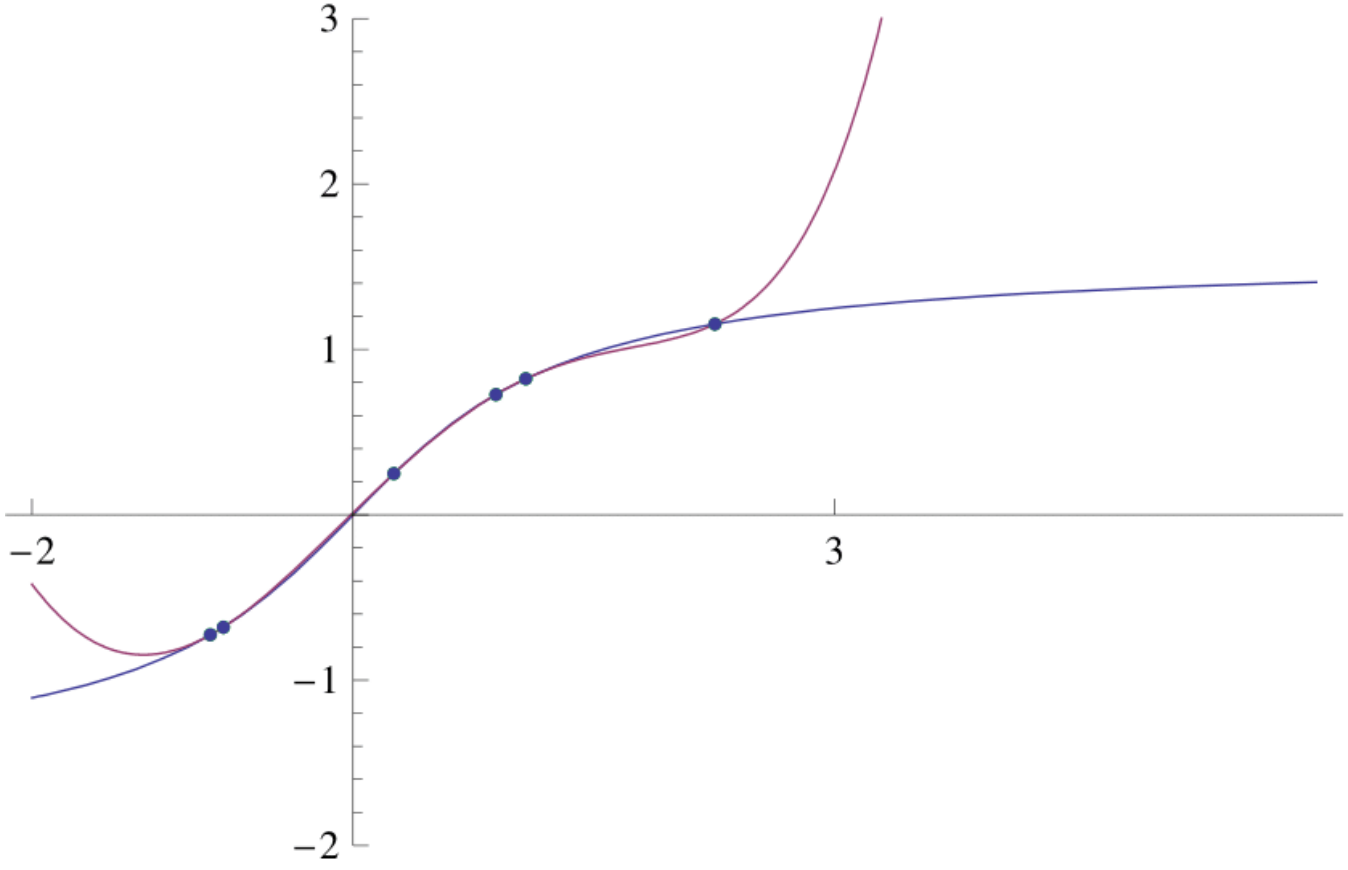
Active learning

Solution: detect when we are extrapolating and switch on learning



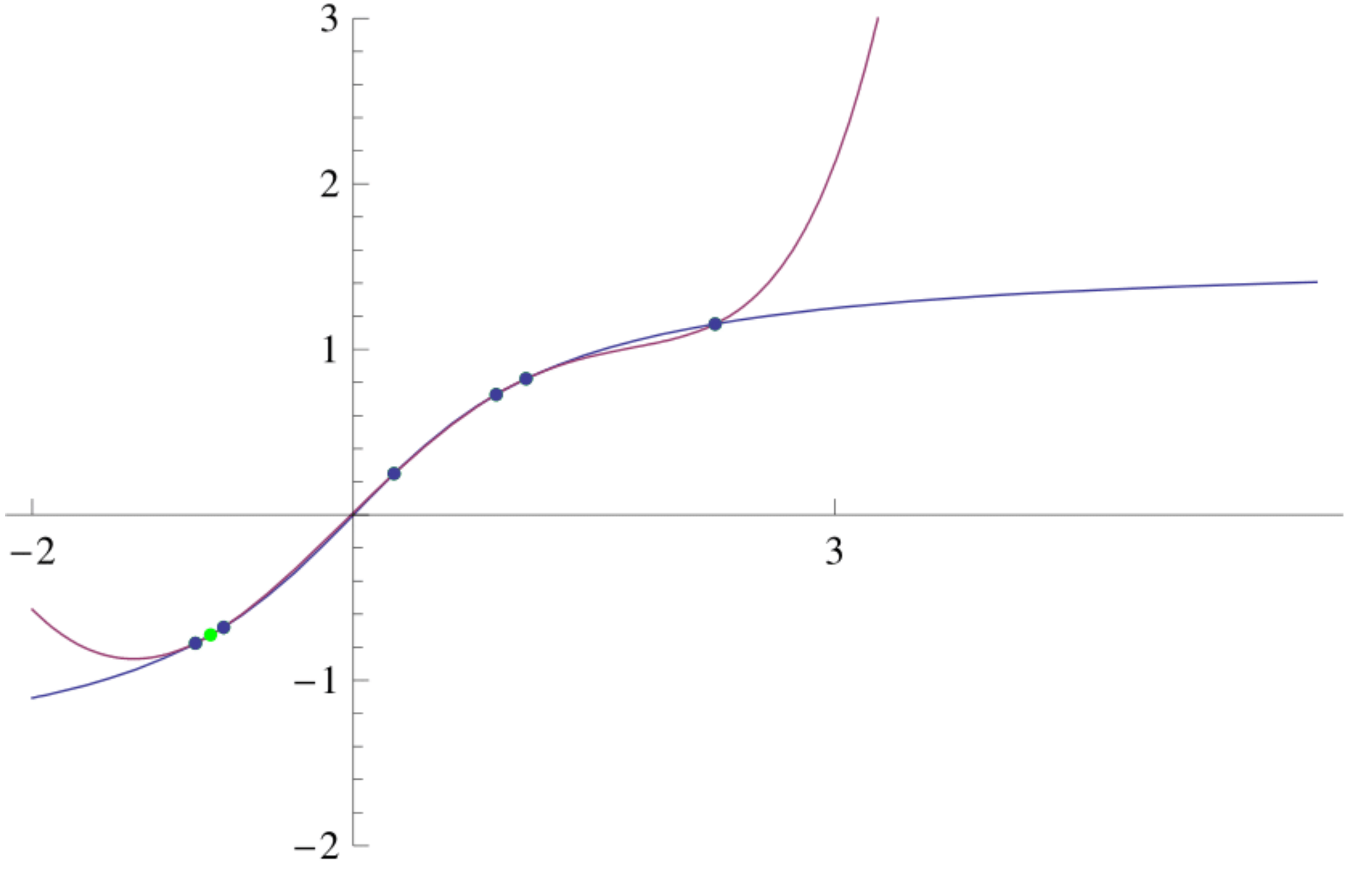
Active learning

Solution: detect when we are extrapolating and switch on learning



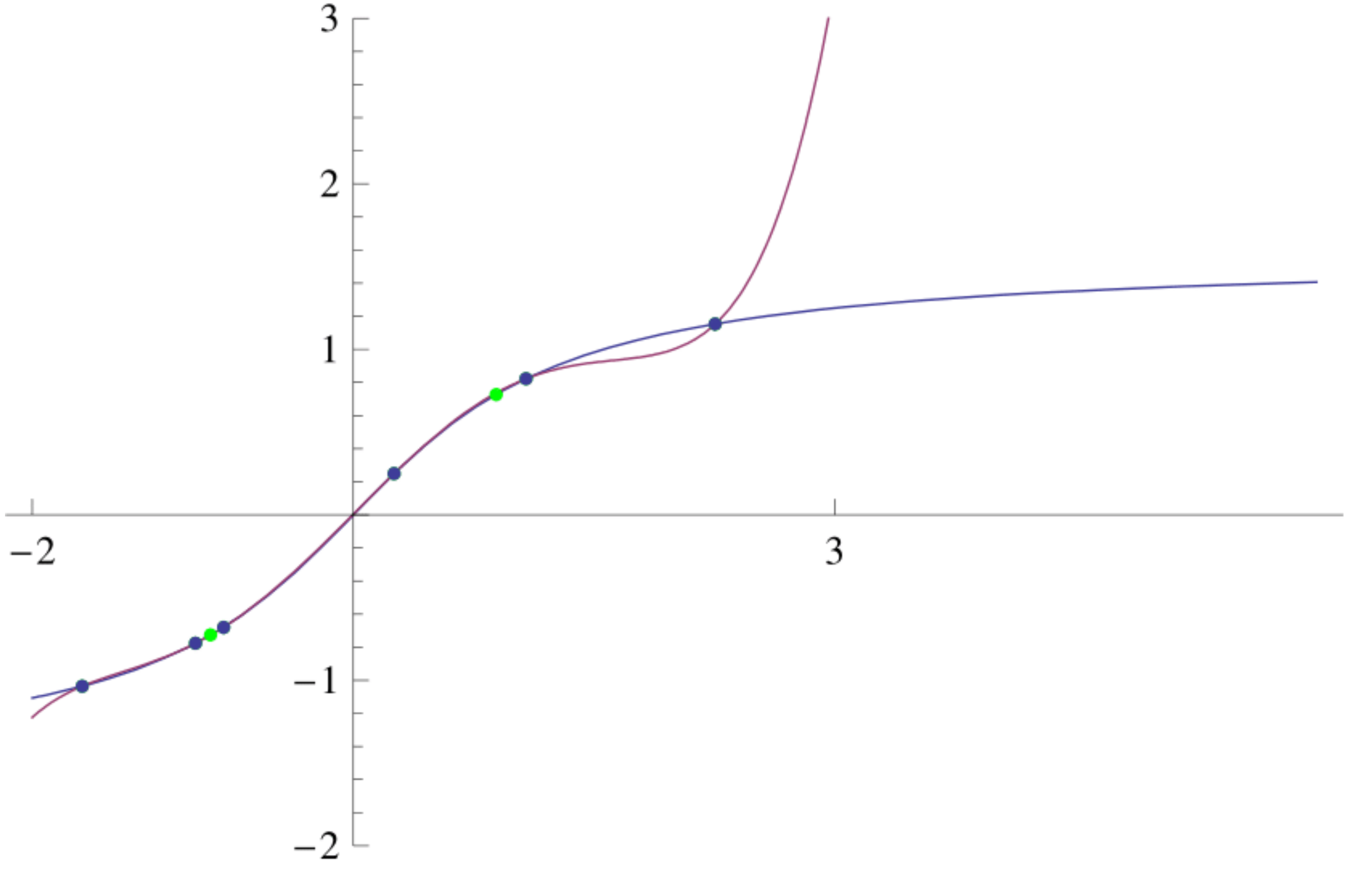
Active learning

Solution: detect when we are extrapolating and switch on learning



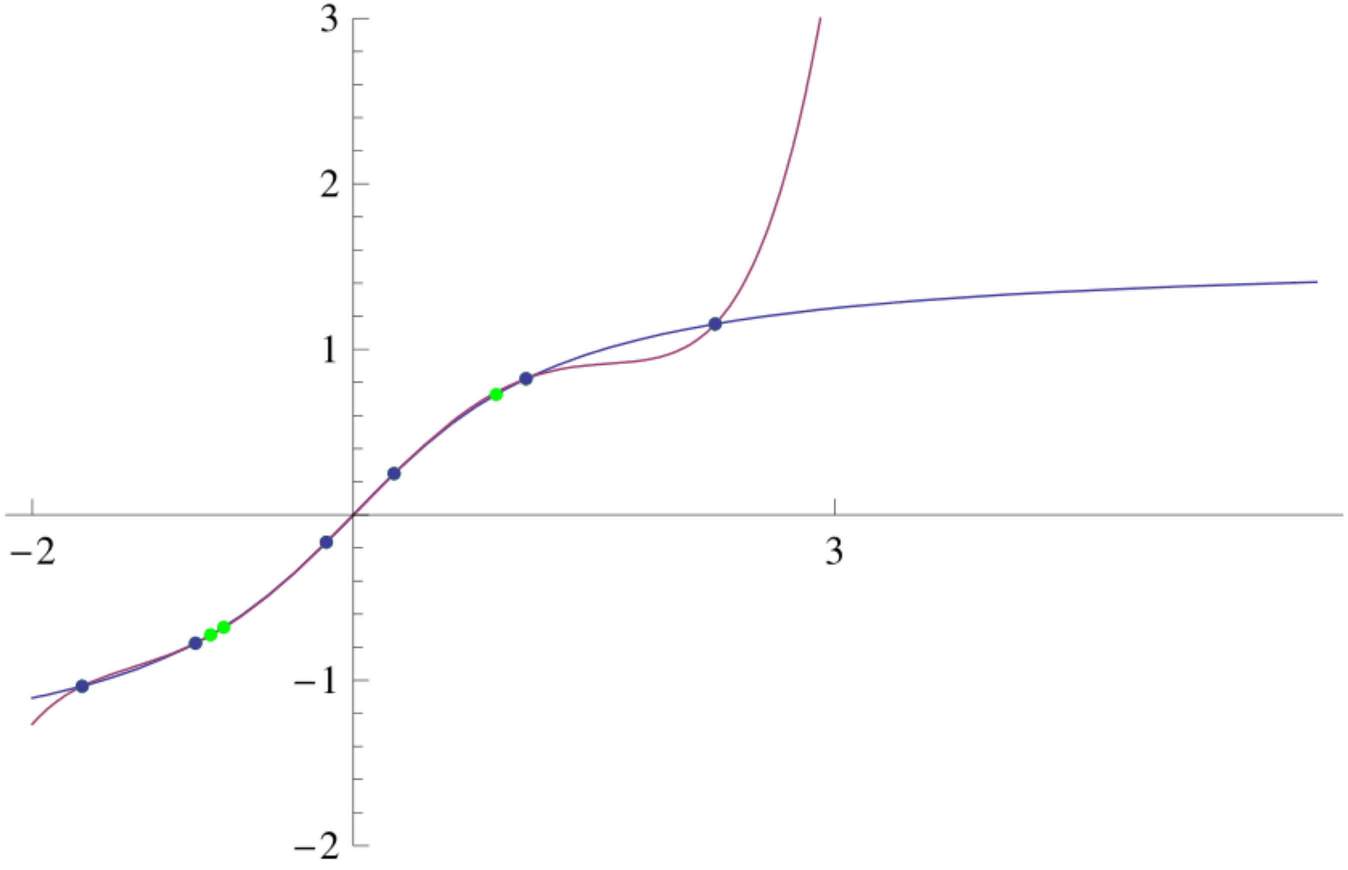
Active learning

Solution: detect when we are extrapolating and switch on learning



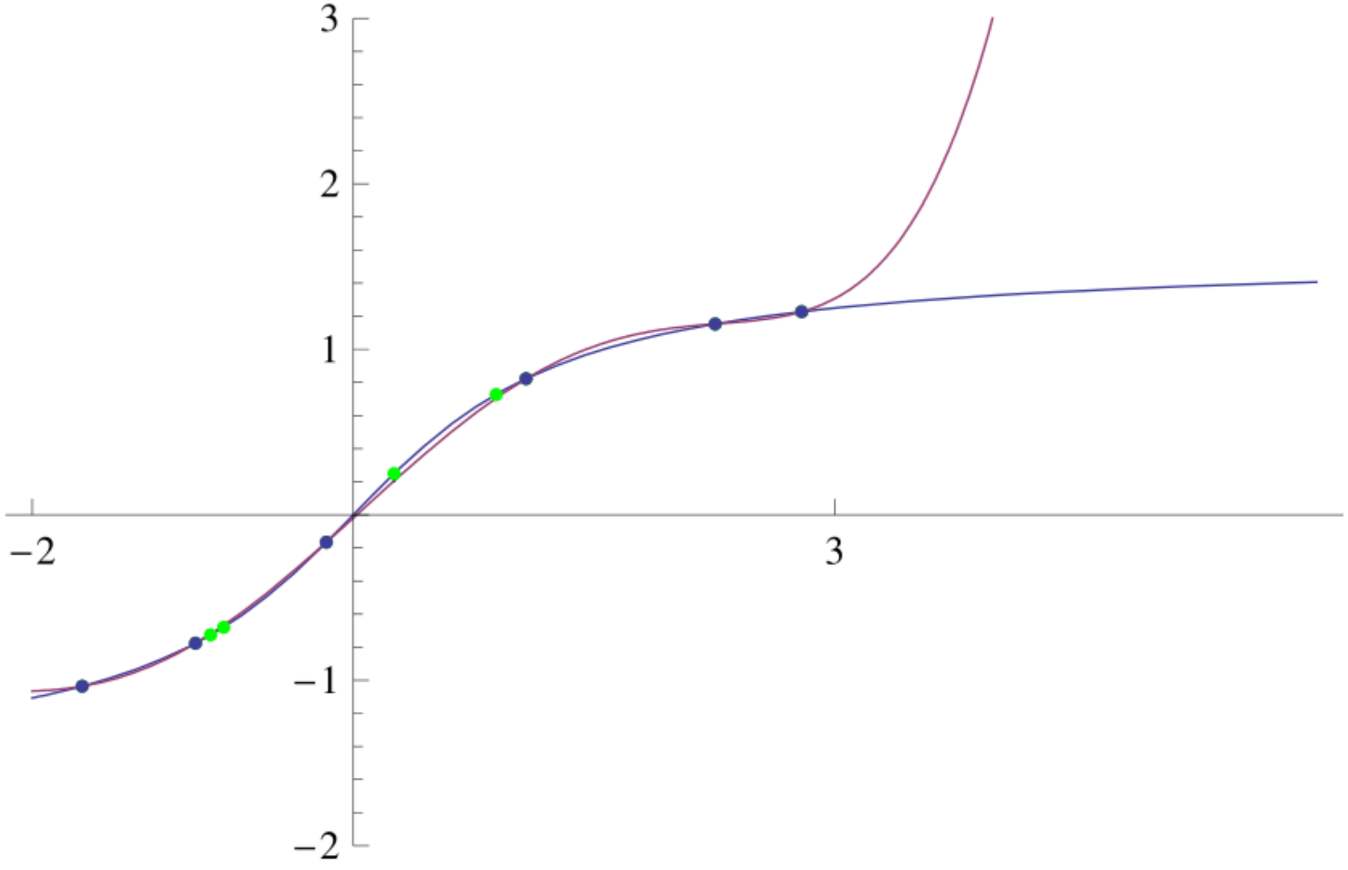
Active learning

Solution: detect when we are extrapolating and switch on learning



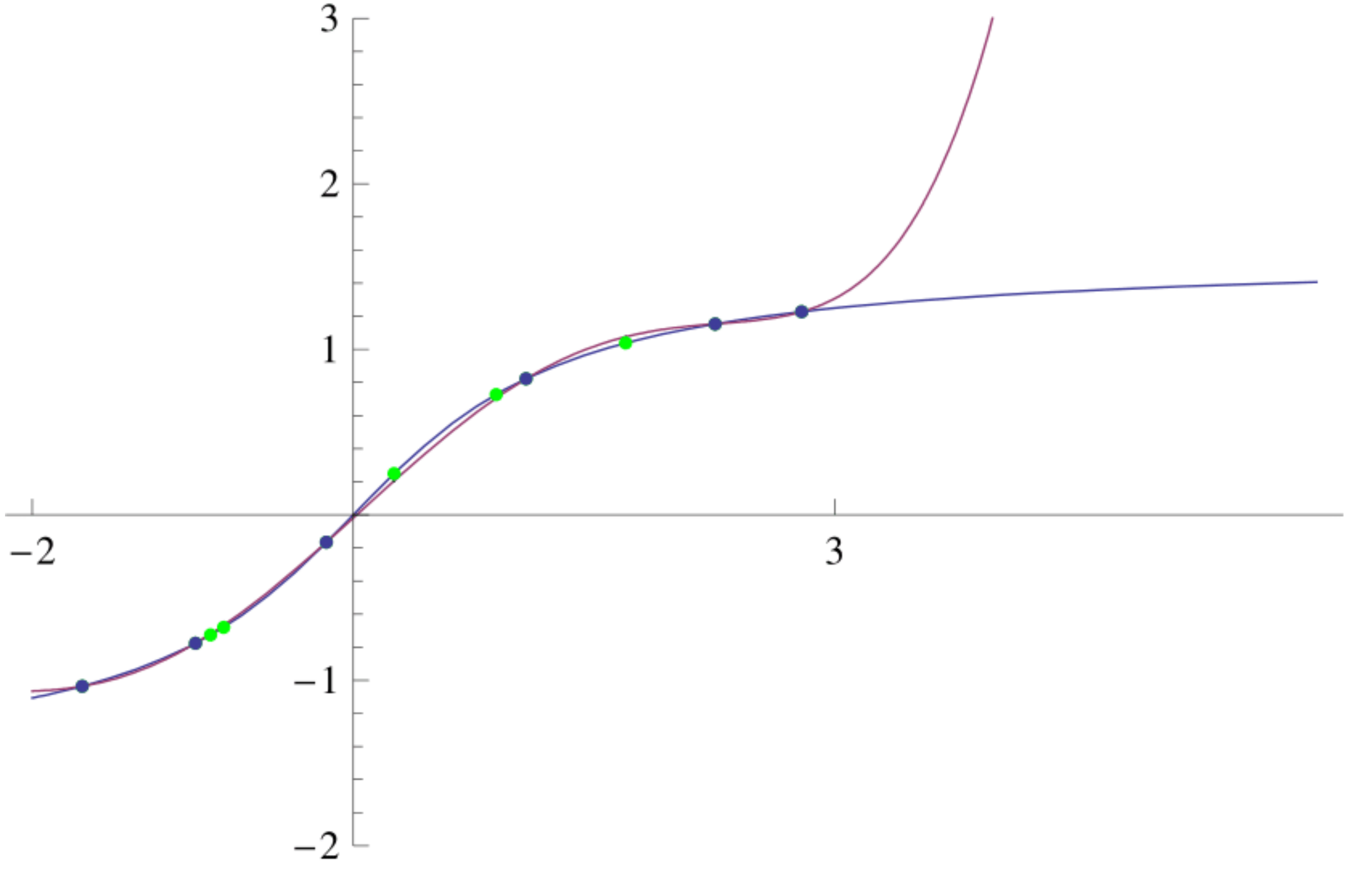
Active learning

