Machine-learning interatomic potentials

an automated tool of accelerating ab initio materials modeling

Alexander Shapeev, Skoltech

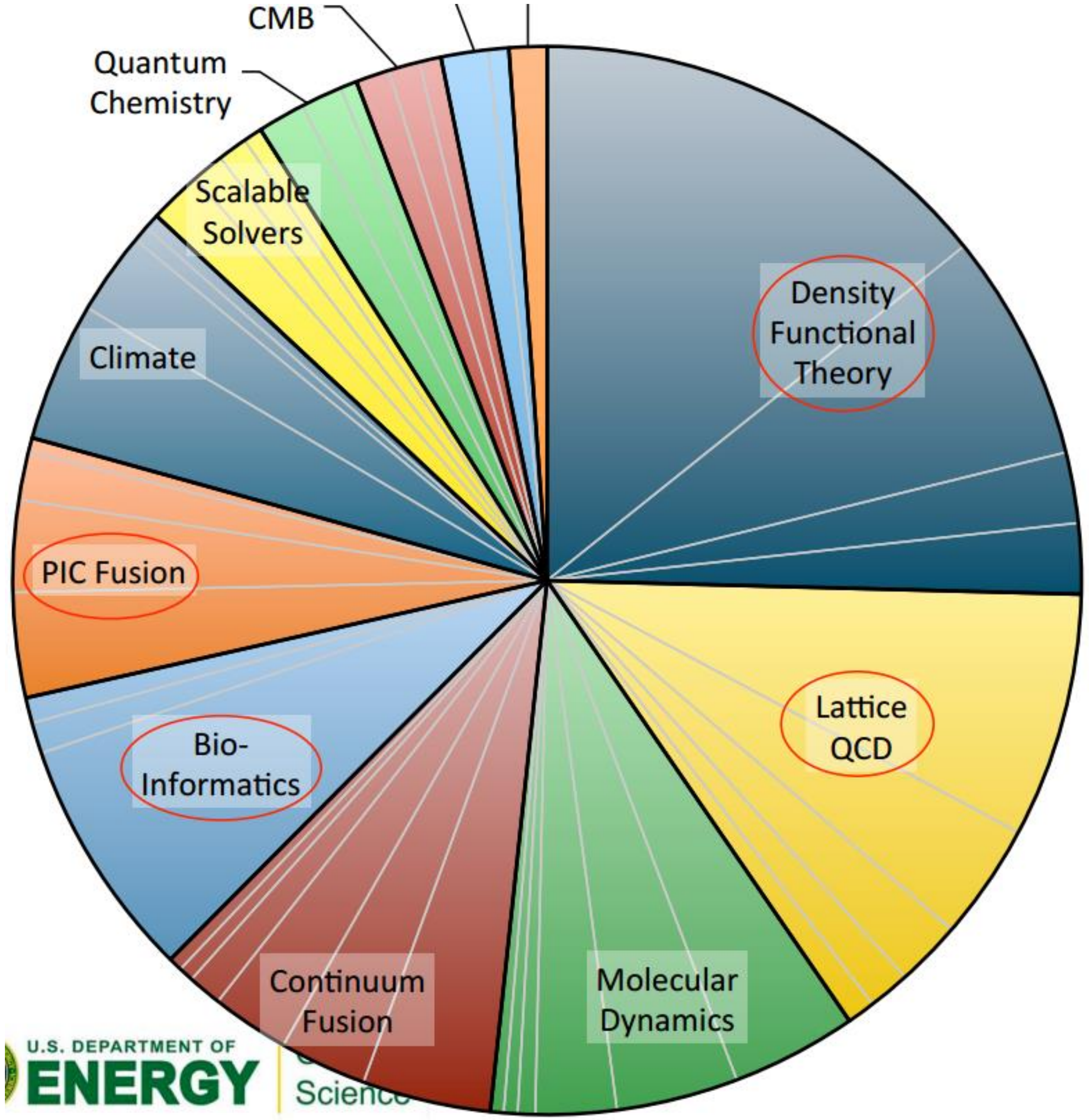
USPEX SCHOOL 2021,

organized with support from:

RSF | Russian Science Foundation

Molecular modeling

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



[Adopted from nersc.gov]

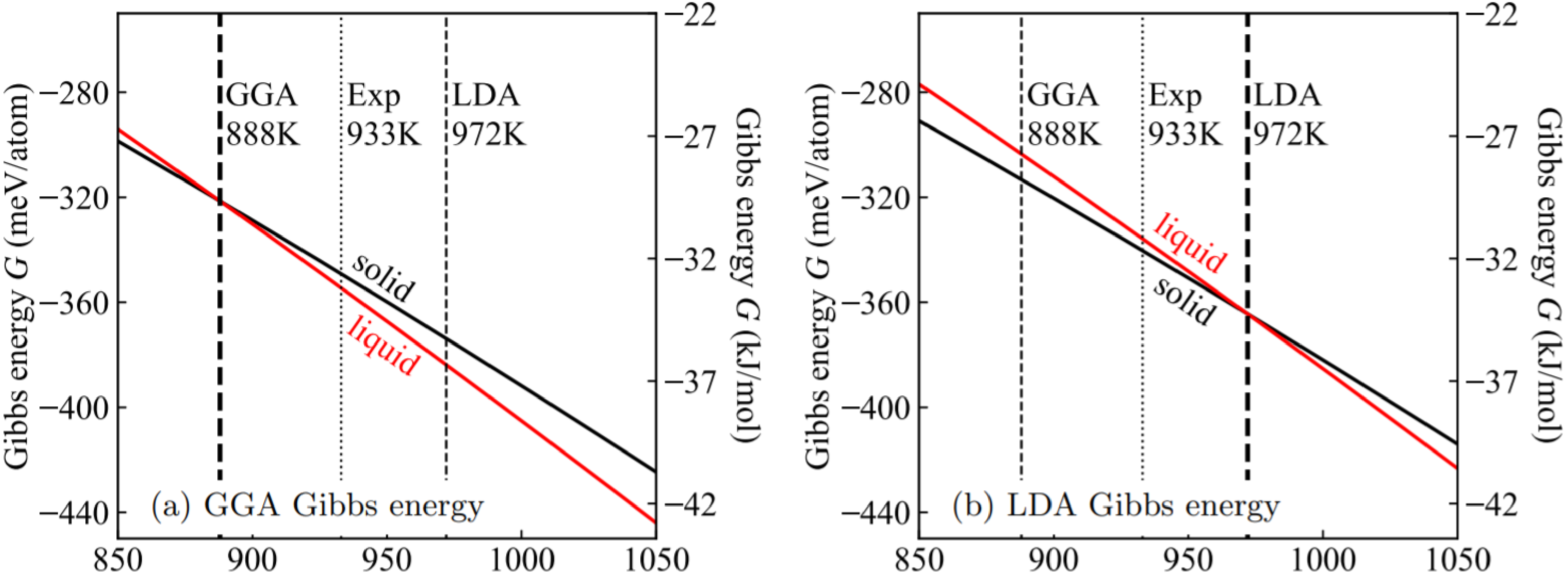
Motivation:

more and more materials properties can be computed with DFT

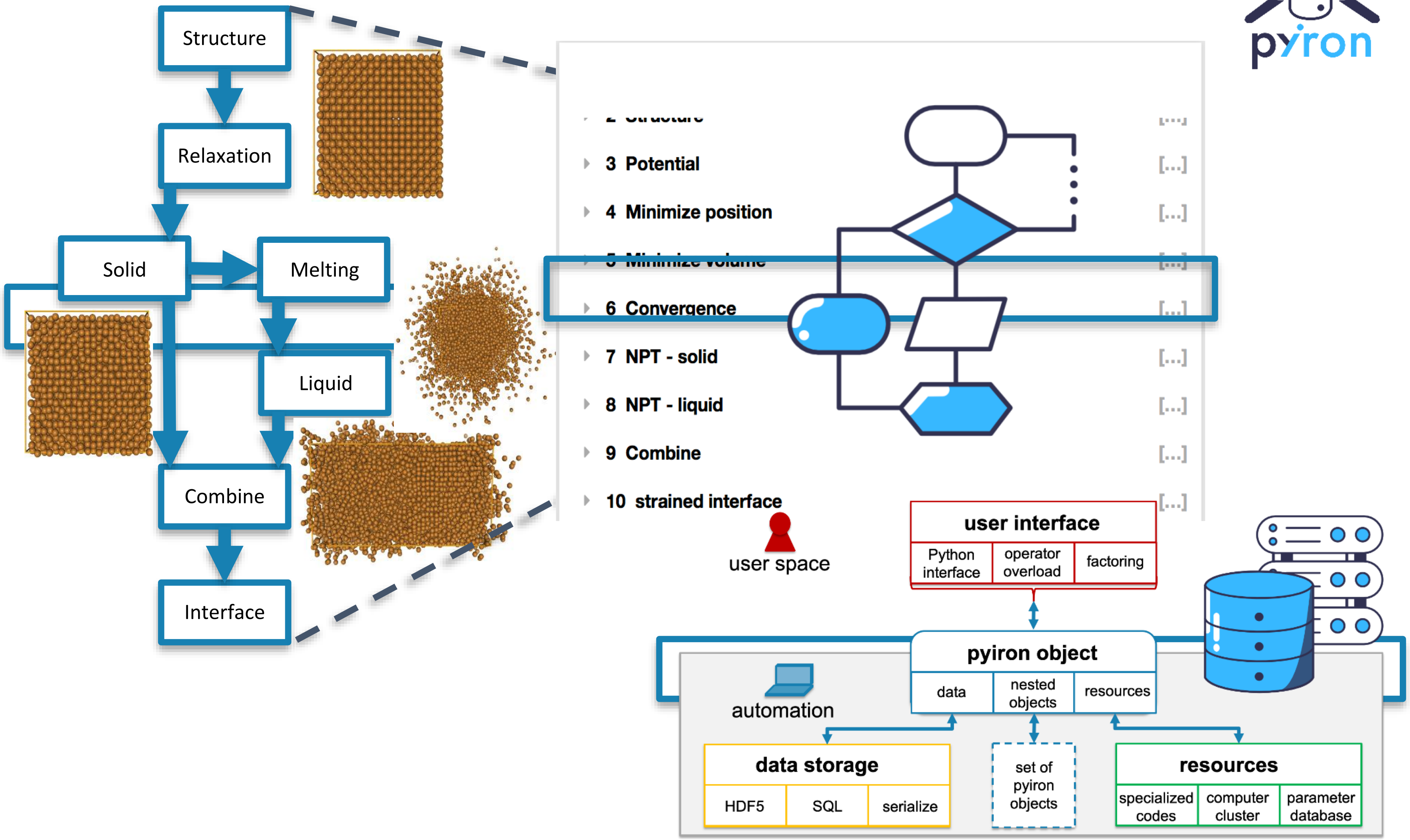
Ab initio Melting point calculation

Aluminum (8x8x8 k-point mesh):