Solution: detect when we are extrapolating and switch on learning



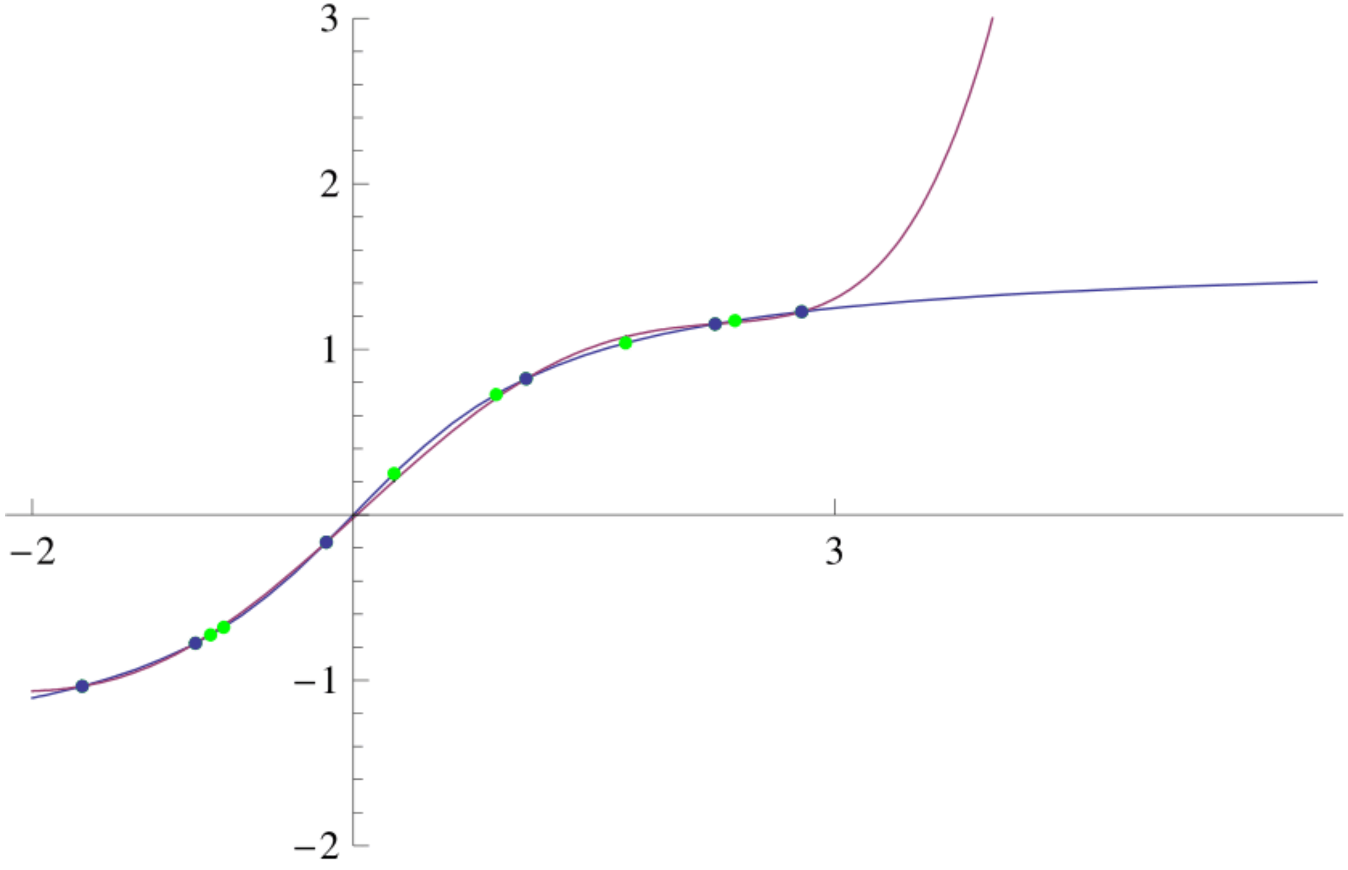
Active learning

Solution: detect when we are extrapolating and switch on learning



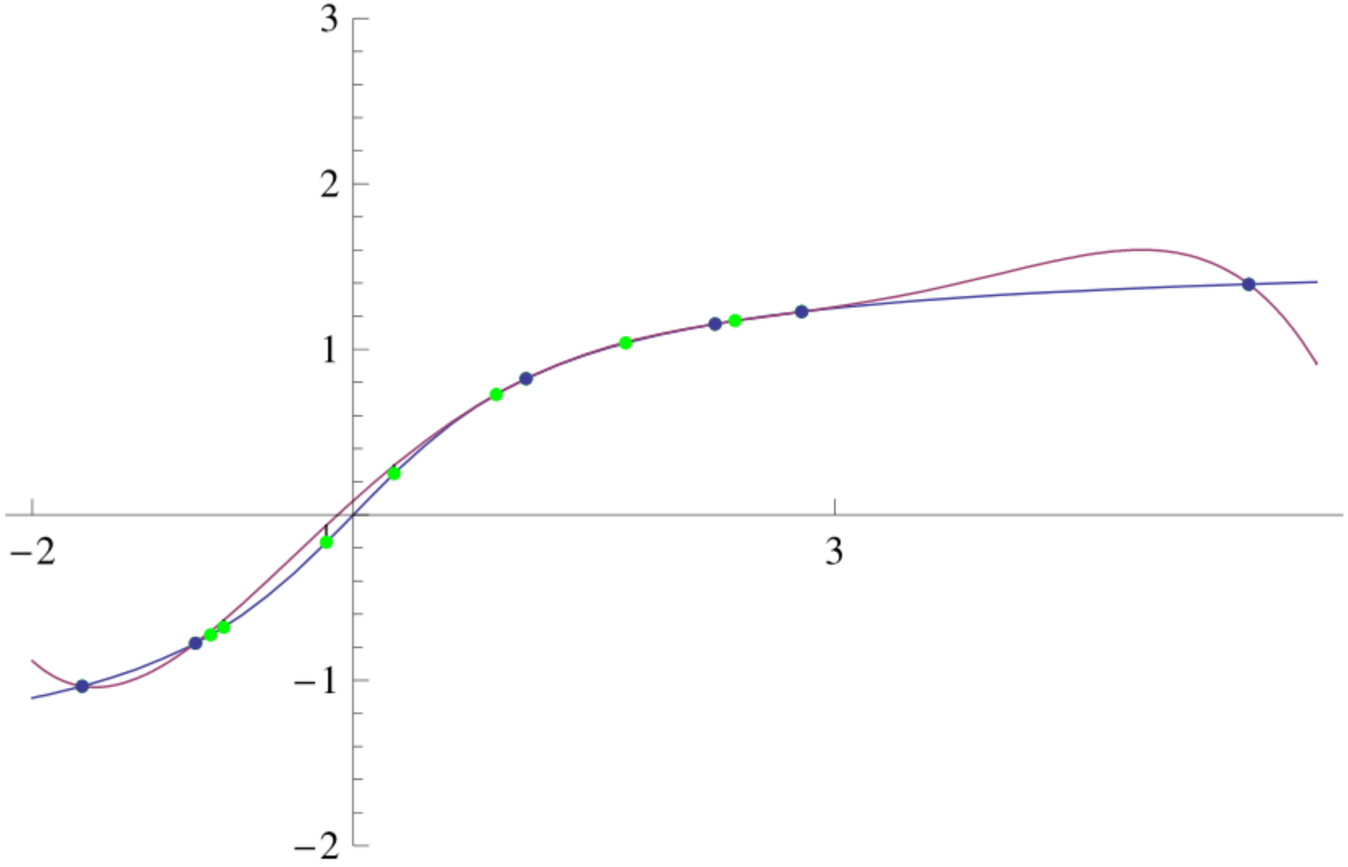
Active learning

Solution: detect when we are extrapolating and switch on learning



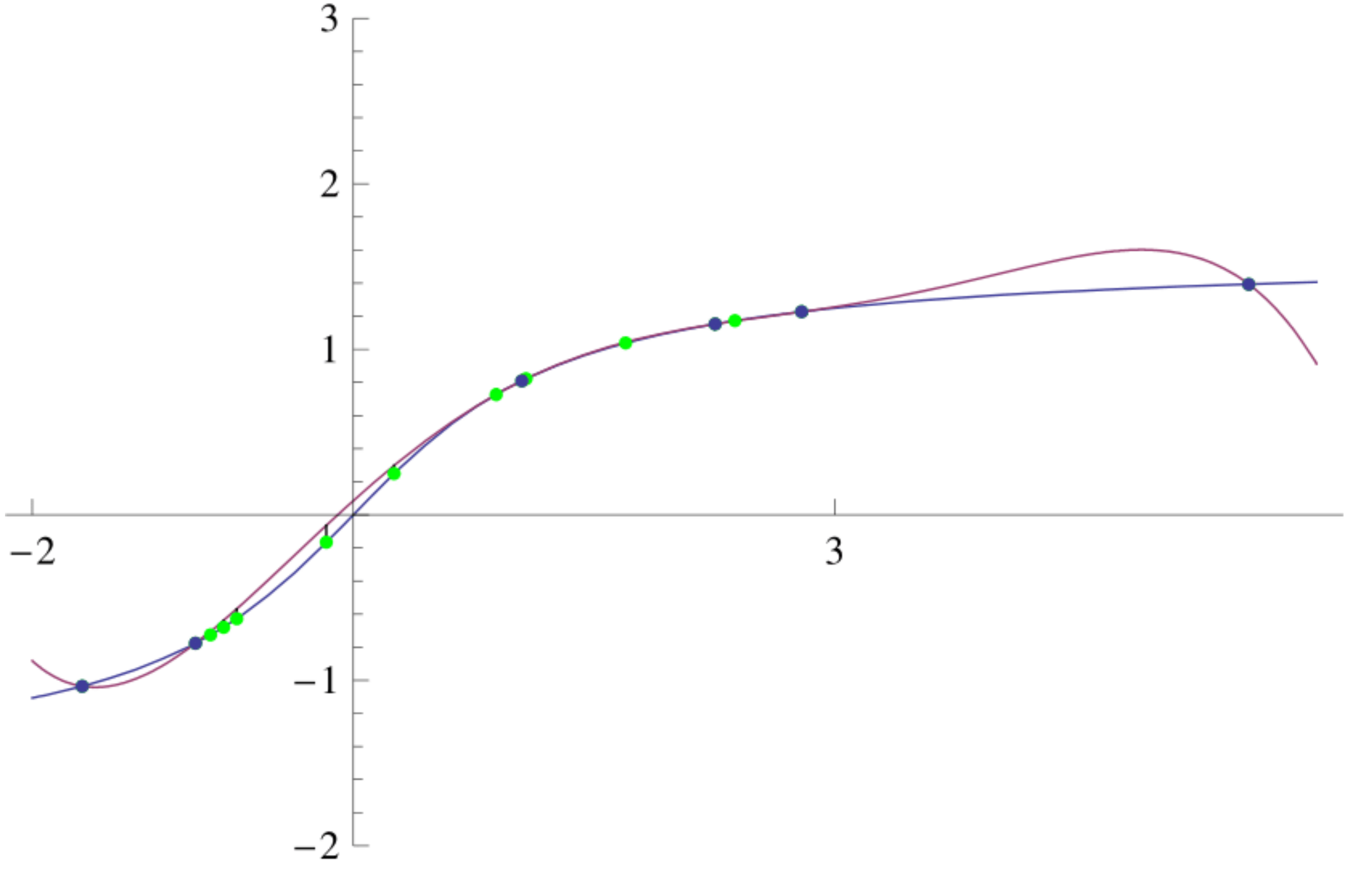
Active learning

Solution: detect when we are extrapolating and switch on learning



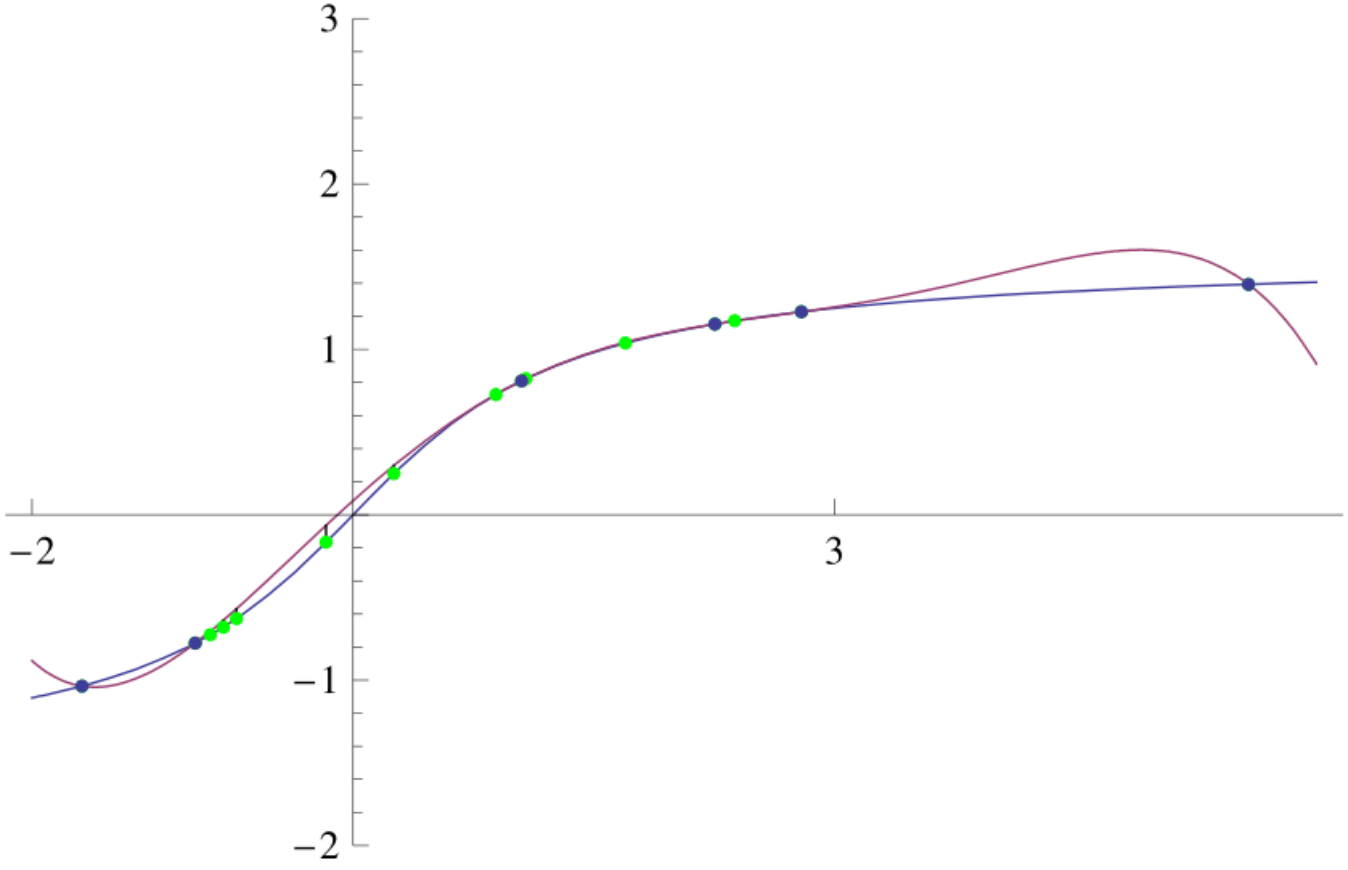
Active learning

Solution: detect when we are extrapolating and switch on learning



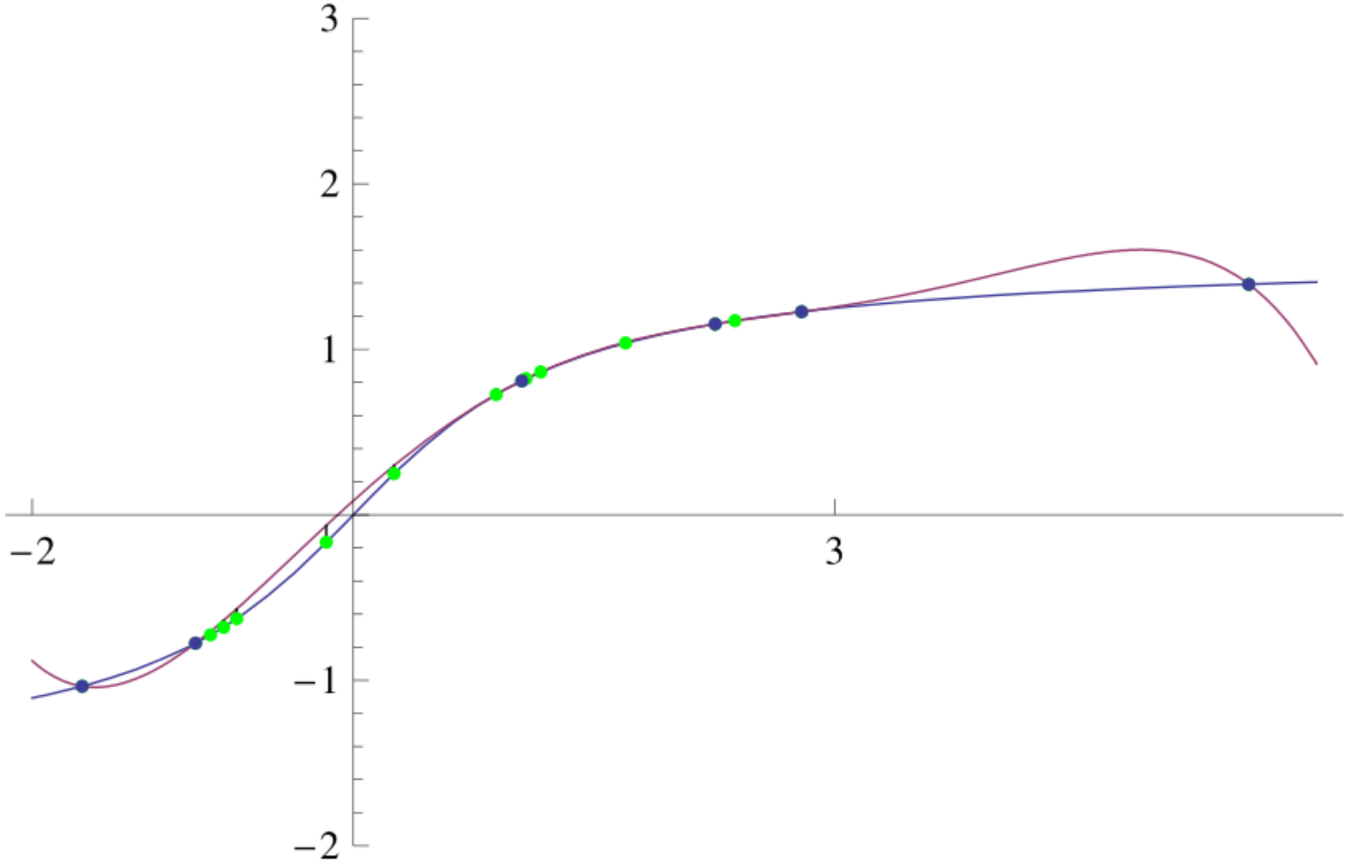
Active learning

Solution: detect when we are extrapolating and switch on learning



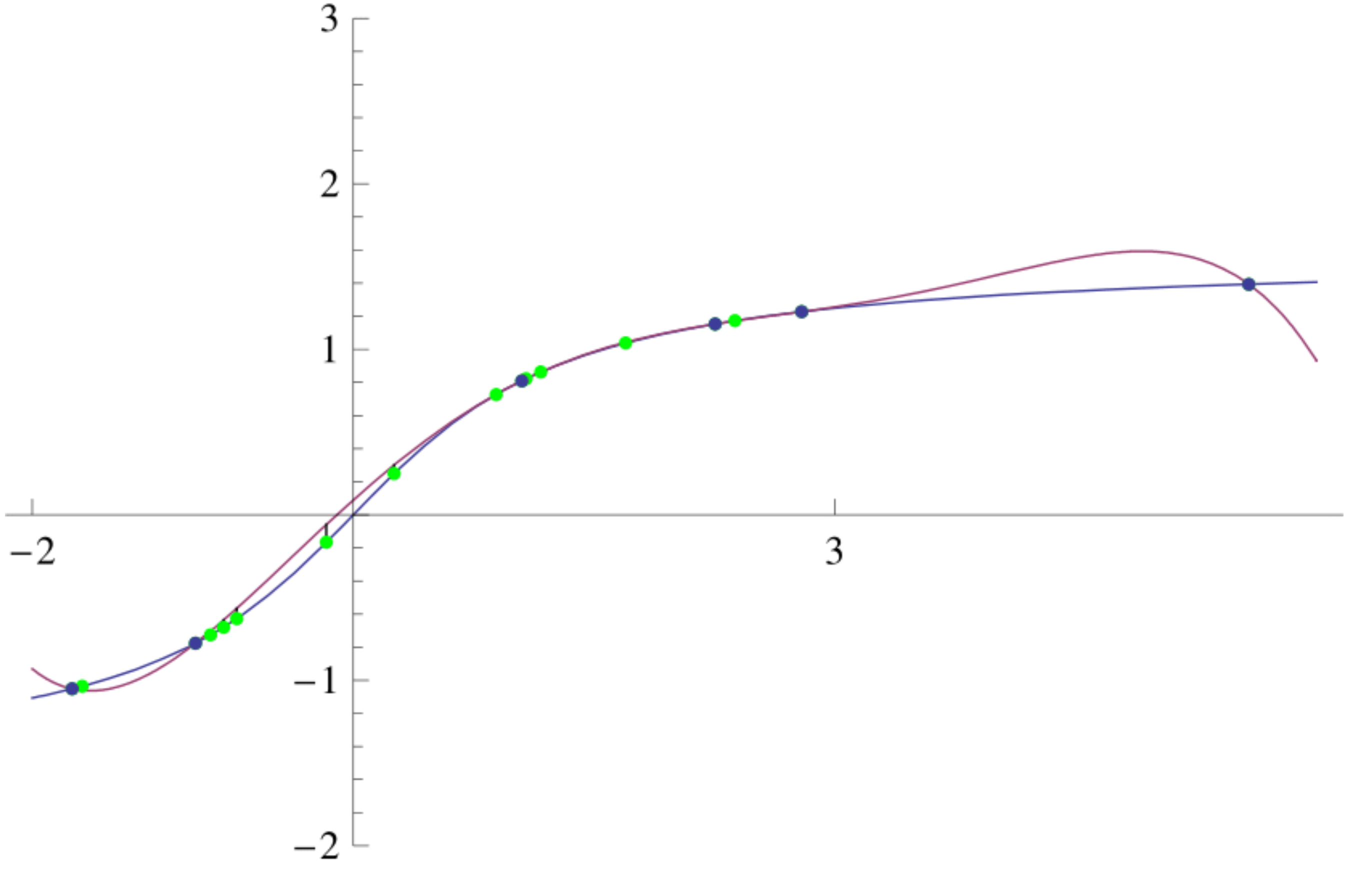
Active learning

Solution: detect when we are extrapolating and switch on learning



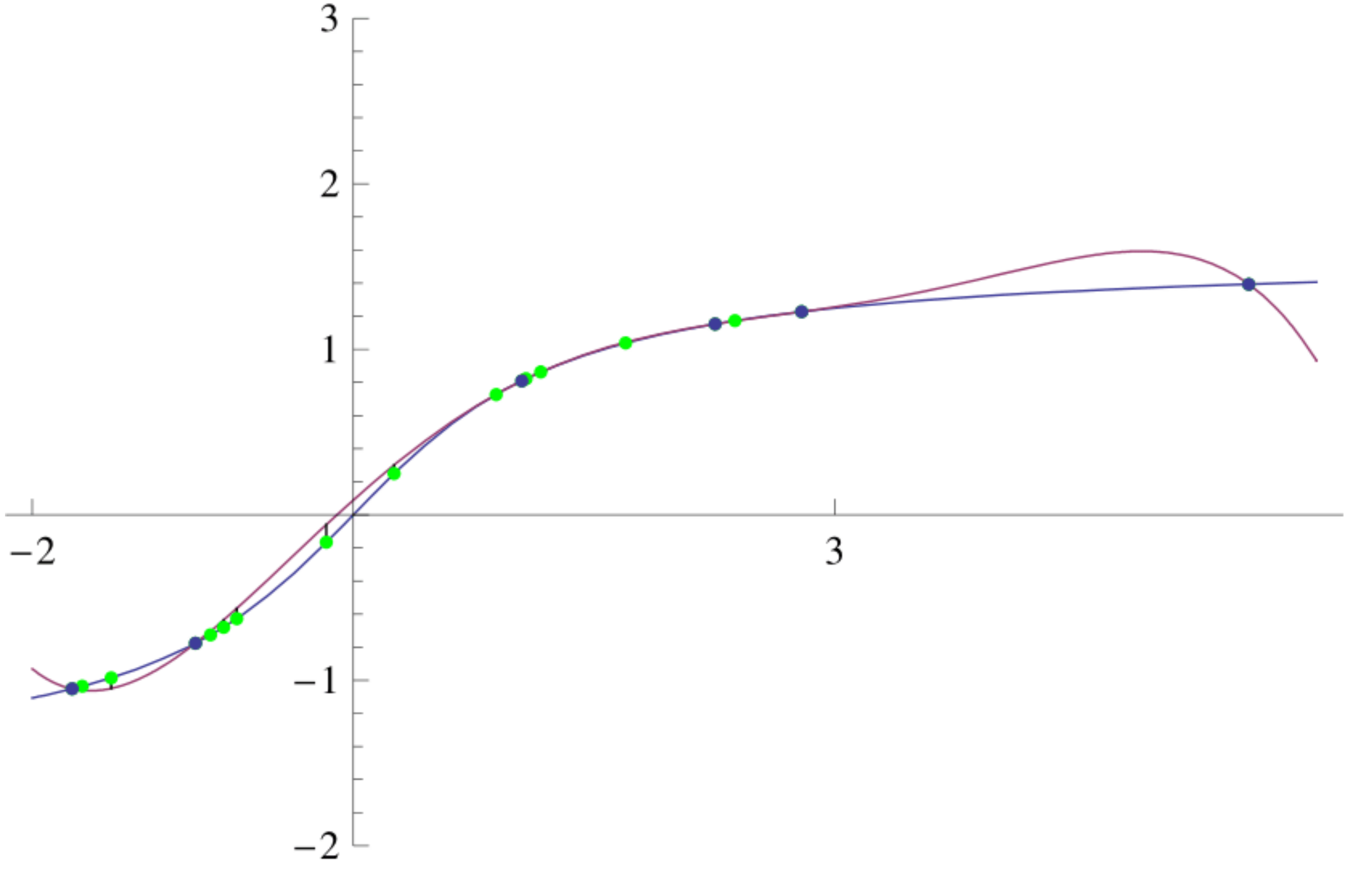
Active learning

Solution: detect when we are extrapolating and switch on learning



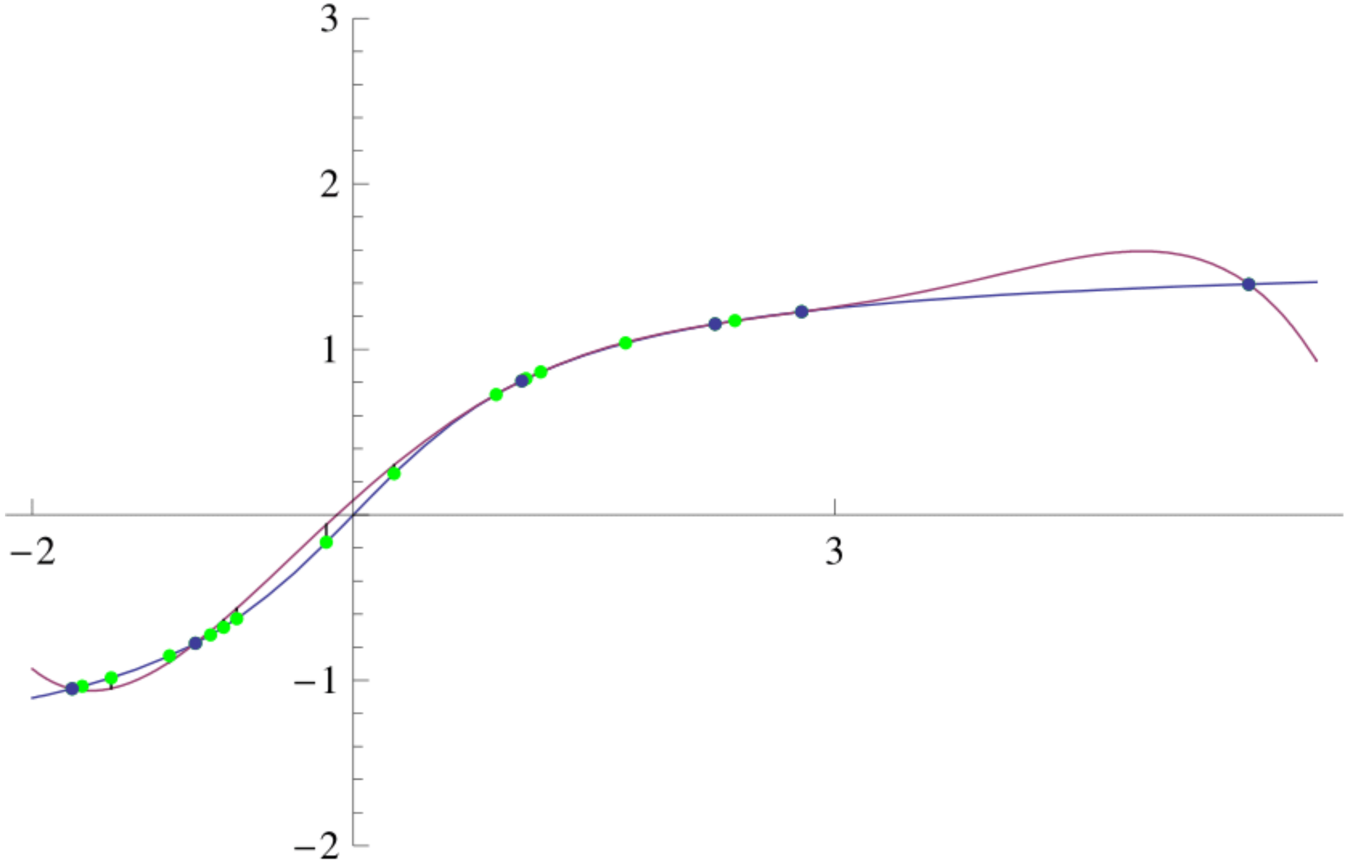
Active learning

Solution: detect when we are extrapolating and switch on learning



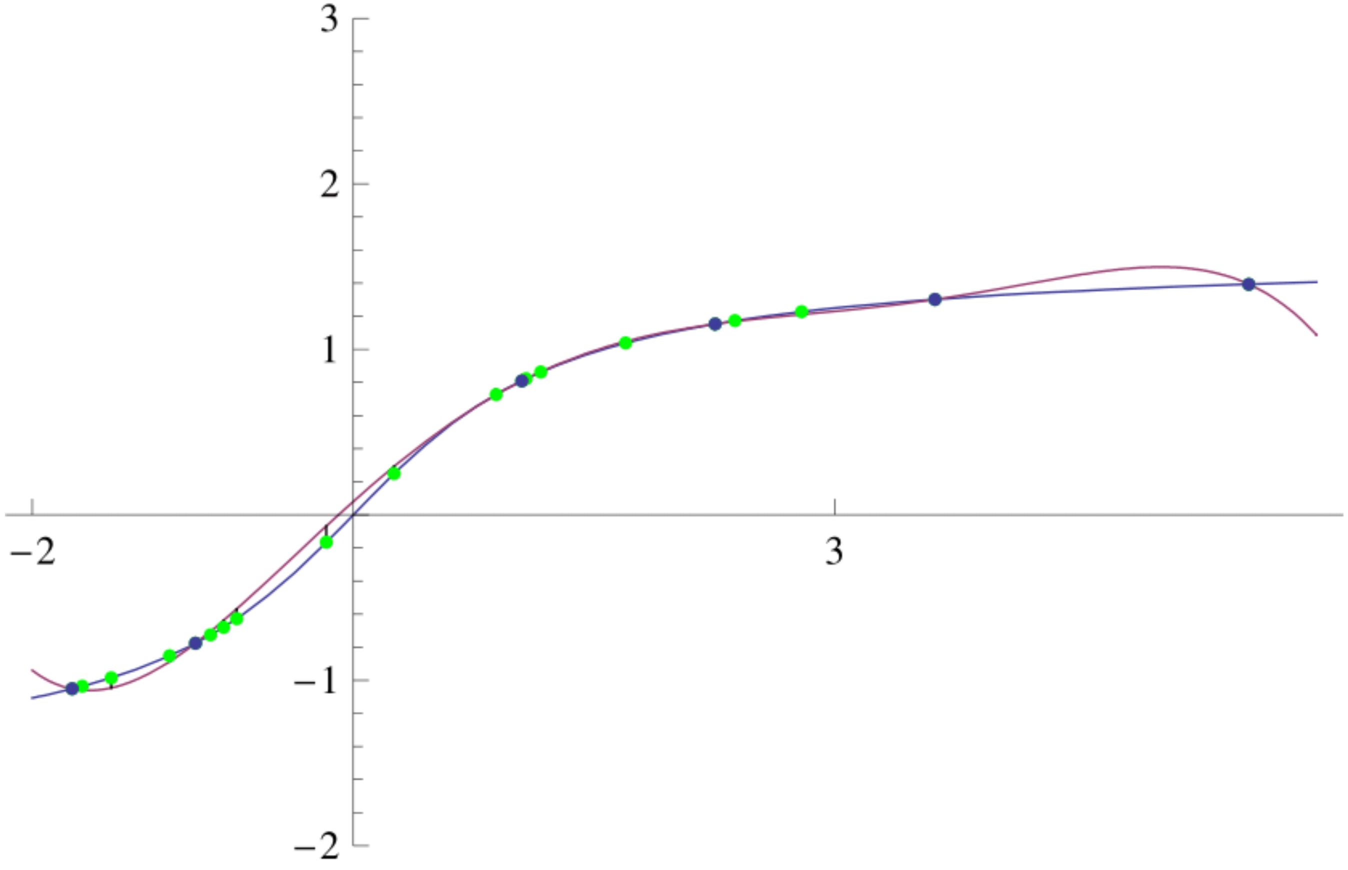
Active learning

Solution: detect when we are extrapolating and switch on learning



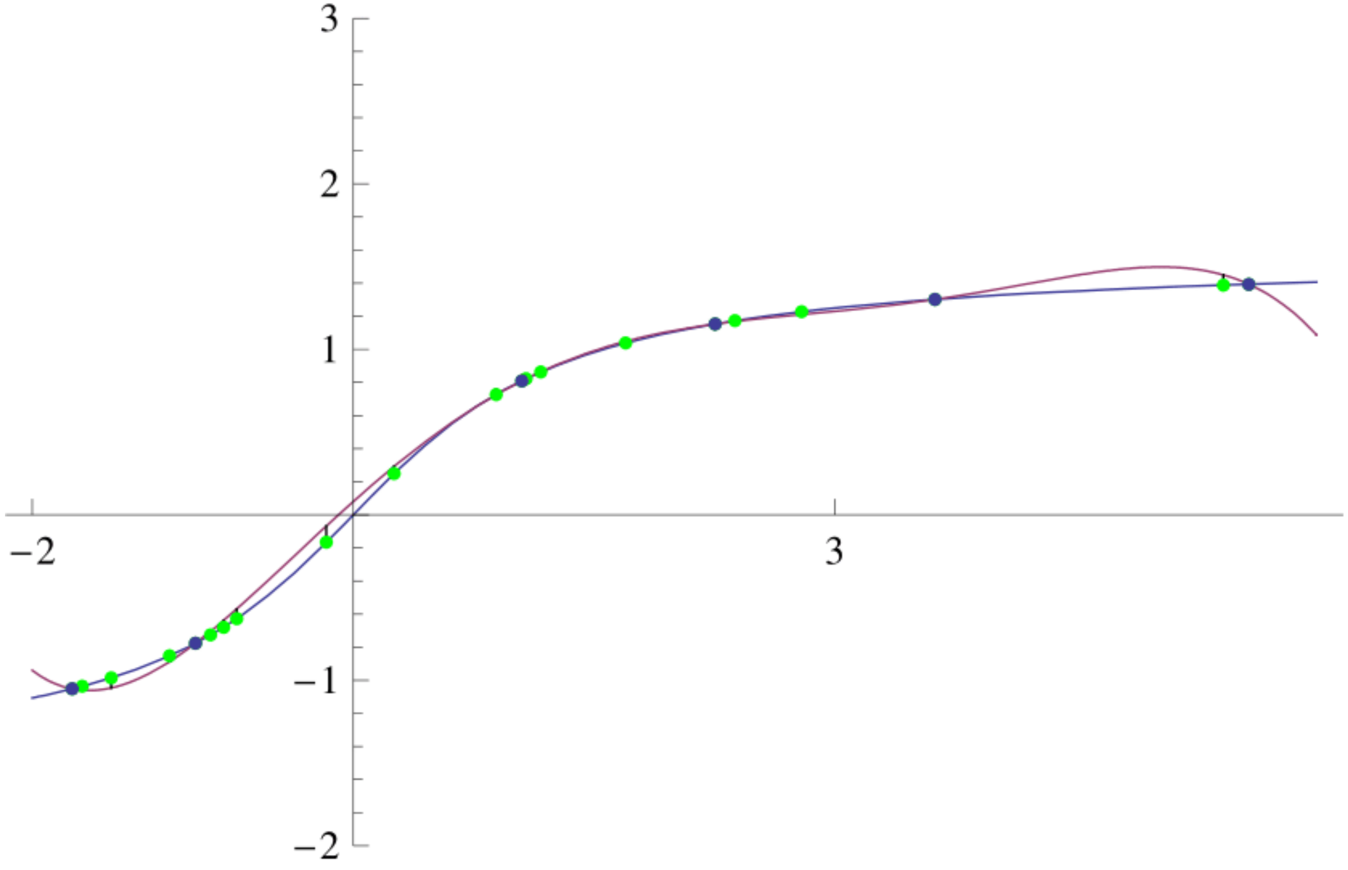
Active learning

Solution: detect when we are extrapolating and switch on learning



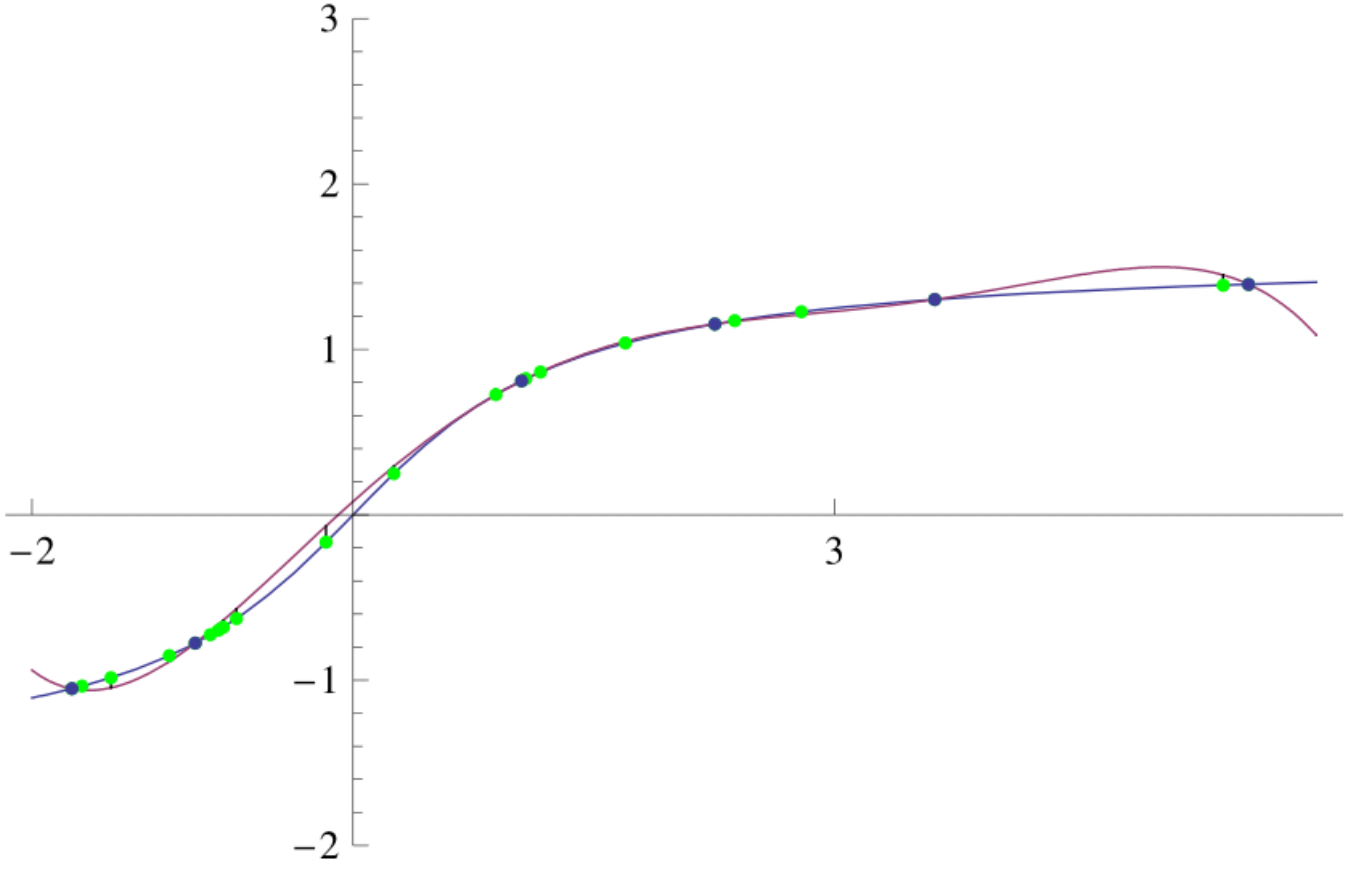
Active learning