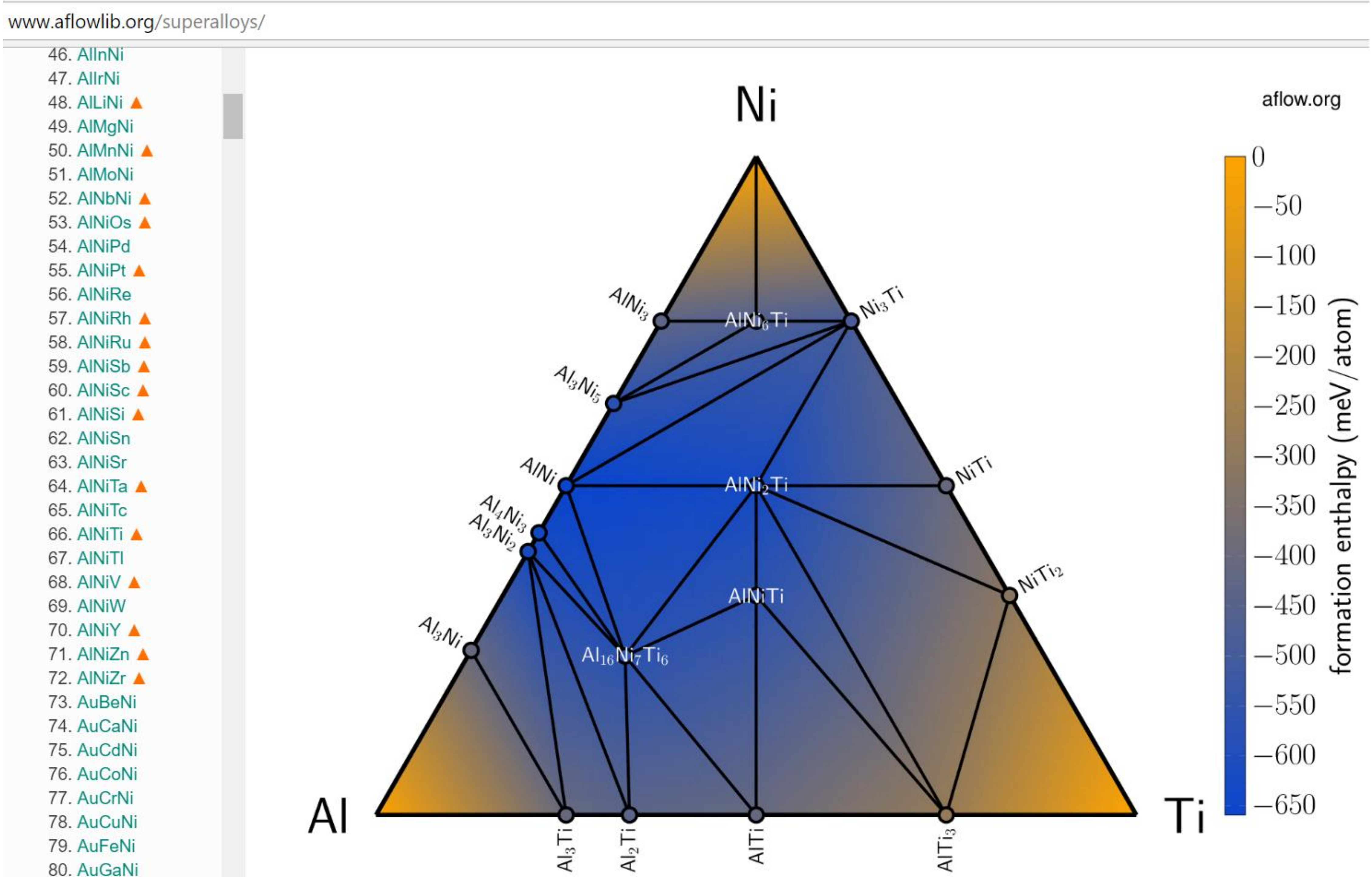
Zhu, Körmann, Ruban, Neugebauer, Grabowski (2020):



From rapid prototyping to high performance computing



J. Janssen, et al., Comp. Mat. Sci. 161 (2019) - http://pyiron.org - https://github.com/pyiron/



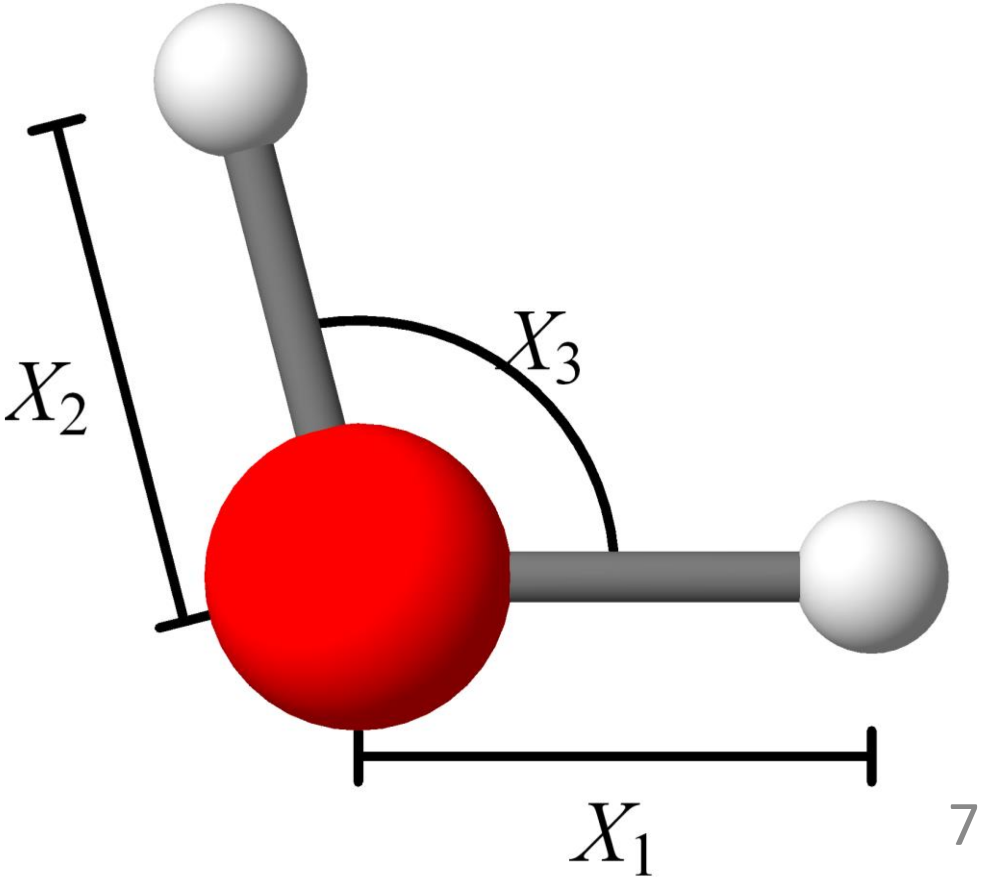
www.aflowlib.org/superalloys/

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

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)