Solution: detect when we are extrapolating and switch on learning



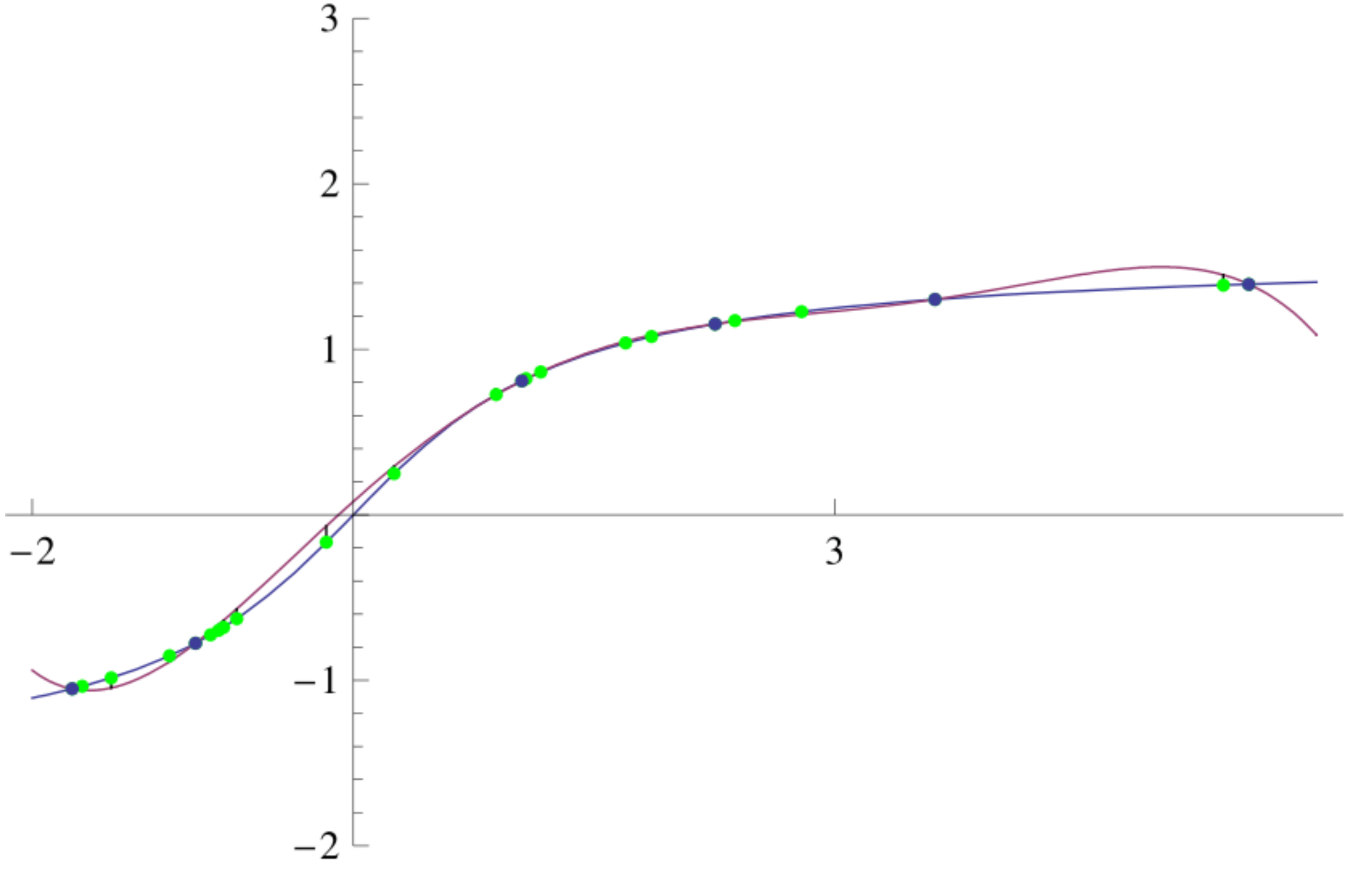
Active learning

Solution: detect when we are extrapolating and switch on learning



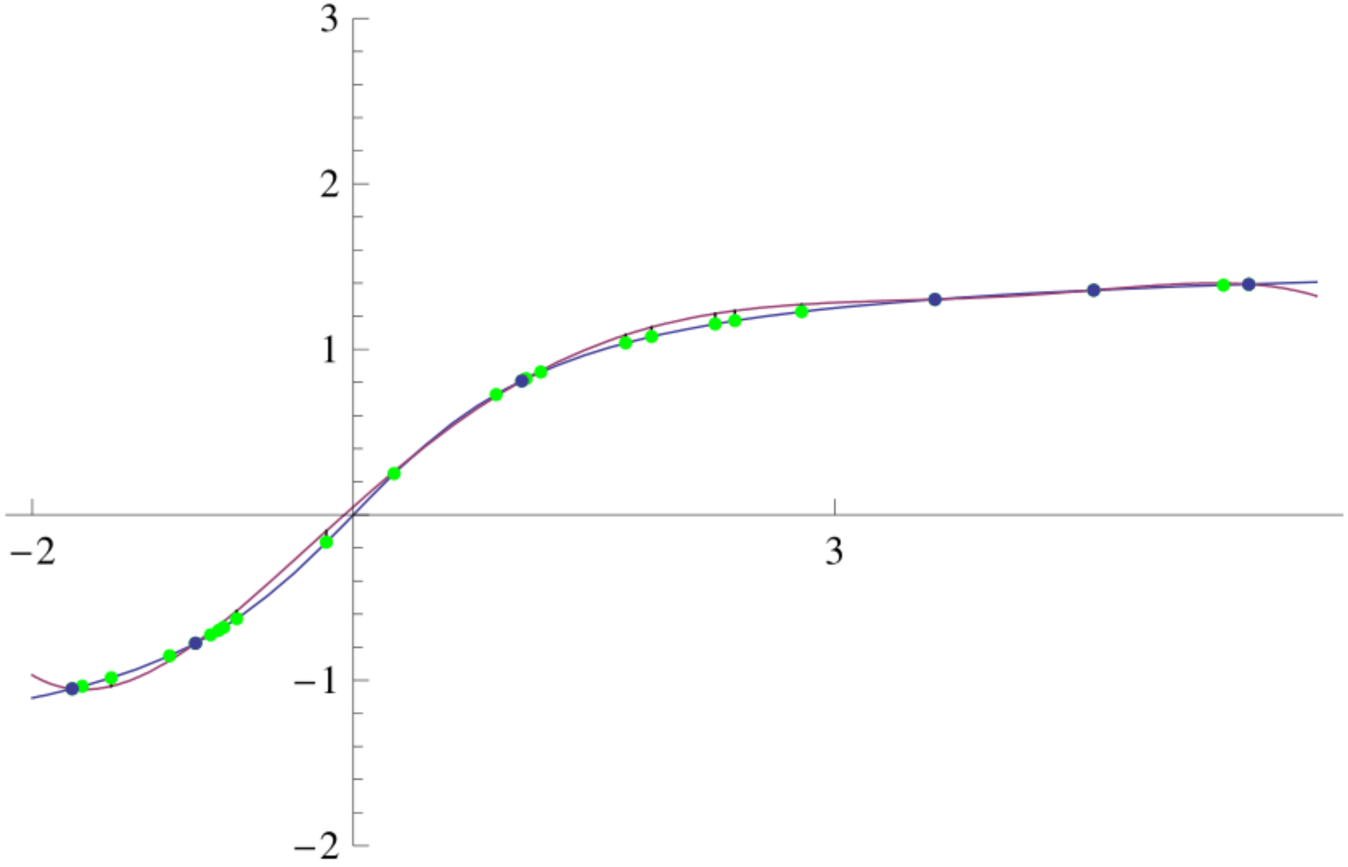
Active learning

Solution: detect when we are extrapolating and switch on learning



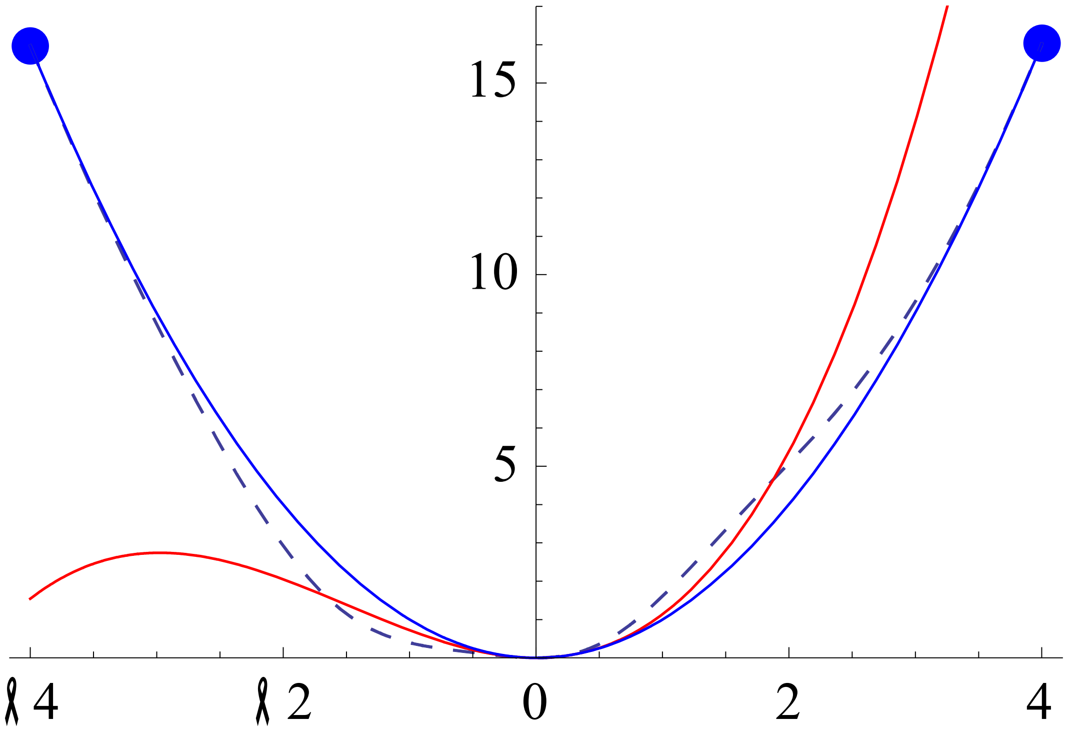
Active learning

Solution: detect when we are extrapolating and switch on learning



Active learning: it’s about reliability, not accuracy

- Fitting $E(x) = x^2 + x^3\exp(-x^2\ /2)$ with $E(x) = c_1x^2 + x_2x^3$
- Red: Minimizing the error on an “exact, infinitely long” MD, error=0.25
- Blue: Active learning, error=0.46



Reliable ->

Not reliable ->

How we do it?

D-optimality

Skip to Applications

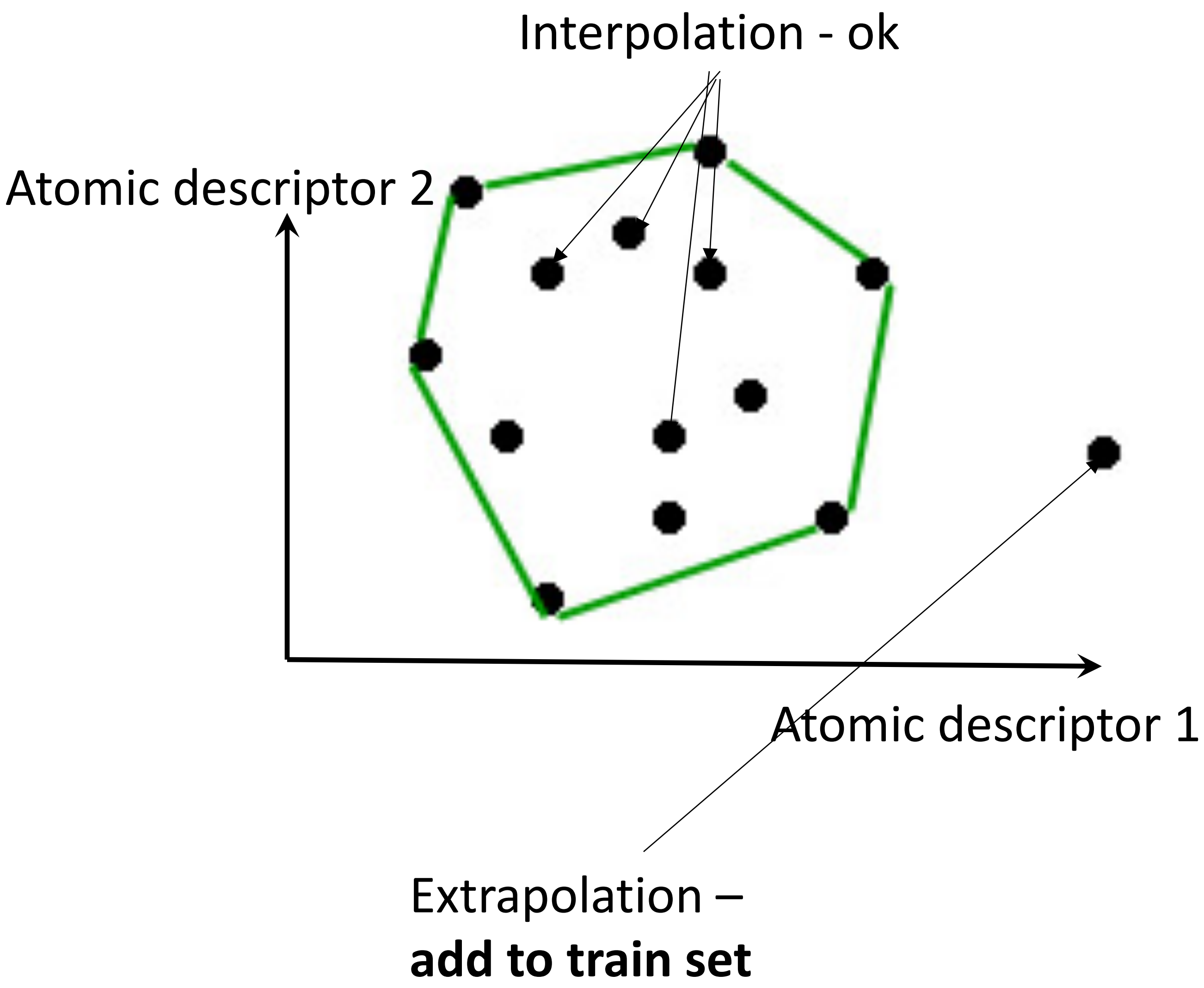
D-optimality

essentially

- detects hitting outside a convex hull,

but for linear models

(convex hull -> simplex)



E. Podryabinkin, A. Shapeev (2017)

Active Learning (AL) of Interatomic Potentials

- Fitting equations (overdetermined):

$$\sum_{\alpha=1}^{N} \theta_\alpha b_\alpha\left(\mathbf{cfg}^{(\boldsymbol{k})}\right) = E^{\mathrm{qm},(\mathbf{k})}$$

- Its matrix: $B = \begin{pmatrix} b_1\left(\mathbf{cfg}^{(\mathbf{1})}\right) & \dots & b_N\left(\mathbf{cfg}^{(\mathbf{1})}\right) \\ \vdots & \ddots & \vdots \\ b_1\left(\mathbf{cfg}^{(\boldsymbol{K})}\right) & \dots & b_N\left(\mathbf{cfg}^{(\boldsymbol{K})}\right) \end{pmatrix}$
- D-optimality criterion: find an $N \times N$ submatrix A with largest $|\det(A)|$
- Selecting rows = selecting configurations

AL in practice

MAXVOL algorithm (*Goreinov et al., 2010*):

- Given:
  - Current set $A = \begin{pmatrix} b_1\left(\mathbf{cfg}^{(\mathbf{1})}\right) & \dots & b_N\left(\mathbf{cfg}^{(\mathbf{1})}\right) \\ \vdots & \ddots & \vdots \\ b_1\left(\mathbf{cfg}^{(\boldsymbol{N})}\right) & \dots & b_N\left(\mathbf{cfg}^{(\boldsymbol{N})}\right) \end{pmatrix}$
  - Candidate (new) $\mathbf{cfg}^*$
- Define extrapolation grade (EG) = factor by which $|\mathrm{det}A|$ can increase
- EG<1 = interpolation. $\mathrm{EG} > 1 + \epsilon$ = include $\mathbf{cfg}^*$ in the training set
- Can be done at $O(N^2)$ complexity

AL: interpretations

Geometric interpretation

- The vector $b_1(\mathbf{cfg}^*) \quad \ldots \quad b_N(\mathbf{cfg}^*)$ is a descriptor in an N-dimensional space
- Increasing $|\det A|$ = increasing the volume of the simplex based on $\mathbf{cfg}^{(\mathbf{1})}, \ldots, \mathbf{cfg}^{(\boldsymbol{N})}$

Information-theoretic interpretation

- information = $\log |\det A|$. Configuration is trained on if this increases the information.

AL: interpretations

Statistical interpretation

- If $\mathrm{E}^{\mathrm{qm},(\mathbf{1})}, \dots, \mathrm{E}^{\mathrm{qm},(\boldsymbol{N})}$ have random independent noise, then the noise in the model is minimized

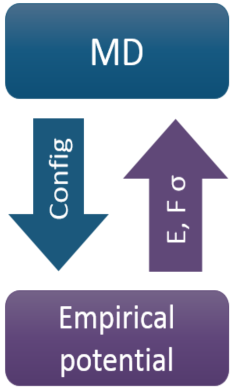
Algebraic interpretation

- It can be shown that $E(\mathbf{cfg}^*) = \sum_i c_i E^{\mathrm{qm},(\boldsymbol{i})}$, hence all $|c_i| \leq 1 \Leftrightarrow$ $E(\mathbf{cfg}^*)$ is interpolated through $E^{\mathrm{qm},(\boldsymbol{i})}$.

Applications

Application #1: Learning on the fly

MD with empirical potential



MD

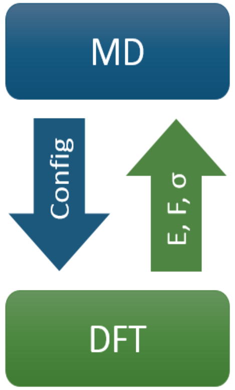
Config

E, F σ

Empirical potential

+ Fast
– Qualitative accuracy only

Ab-initio MD



MD

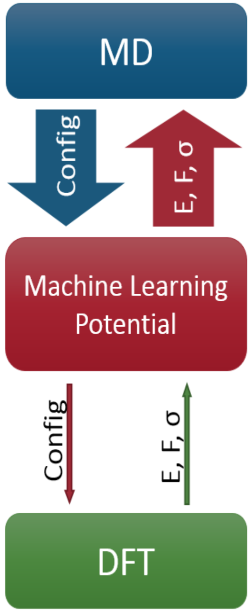
Config

E, F, σ

DFT

– Time consuming
+ Accurate

Hybrid MD



MD

Config

E, F, σ

Machine Learning Potential

Config

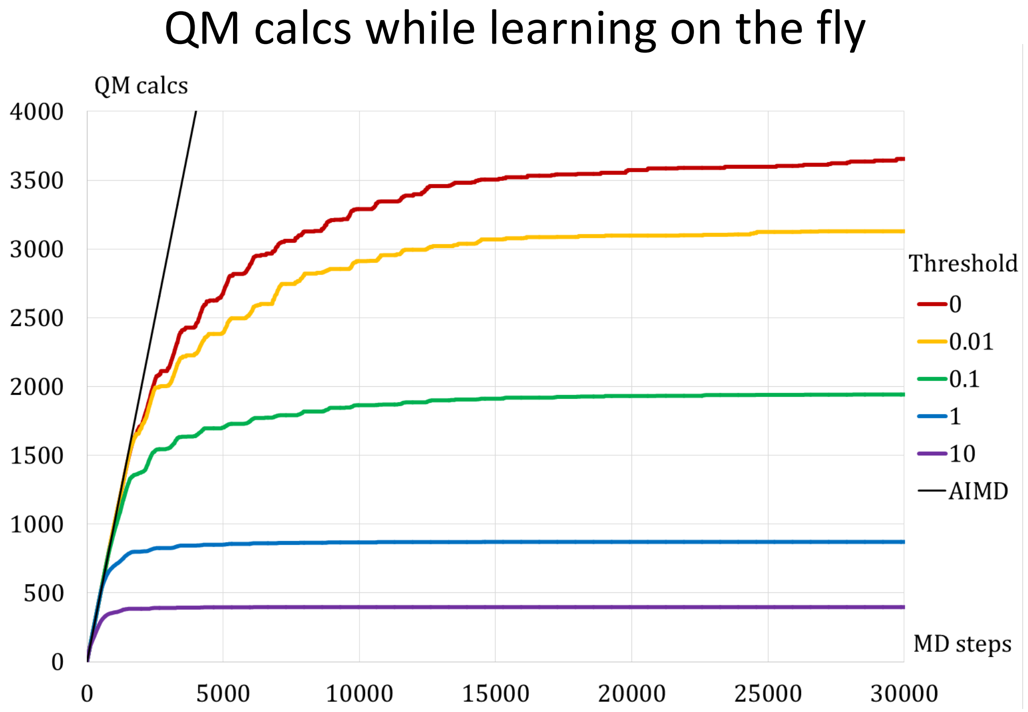
E, F, σ

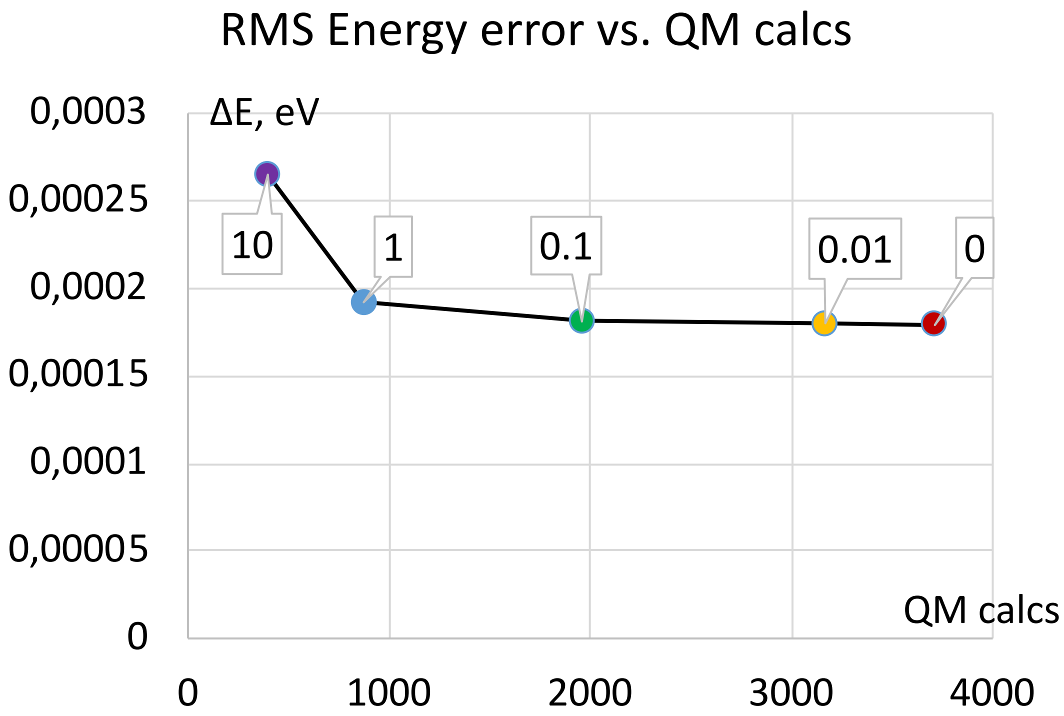
DFT

+ Fast
+ Accurate (hopefully)

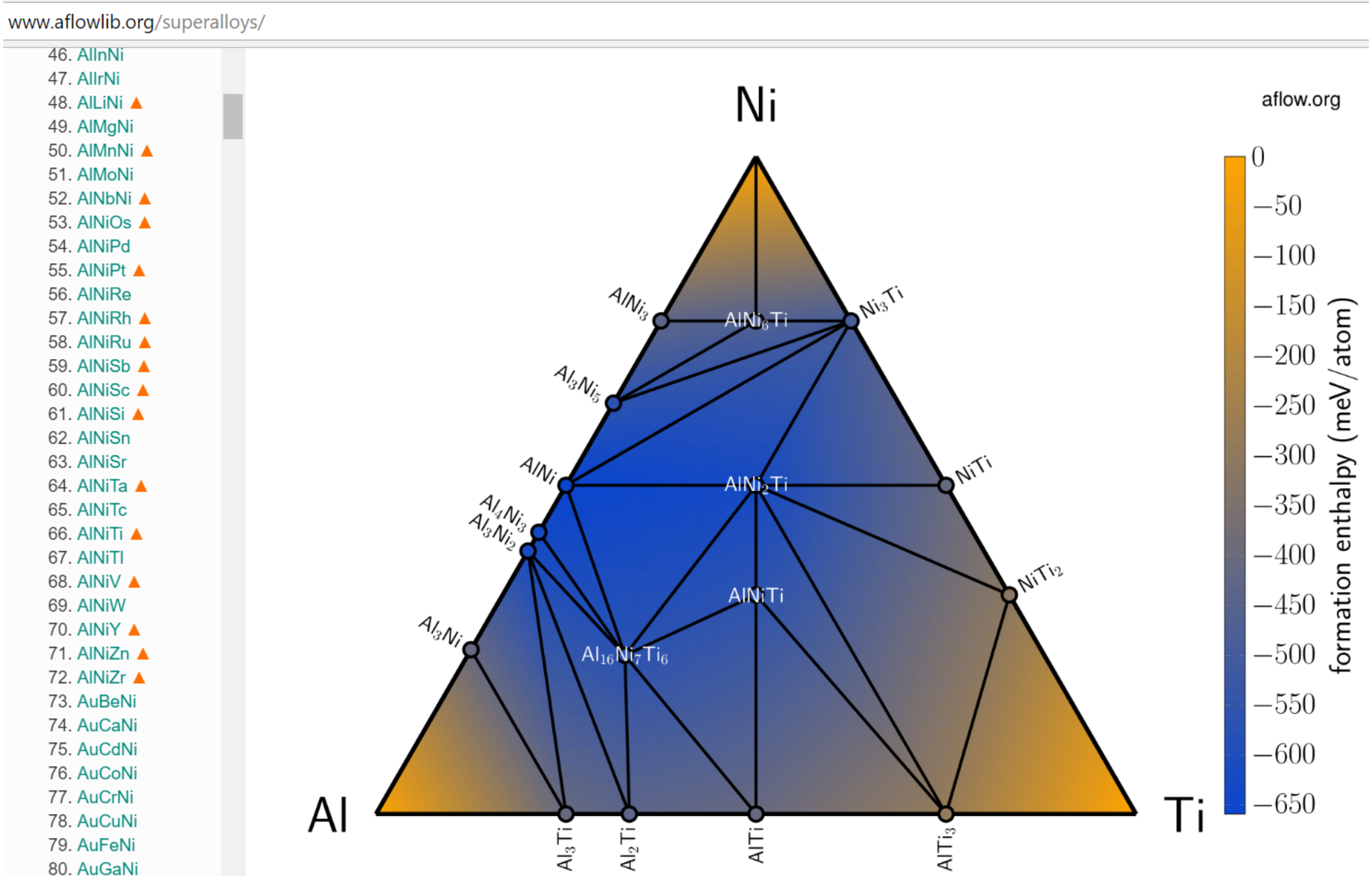
- Combines training and evaluation of MLIP
- Detects and learns “extrapolative” configurations
- Robust
- Balancing accuracy and amount of QM calcs

Application example #0: Learning on the fly in MD process at NVT-ensemble of 128 BCC-Li atoms





Conclusion: Amount of QM calcs can be reduced several times at the cost of minor losses in accuracy



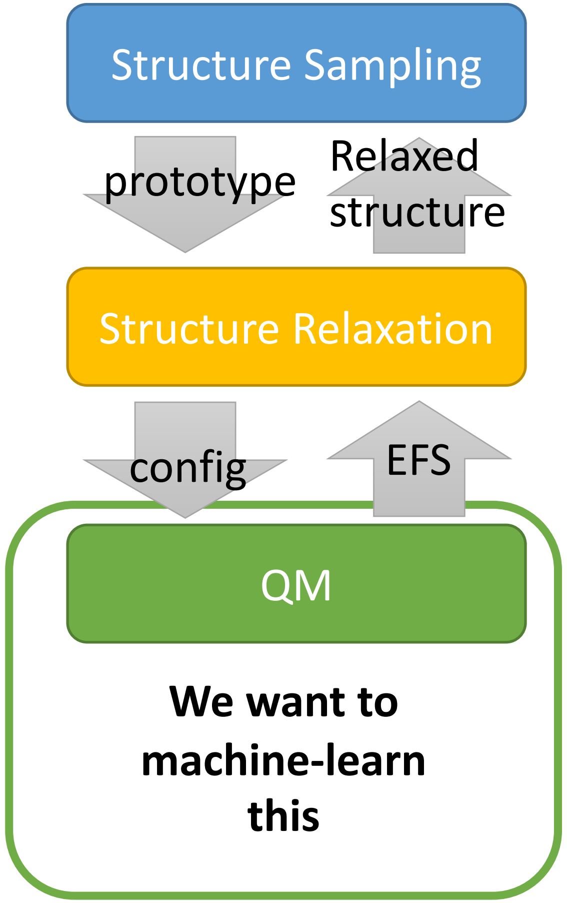
www.aflowlib.org/superalloys/

46. AlInNi
47. AlIrNi
48. AlLiNi ▲
49. AlMgNi
50. AlMnNi ▲
51. AlMoNi
52. AlNbNi ▲
53. AlNiOs ▲
54. AlNiPd
55. AlNiPt ▲
56. AlNiRe
57. AlNiRh ▲
58. AlNiRu ▲
59. AlNiSb ▲
60. AlNiSc ▲
61. AlNiSi ▲
62. AlNiSn
63. AlNiSr
64. AlNiTa ▲
65. AlNiTc
66. AlNiTi ▲
67. AlNiTl
68. AlNiV ▲
69. AlNiW
70. AlNiY ▲
71. AlNiZn ▲
72. AlNiZr ▲
73. AuBeNi
74. AuCaNi
75. AuCdNi
76. AuCoNi
77. AuCrNi
78. AuCuNi
79. AuFeNi
80. AuGaNi

Prediction of convex hull of stable alloys

How it is done:

1. Start with 1500 crystal prototypes (unequilibrated structures)
2. Equilibrate (relax) them with DFT and choose the ones on the convex hull