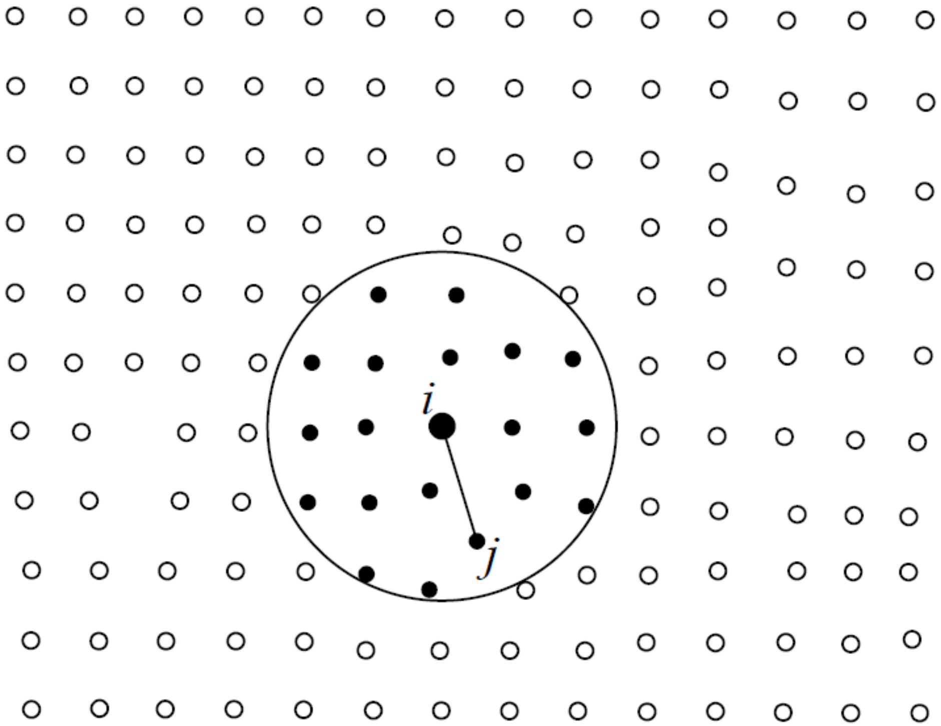


Locality: Energy

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

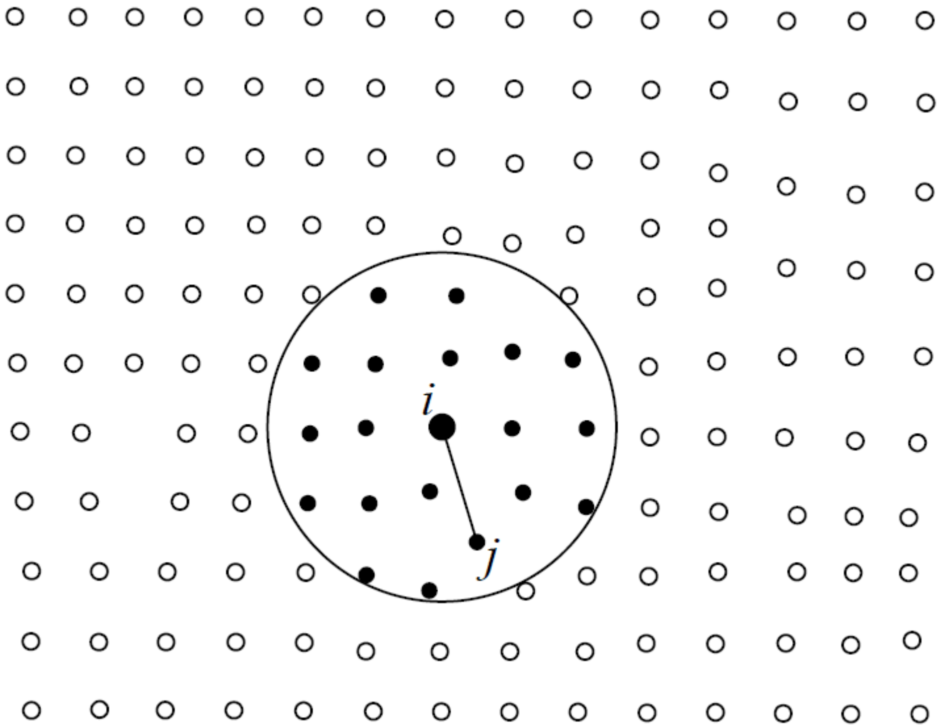
- Most interatomic potentials are covered. (Coulomb should be added explicitly.)

- Problem: find a good *V.*



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*?



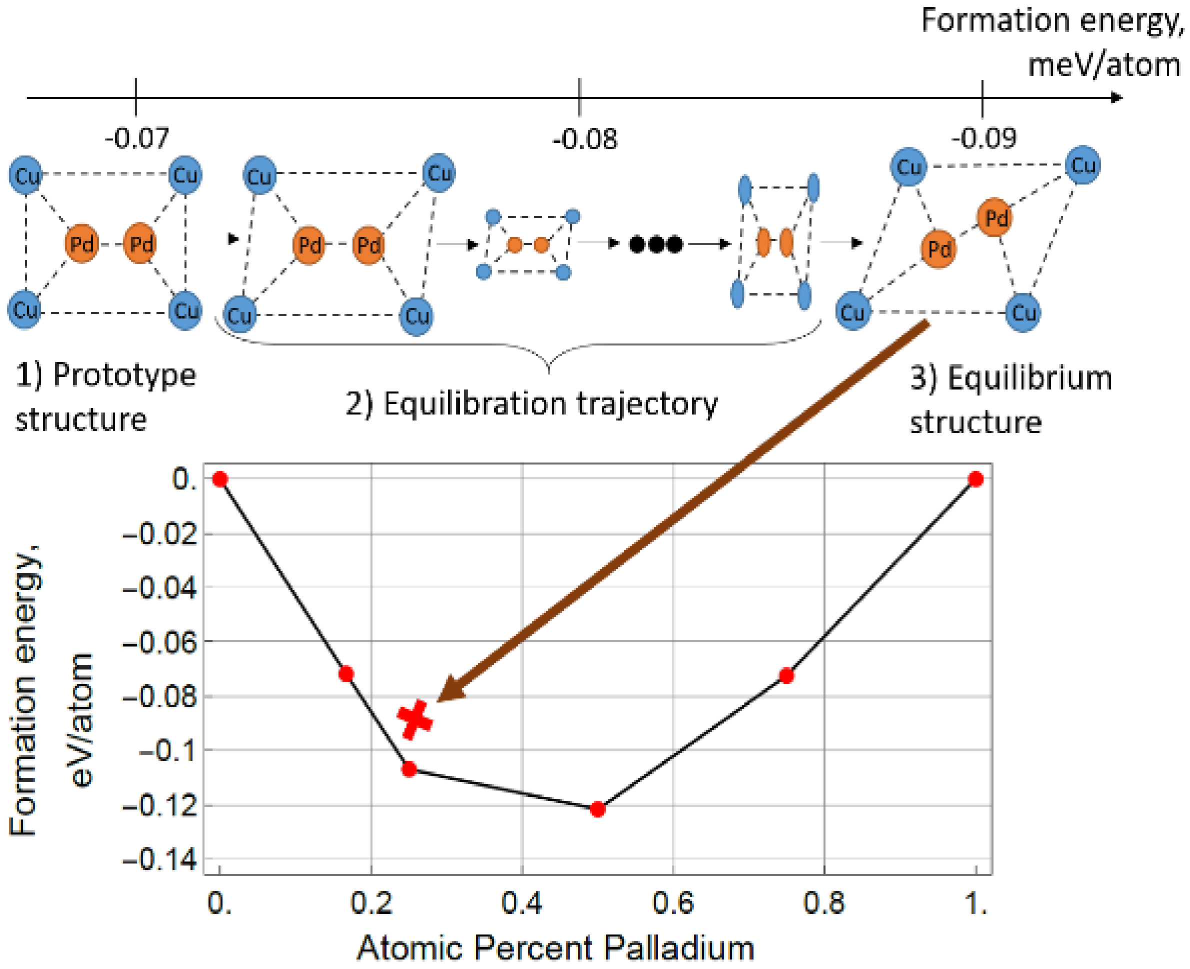
Machine-learning ideology:

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

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

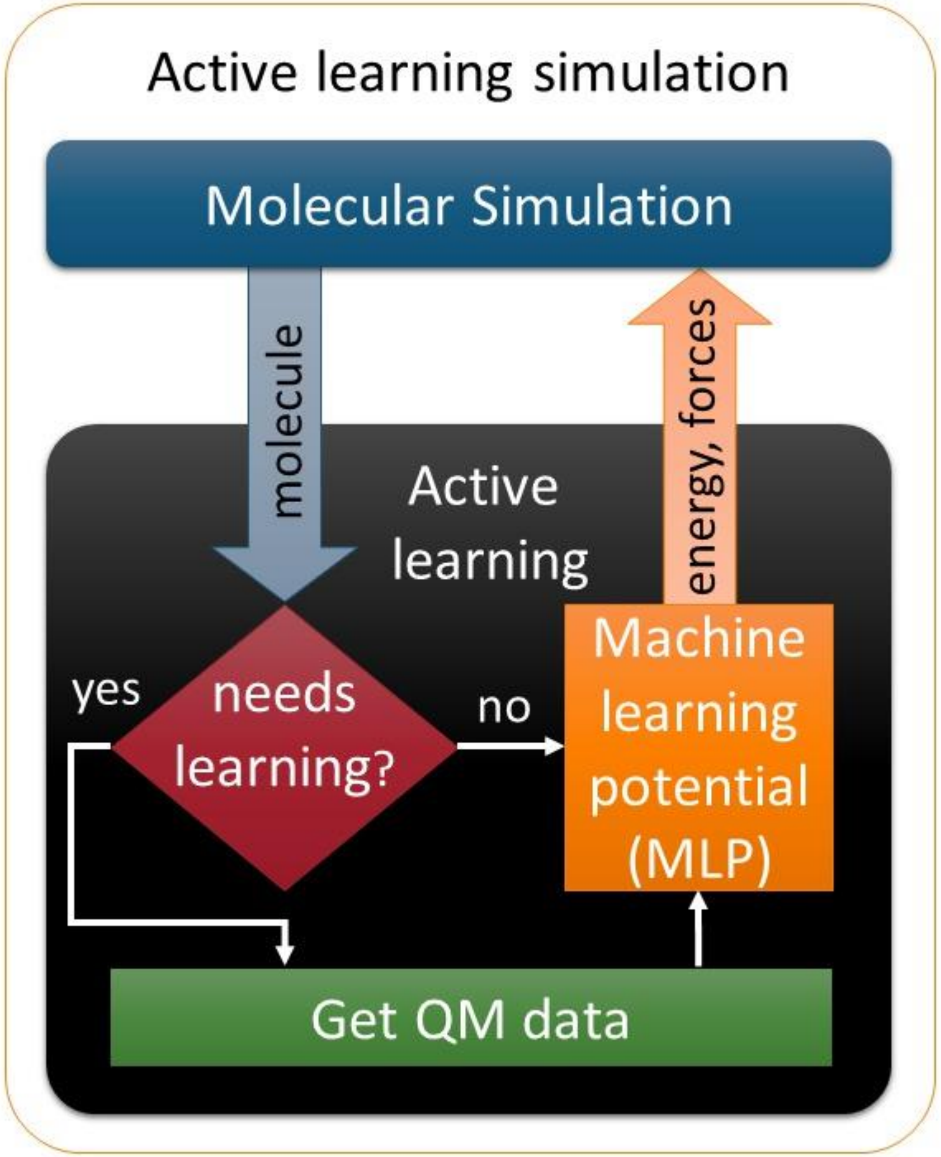
- accurate approximation of PES

which needs

- accurate sampling of ground state structures

which needs …

Solution: Active learning / Learning on-the-fly



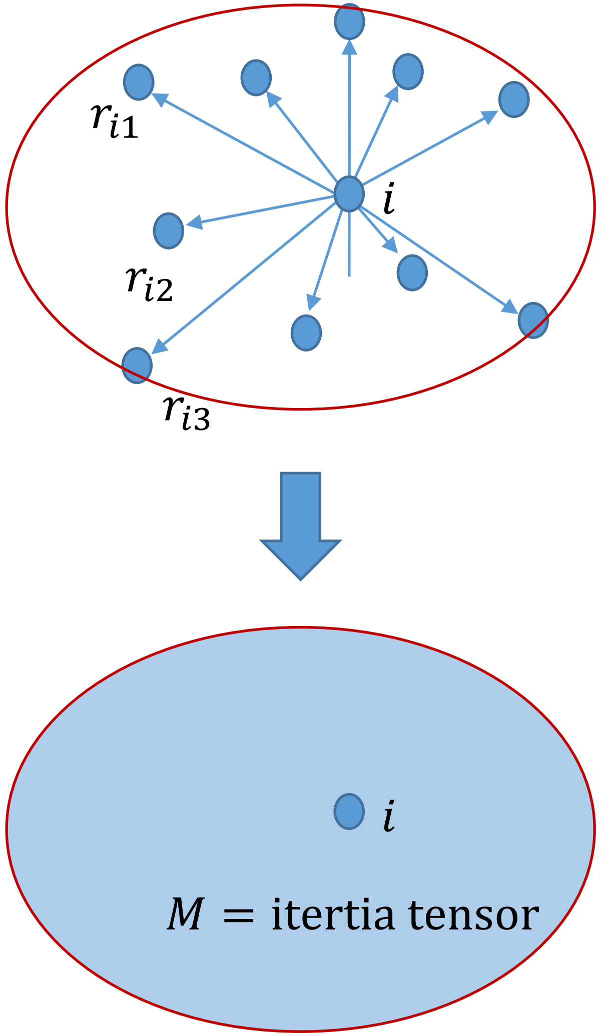
Moment Tensor Potentials: descriptors

Descriptors of atomic environments:

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

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 an (over-)complete basis

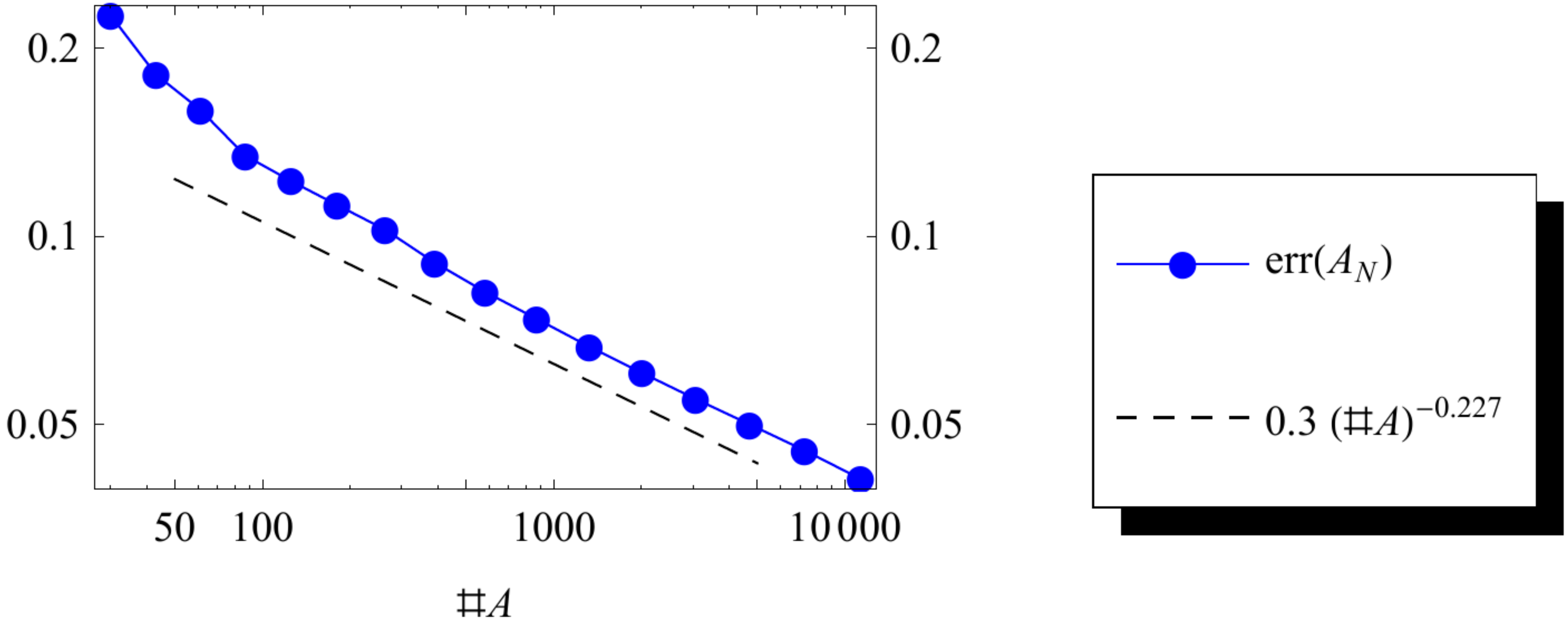
Similar to Atomic Cluster Expansion [*Drautz (2019)*], see

[Bachmayr, Csanyi, Dusson, Etter, van der Oord, Ortner (2020)]

Learning curves

Database (Csanyi, Bartok, Szlachta, 2014)

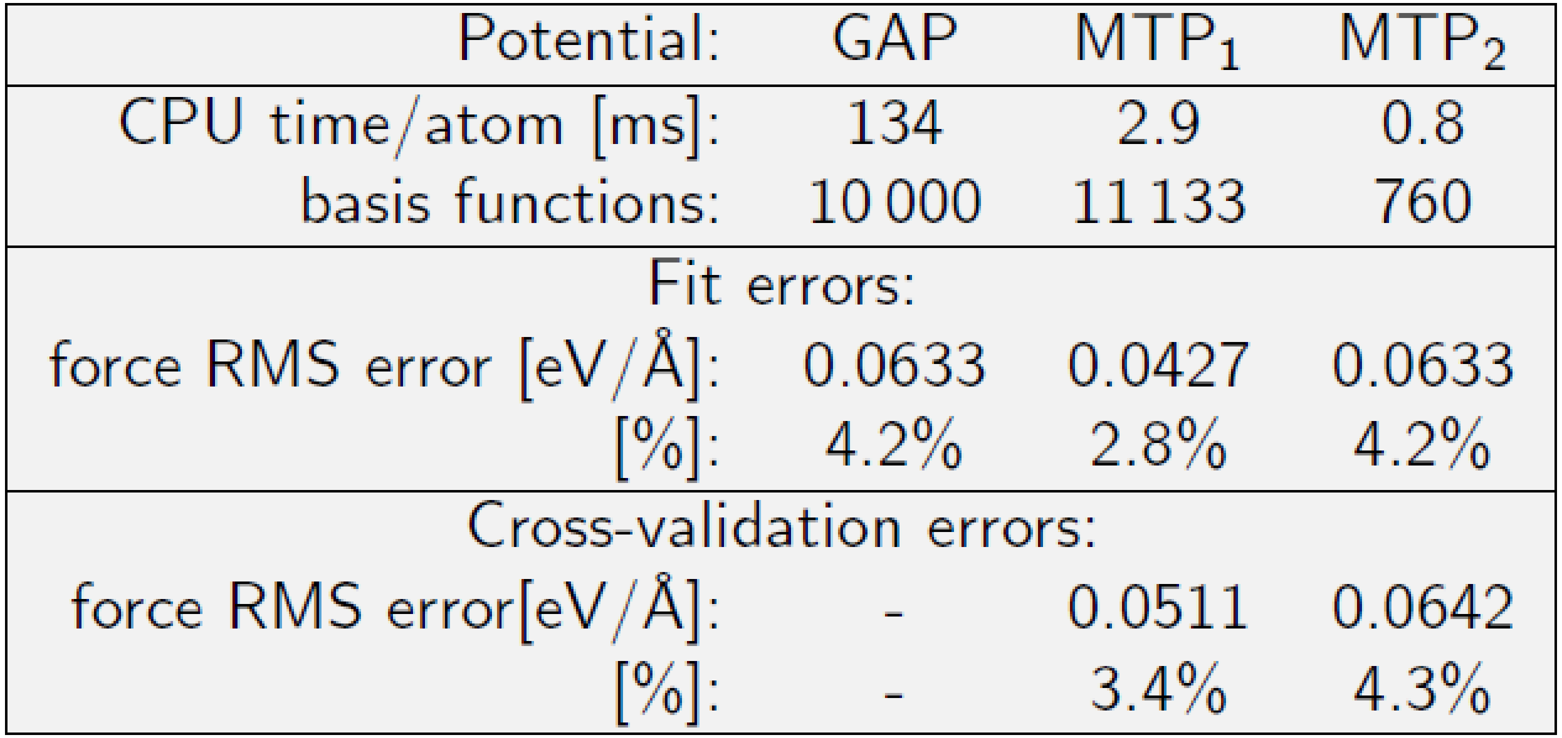
- Tungsten: uniform and perturbed lattices, vacancies, dislocations