Structure Sampling

prototype

Relaxed structure

Structure Relaxation

config

EFS

QM

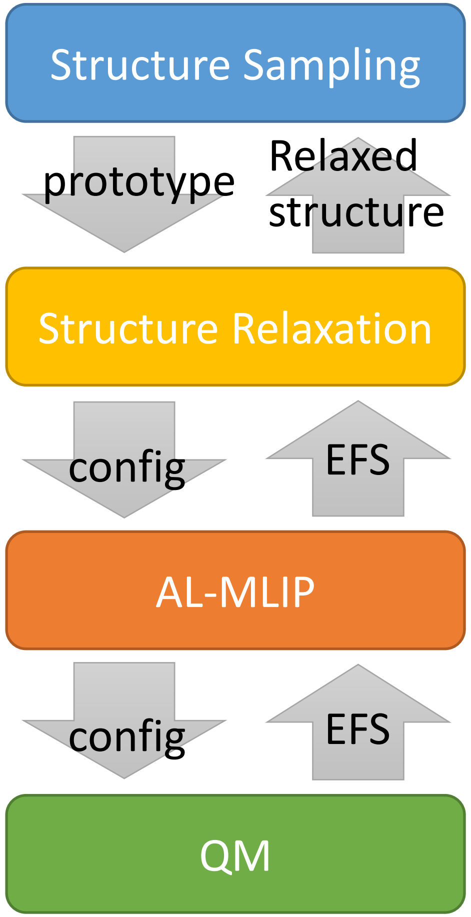
We want to machine-learn this

Convex hulls now

K. Gubaev,E. Podryabinkin,
Gus L.W. Hart, A.S. (2019)

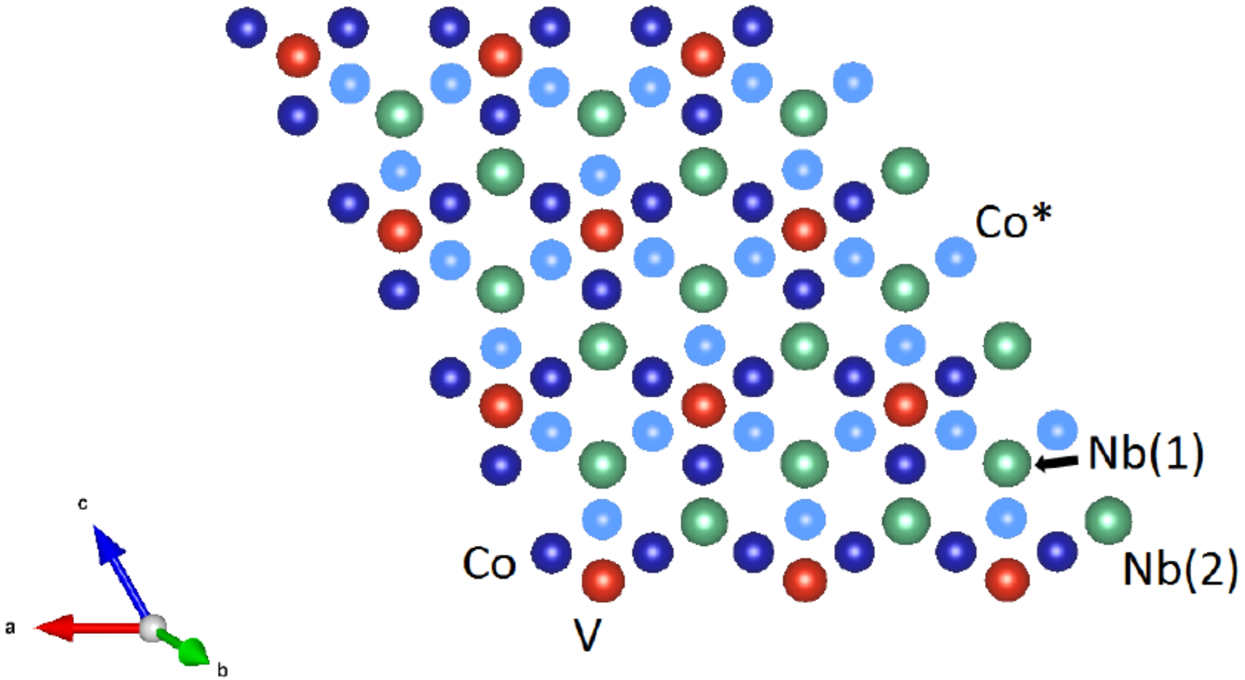
How it is done:

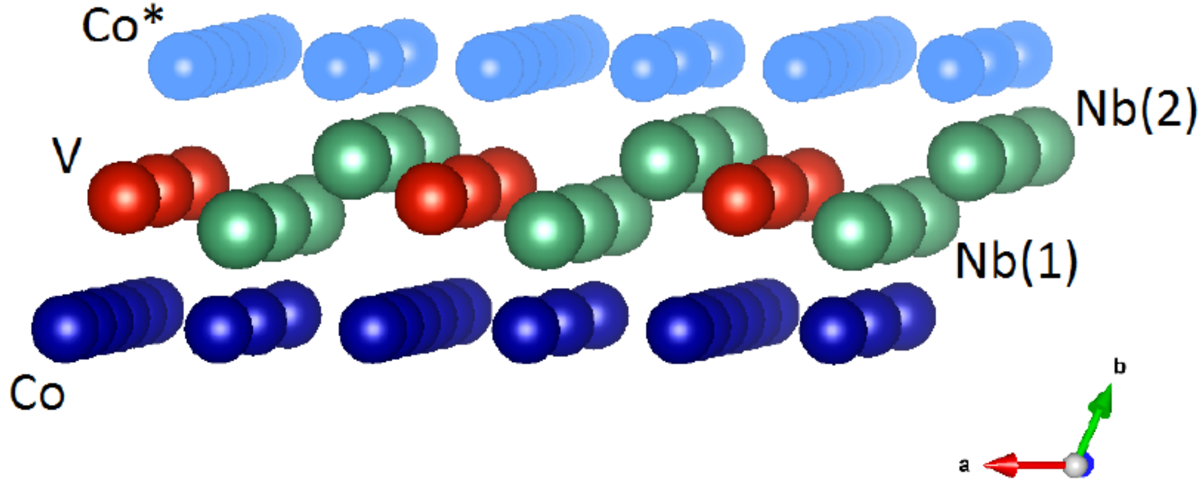
1. Start with 400K crystal prototypes (unequilibrated structures)
2. Equilibrate (relax) them with MLIP while learning on the fly



Results

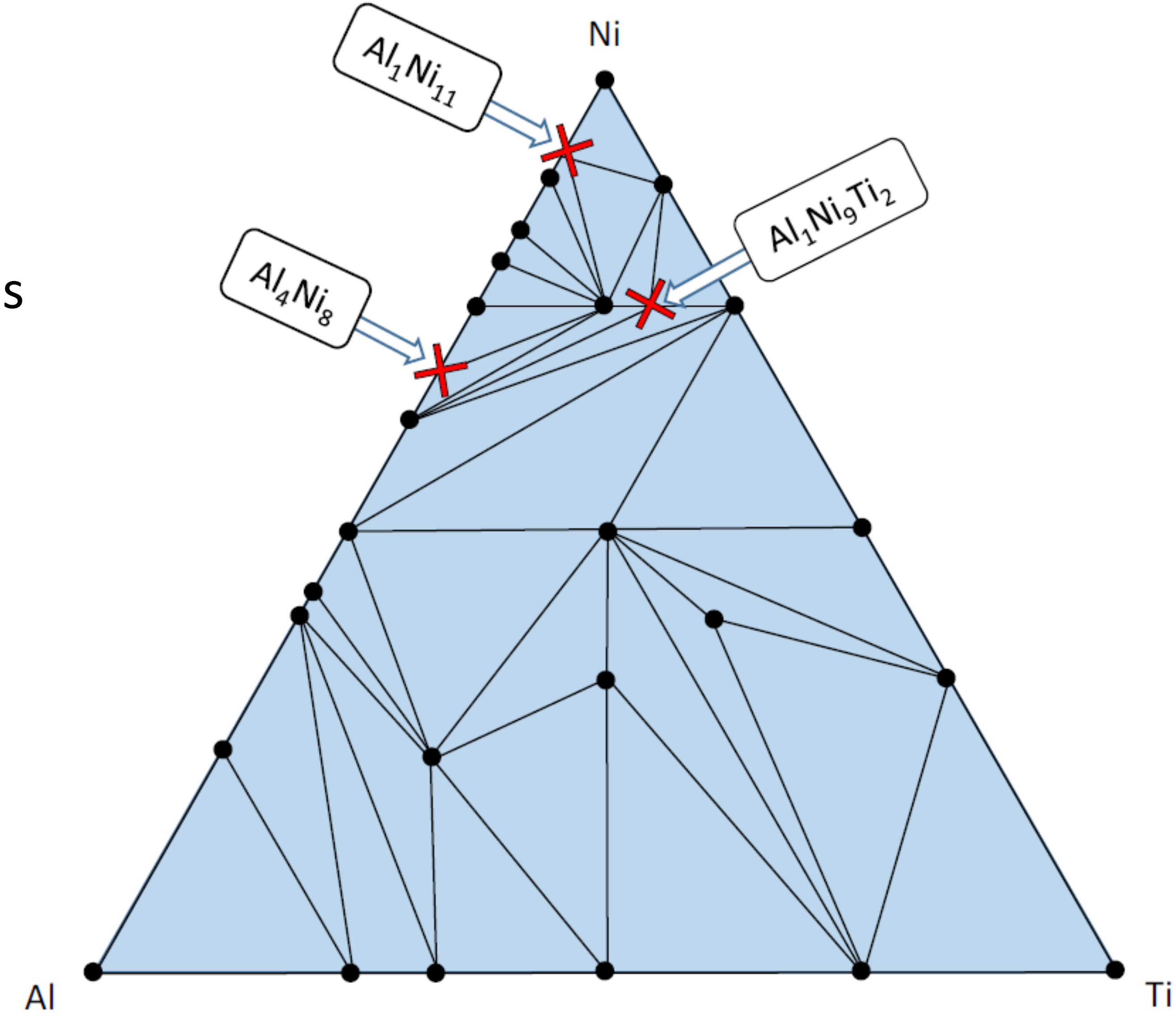
- Some newly discovered structures are hard to “sample passively”:





Convex hulls now: details

1. Screen-1:
   1. Start with **400K** structures
   2. Obtain **400K** relaxed structures, with RMSE = **25** meV/atom
   3. Retain **60K** low-energy structures (within 4-σ)
2. Screen-2:
   1. Start with **60K** structures
   2. Obtain **60K** relaxed structures, with RMSE = **8** meV/atom
   3. Retain **7K** low-energy structures (within 4-σ)
3. Final relaxation:
   1. Relax **7K** structures on DFT

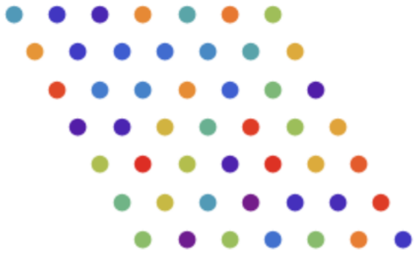


Results and Discussion

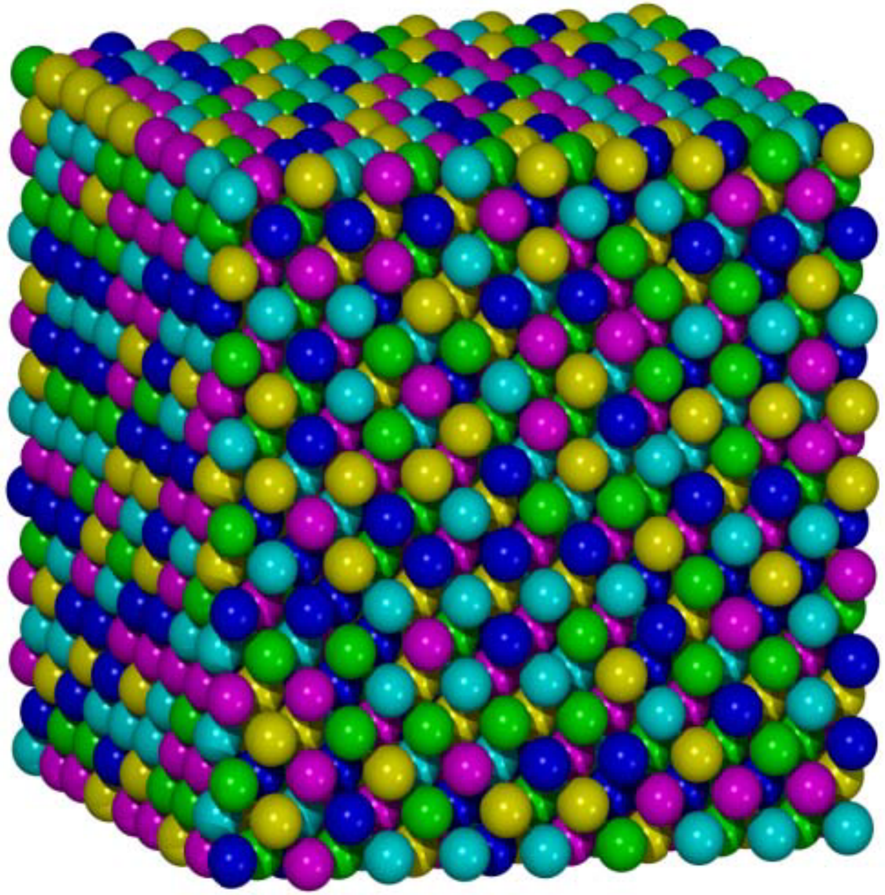
- No approximation error in the answer!
  (We only take a risk of missing a structure in the 4-σ interval.)
- 100x speed-up; CPU time:
  1. Final relaxation: 90%
  2. Training set: 9%
  3. Training, Relaxation: 1%
- Main challenge: reduce the 90% $\Leftarrow$ improve accuracy (8 meV/atom):
  - Go beyond local environments (we quickly reach the limit with local interaction)
  - Include spins (suffer from “jumping” from nonmagnetic PES to ferromagnetic PES)
  - Periodic table-wide potential (reuse data from old systems from new systems)
  - Better uncertainty estimation (better than just 4-σ).
- Sampling is now the bottleneck, not DFT (we should make friends with Complex High-Dimensional Energy Landscapes)

On-lattice models: Cluster expansion

- Atoms of different kind sit in the lattice sites.
- Problem: predict the interatomic interaction energy (formation energy, mixing enthalpy)

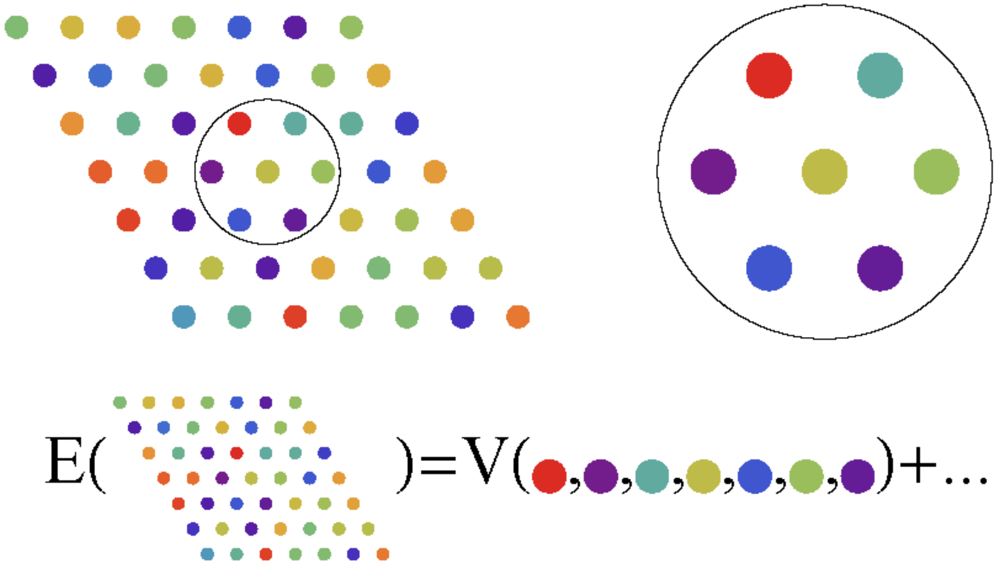
E() = V_1(●)+V_1 (●)+…

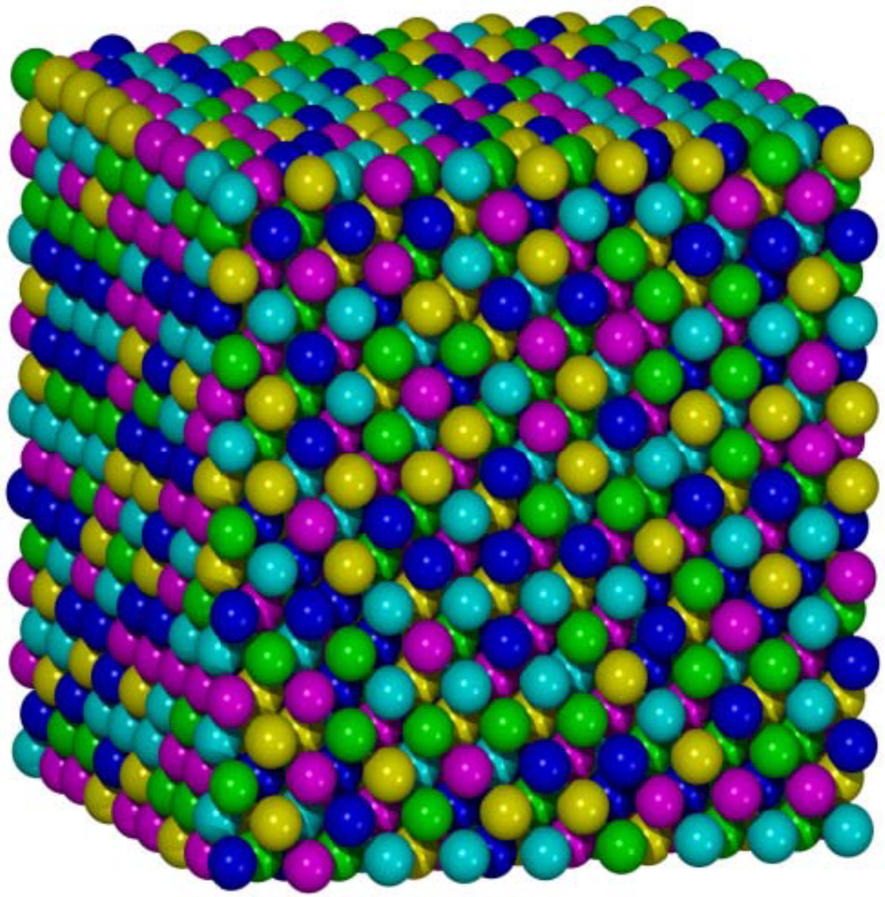
+V_2(● ●) +V_2…
+V_3….



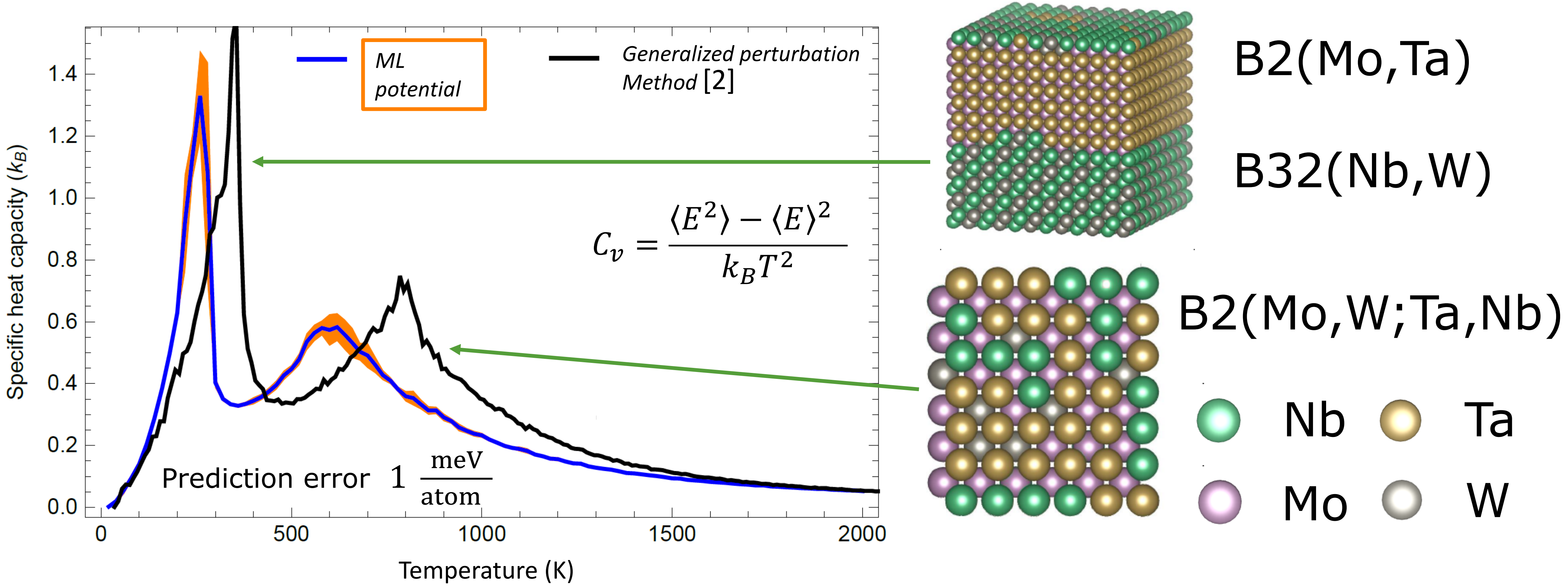
On-lattice models: Potentials

- Atoms of different kind sit in the lattice sites.
- Problem: predict the interatomic interaction energy (formation energy, mixing enthalpy)





Comparison with existing methods: without local lattice distortions

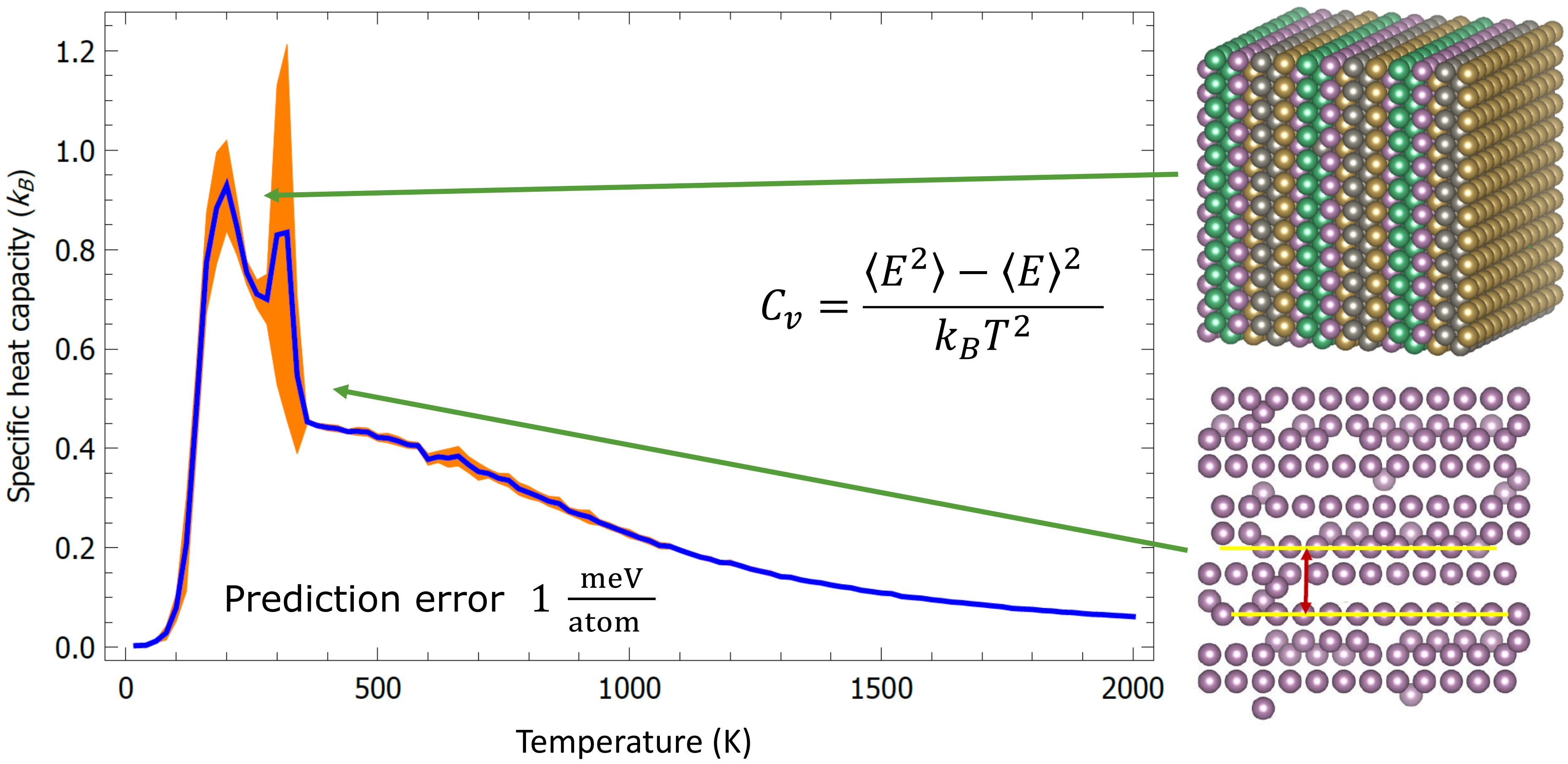


$$C_v = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}$$

Prediction error $1 \, \frac{\text{meV}}{\text{atom}}$

[2] Fritz Körmann, Andrei V Ruban, and Marcel HF Sluiter. Long-ranged interactions in bcc NbMoTaW high-entropy alloys. Materials Research Letters, 5(1):35-40, 2017.

Results & discussion: accounting for local lattice distortions



$$C_v = \frac{\langle E^2\rangle - \langle E\rangle^2}{k_B T^2}$$

Prediction error $1\ \frac{\text{meV}}{\text{atom}}$

Nb-Mo-Ta-W-W-Ta-Mo-Nb

Semi-ordered lattice structure

$\langle 100\rangle$

● Nb ● Ta

● Mo ● W

“Perfect crime” of machine-learning potentials

100-fold speed-up with no detectable trace of using machine learning in the final result (in short, a free lunch)

Alexander Shapeev[1], Konstantin Gubaev[1], Evgeny Podryabinkin[1], Gus Hart[2]

1: Skoltech (Moscow, Russia)

2: BYU (Provo, Utah)

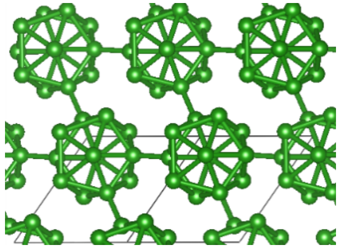
2019 APS Meeting, Boston
04 March, 2019

Application 1b: Boron crystal structure prediction

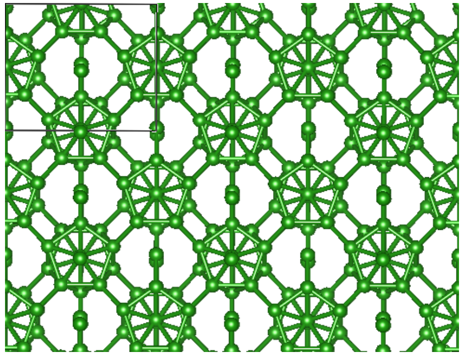
E. Podryabinkin, E. Tikhonov,
A.S., Artem Oganov (2019)

Boron structures prediction challenges:

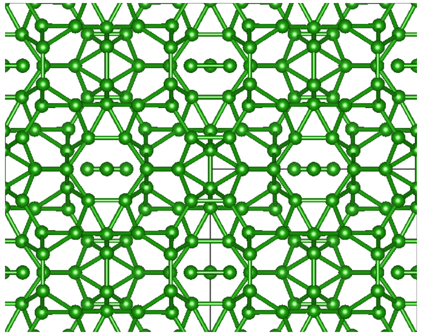
- A lot of allotropes
- Some allotropes has more than 100 atoms (impossible with DFT)
- Small energy/atom difference between structures with PES minima



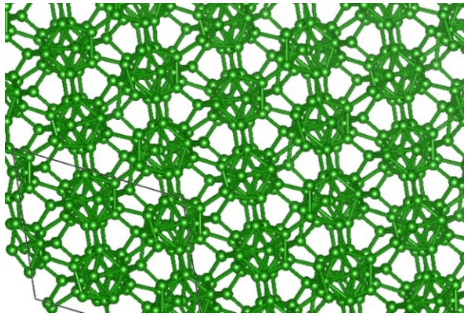
B-12 (6.7058eV/atom)
10 days with DFT
3 days with MLIP



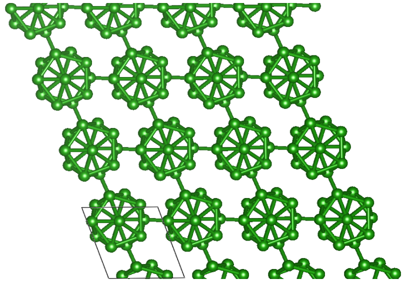
B-28 (6.678eV/atom)
2 months with DFT
5 days with MLIP



B-54 (6.667eV/atom)
2 year on DFT
8 days on MLIP



B-106
Best Found on MLIP
within 2 weeks

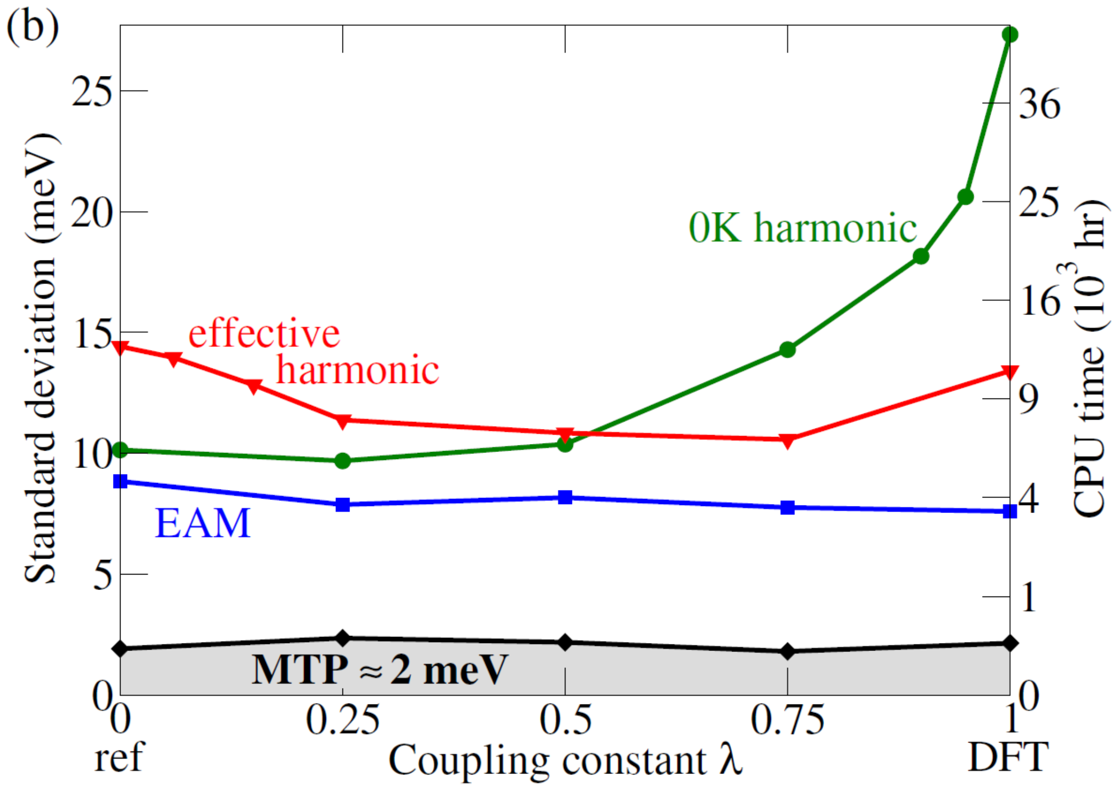


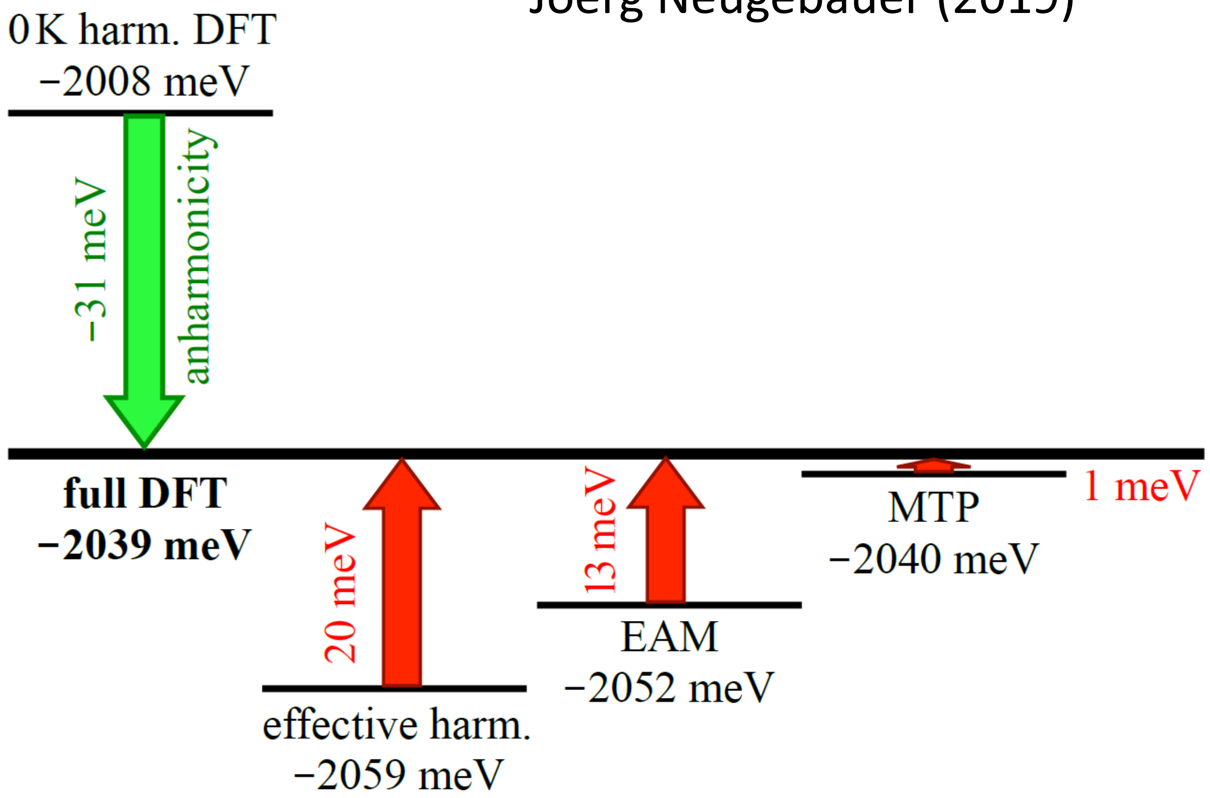
B-108 = B12 x9
10 years with DFT
2 weeks with MLIP

Application #2: thermodynamic integration

- Vibration entropy of a MoNbTaVW quasi-random structure

Blazej Grabowski, Yuji Ikeda,
Fritz Koermann,
Christoph Freysoldt,
Andrew Duff, A.S.,
Joerg Neugebauer (2019)

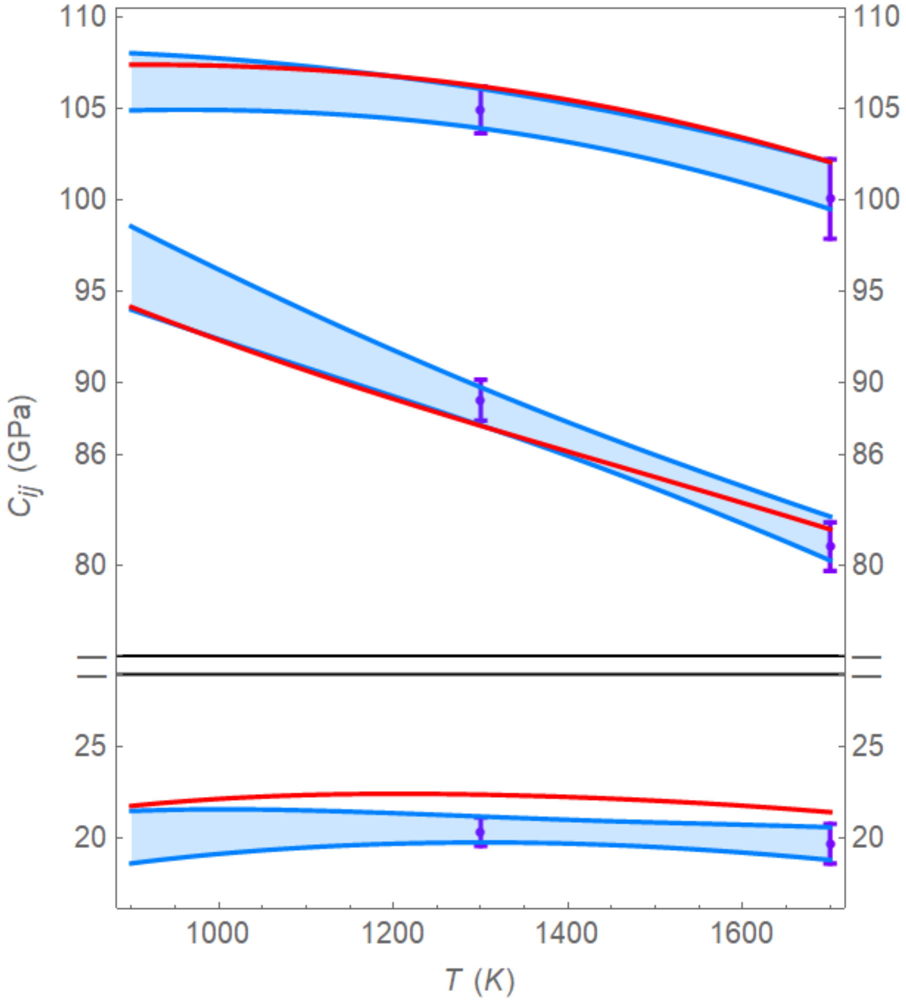




Application #3: elastic properties

A.S., E. Podryabinkin,
K. Gubaev, F. Tasnadi,
Igor Abrikosov (manuscript)

- Elastic constants $C_{11} > C_{12} > C_{44}$ (bcc-Ti)



DFT with uncertainty (50 000 DFT-MD time steps)

MTP (negligible statistical uncertainty)