Performance tests

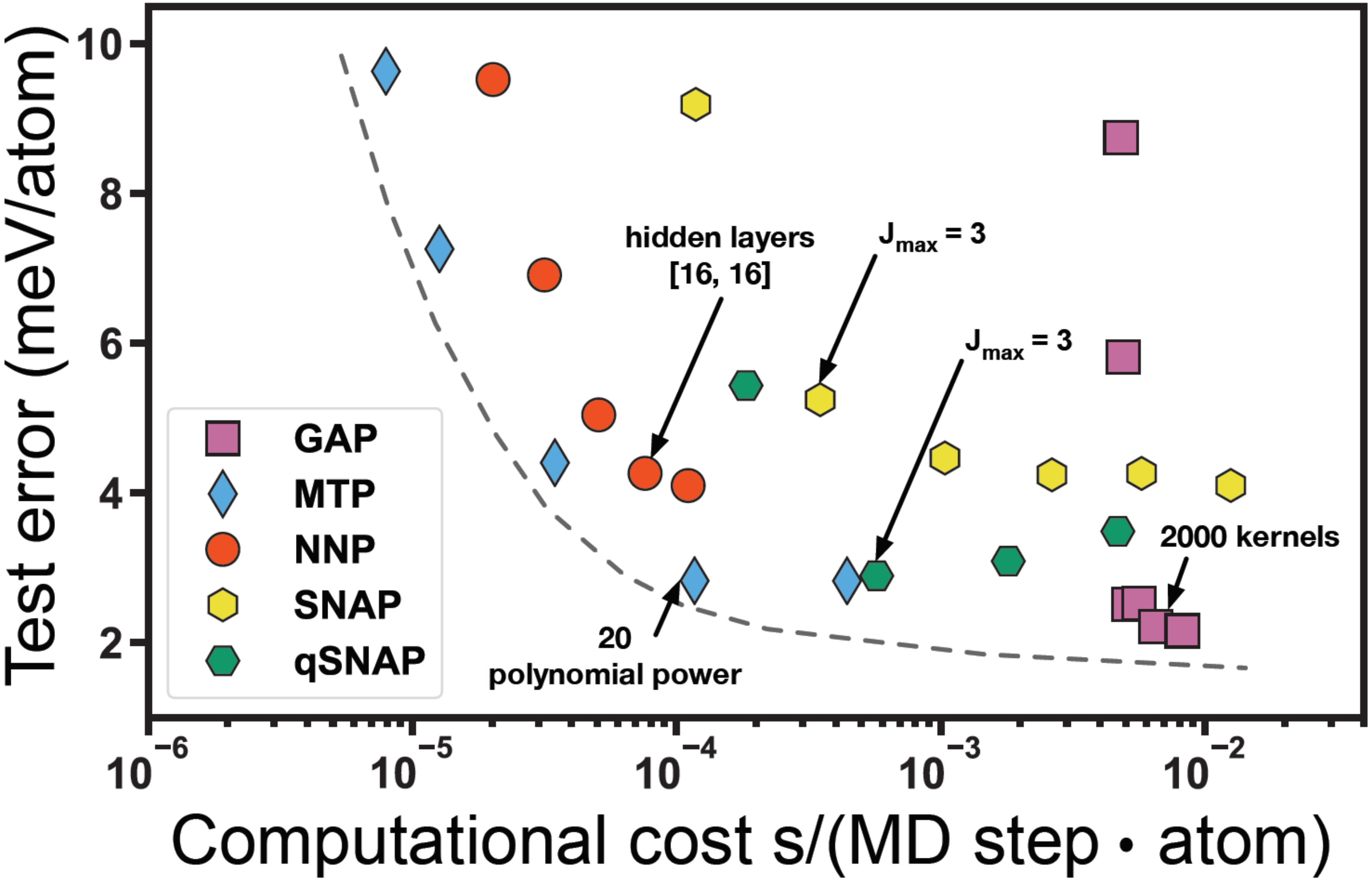
Database (Csanyi, Bartok, Szlachta, 2014)