We trade

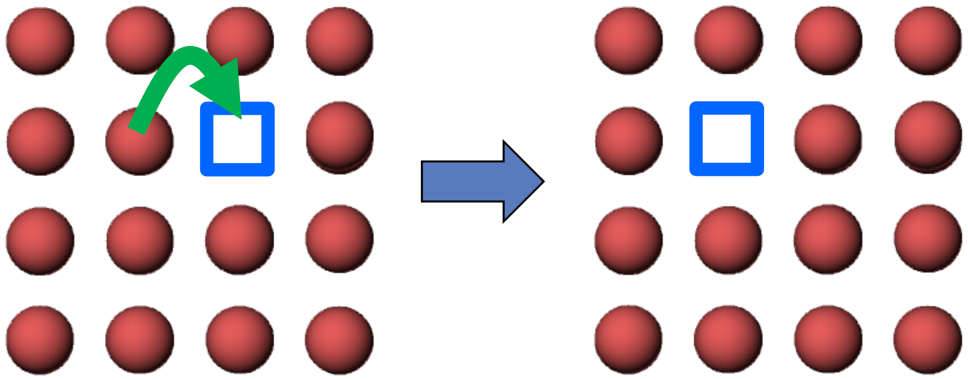
- 1 GPa statistical error

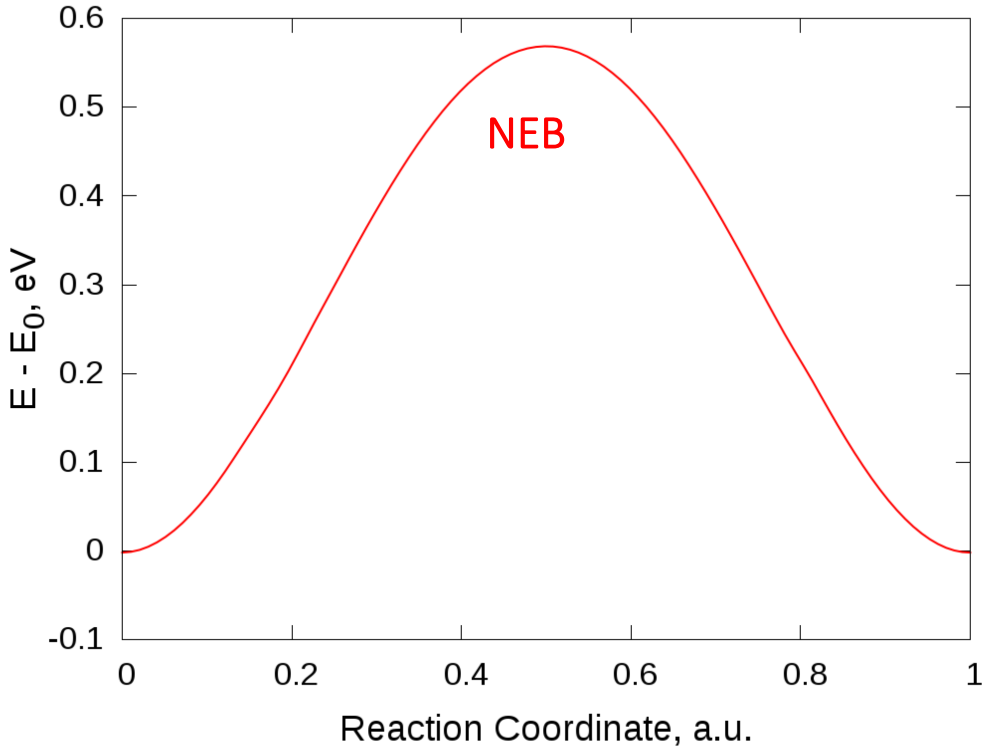
for

- 1 Gpa model error and
- 1000x speed-up

Application #4: Calculation of Diffusivities

Ivan Novoselov,
E. Podryabinkin,
A.S., Alexey Yanilkin (2019)





Potential advantages:

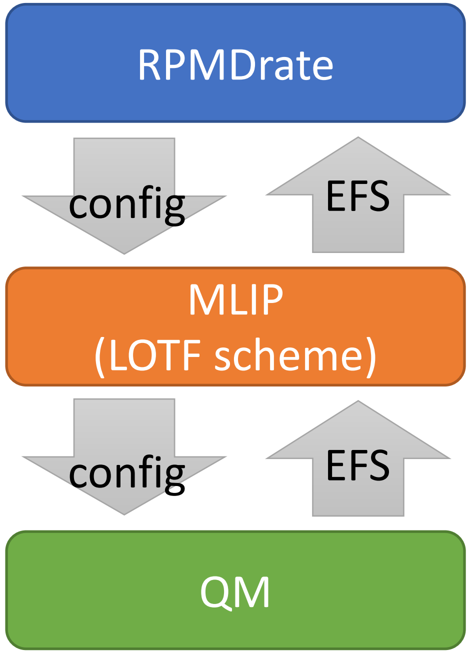
- MTP: accurate description of low-symmetry configurations (e.g. saddle point)
- Active learning: rapid exploration of phase space
- Learning on the fly: effective sampling of rare events

Application #5: Molecular reaction rates

I. Novikov,
Y. Suleimanov, A.S. (2018)

- Use RPMD + MTP

Rate **(cm^3/s)**	**Existing PES** **(reference)**	**AL-MLIP**
Classical	4.5×10^{-14}	4.1×10^{-14} (9% error)
Quantum-corrected (128 RPMD-beads)	2.5×10^{-12}	2.1×10^{-12} (20% error)



RPMDrate

config

EFS

MLIP
(LOTF scheme)

config

EFS

QM

Application #6: automated phase diagrams

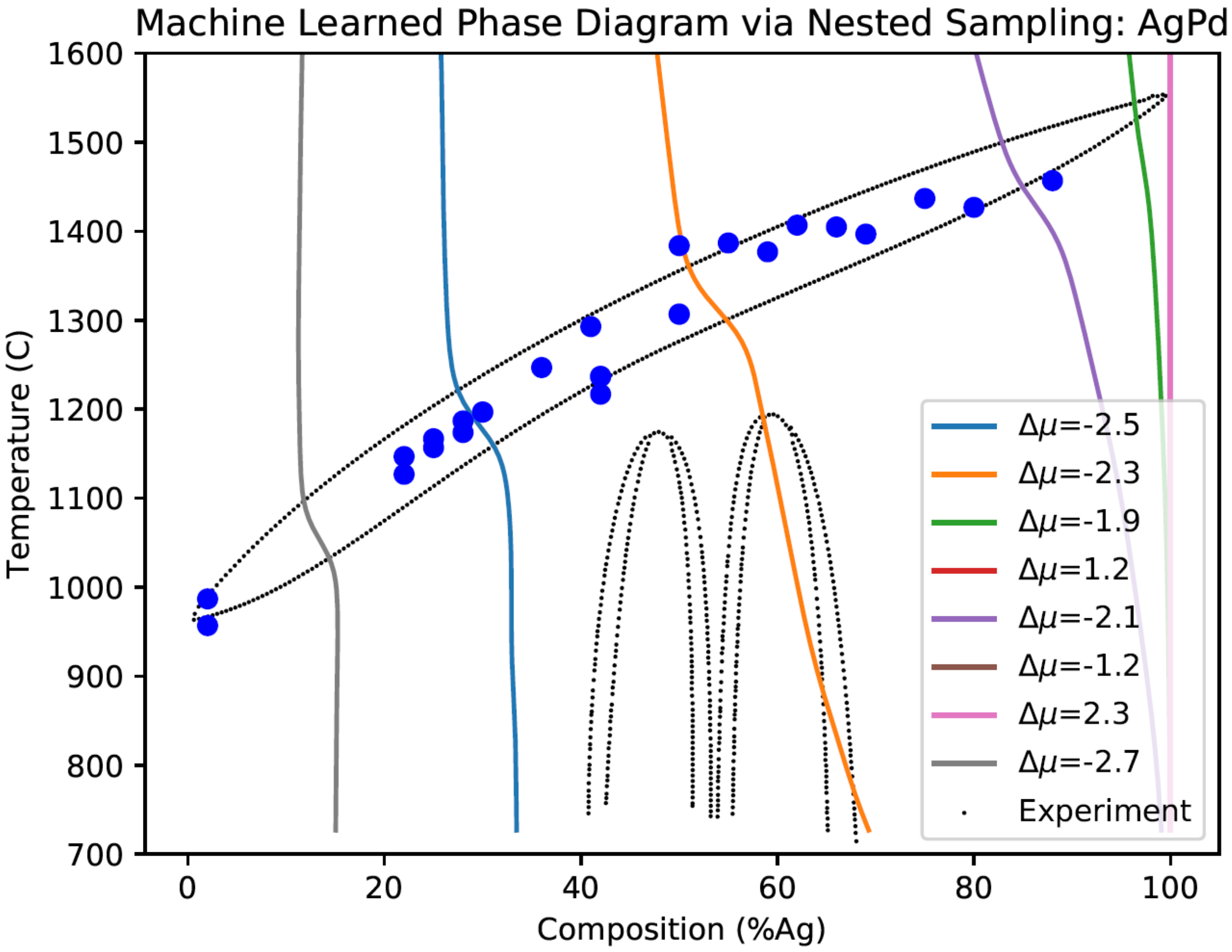
Conrad Rosenbrock,
Livia Bartok-Partay,
Noam Bernstein, K. Gubaev,
Gabor Csanyi, A.S., Gus Hart
(manuscript)

- Fitted a potential for Ag-Pd binary system (solid and liquid)

Application #6: automated phase diagrams

Conrad Rosenbrock,
Livia Bartok-Partay,
Noam Bernstein, K. Gubaev,
Gabor Csanyi, A.S., Gus Hart
(manuscript)

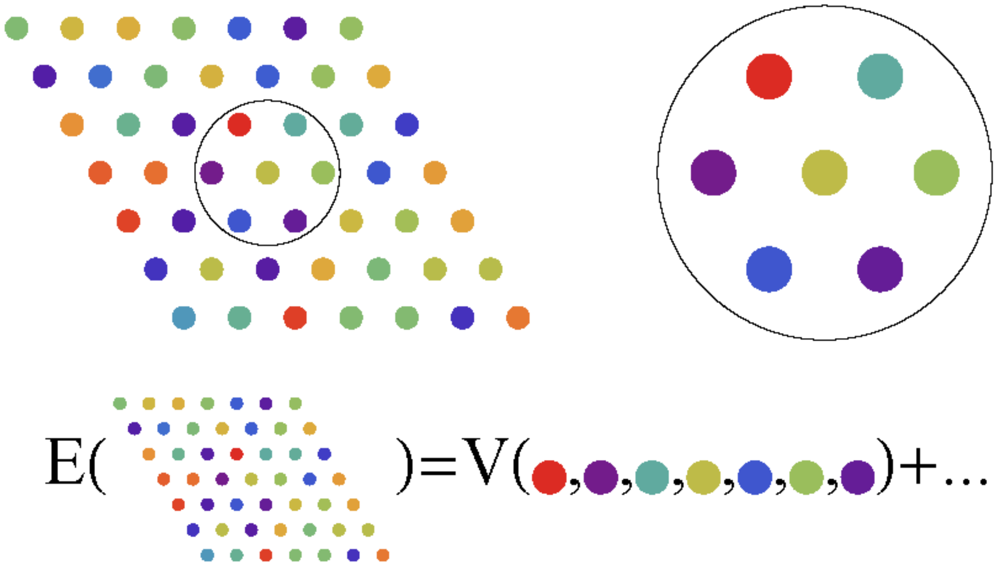
- Fitted a potential for Ag-Pd binary system (solid and liquid)

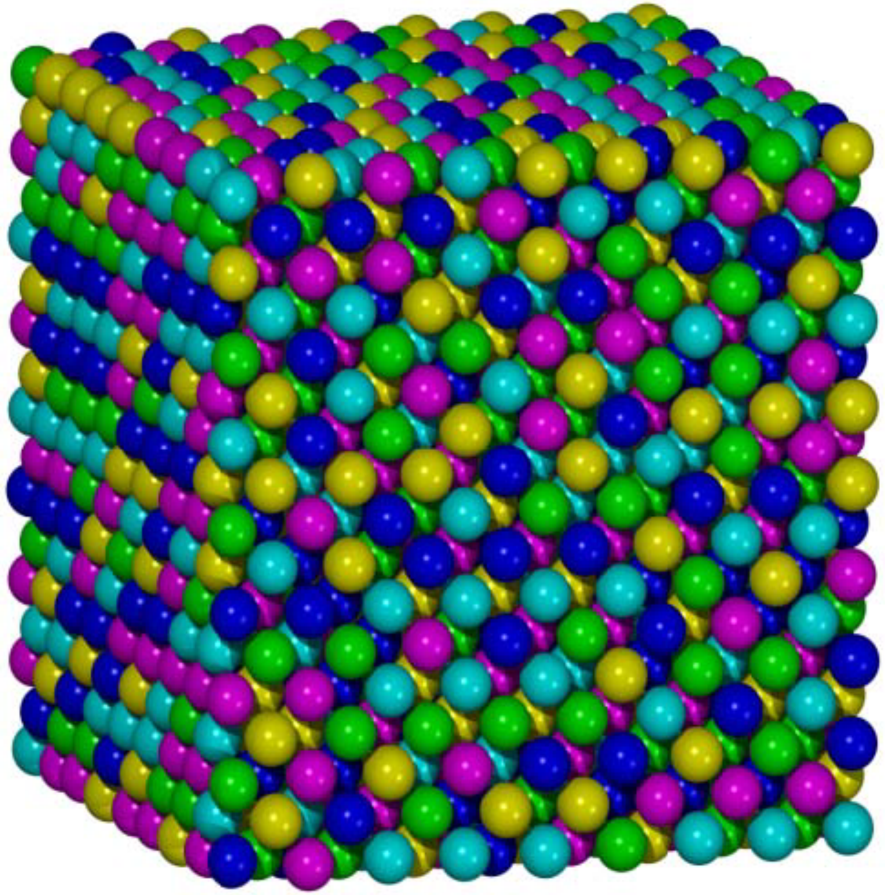


On-lattice models: HEAs

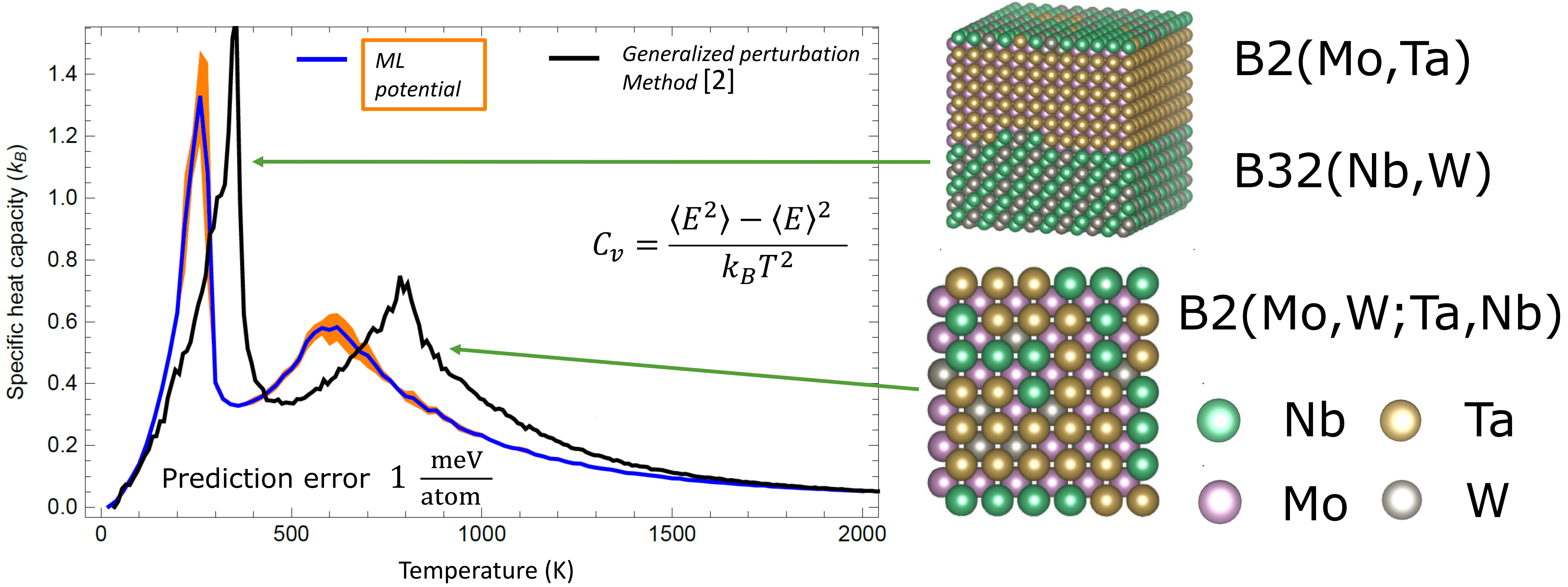
T. Kostiuchenko,
Fritz Koermann,
Joerg Neugebauer, A.S. (2019)

- Atoms of different kind sit in the lattice sites.
- Problem: predict the interatomic interaction energy (formation energy, mixing enthalpy)





Comparison with existing methods: without local lattice distortions

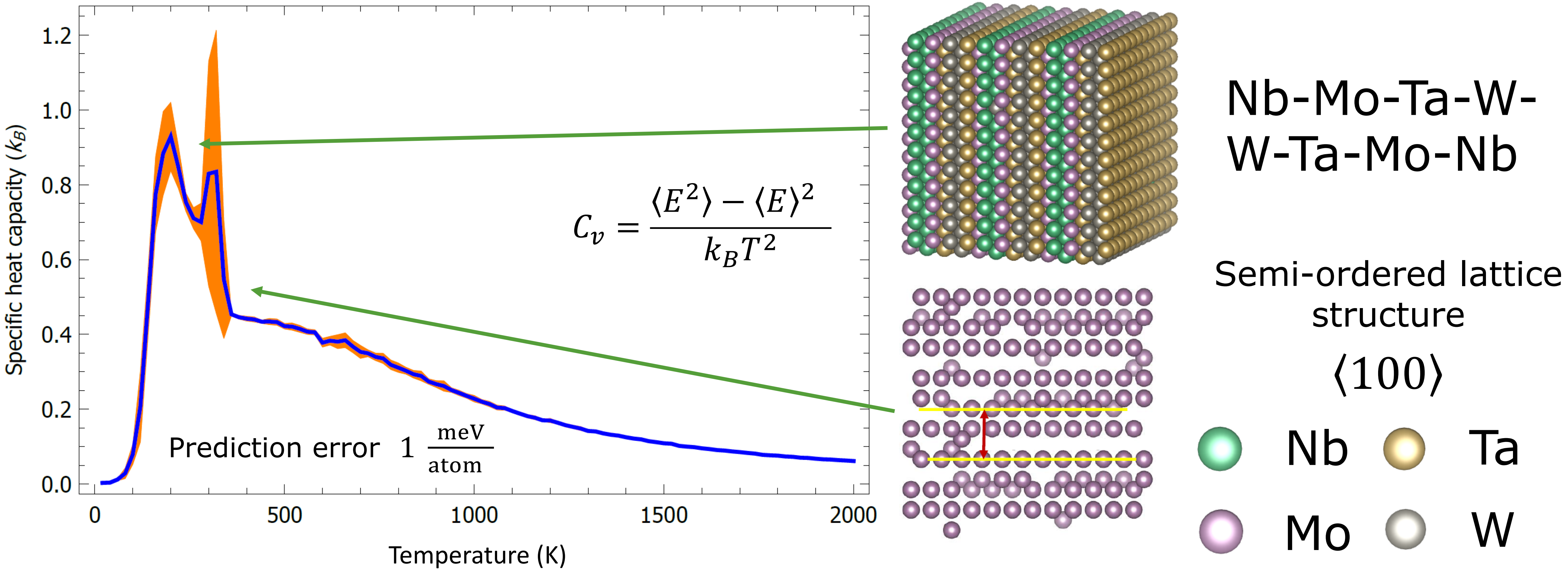


$$C_v = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}$$

Prediction error $1\ \frac{\mathrm{meV}}{\mathrm{atom}}$

[2] Fritz Körmann, Andrei V Ruban, and Marcel HF Sluiter. Long-ranged interactions in bcc NbMoTaW high-entropy alloys. Materials Research Letters, 5(1):35-40, 2017.

Results & discussion: accounting for local lattice distortions



$$C_v = \frac{\langle E^2\rangle - \langle E\rangle^2}{k_B T^2}$$

Prediction error $1\ \frac{\text{meV}}{\text{atom}}$

Nb-Mo-Ta-W-W-Ta-Mo-Nb

Semi-ordered lattice structure

$\langle 100\rangle$

Summary: MLIP Code

- Public version: http://mlip.skoltech.ru/
  - developer’s version (incl. unpublished capabilities) by request
- QM model interfaces:
  - VASP, Gaussian (DFT)
  - PROFESS (OFDFT)
- Atomistic Driver interfaces:
  - LAMMPS, serial and parallel (but no learning on the fly)
  - USPEX
  - ASE
  - RPMDrate
- Active learning / Learning on the fly

Related fields

- Learning Potential Energy Surfaces (PES) of molecules
    - Similar idea, but no locality. Can be described, e.g., by all N (N+1)/2 pairwise distances in the system
    - Older field (started before 2000),
    - first time neural networks were applied to chemistry
- Cheminformatics
    - Structure-property relations – learning things other than energy
    - next lecture