- 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%

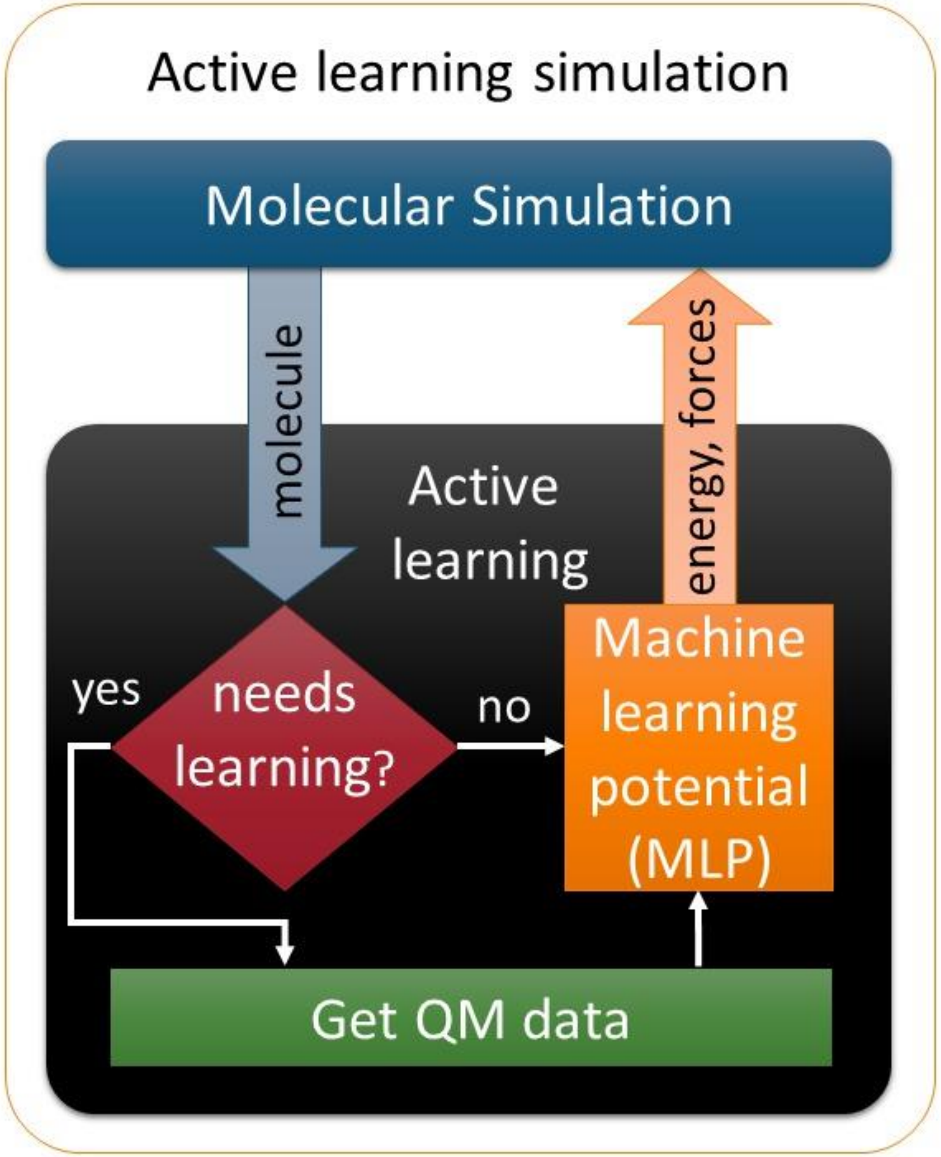
Comparison with more methods



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

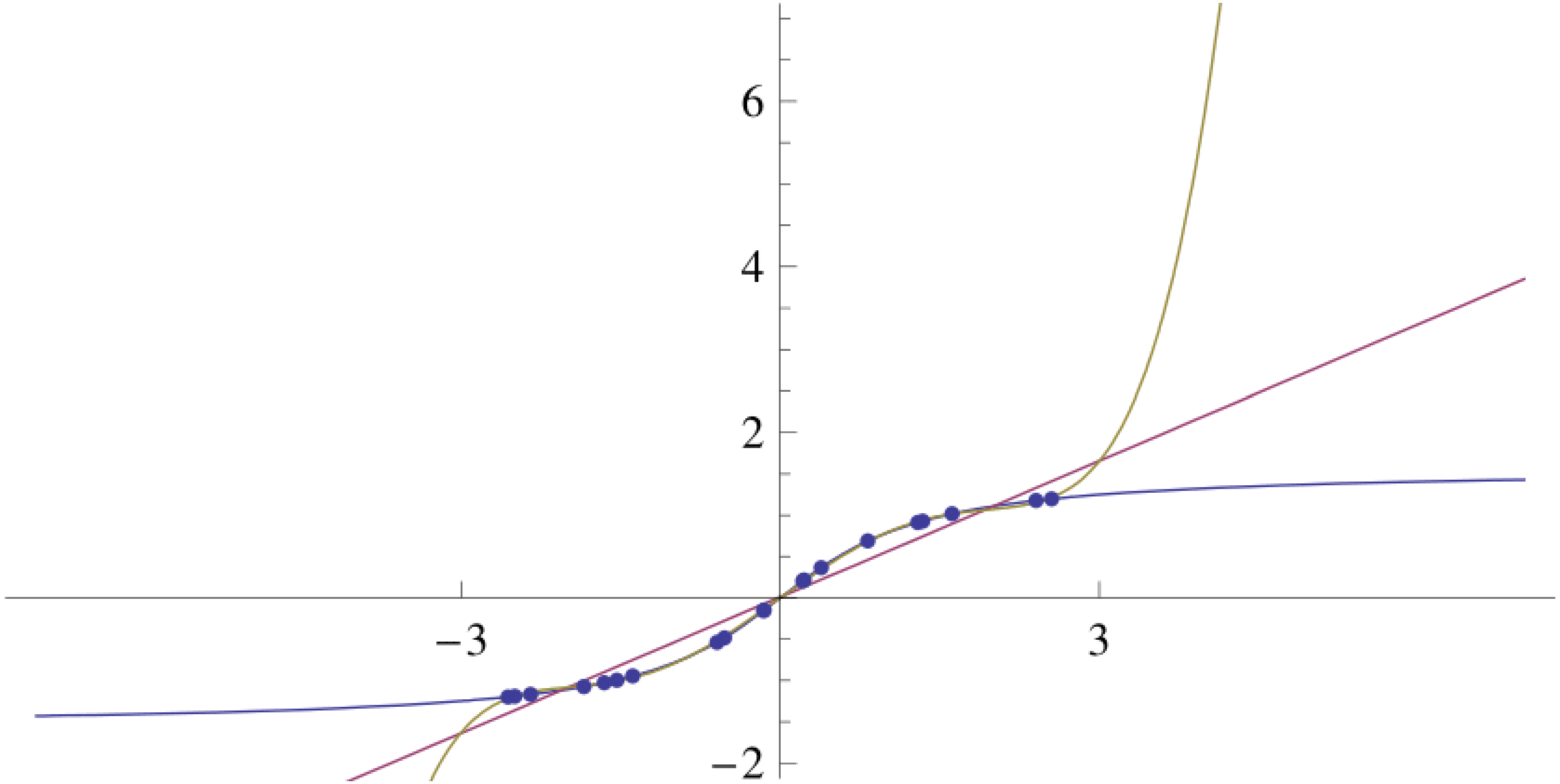
Active Learning of Interatomic Potentials

Active learning



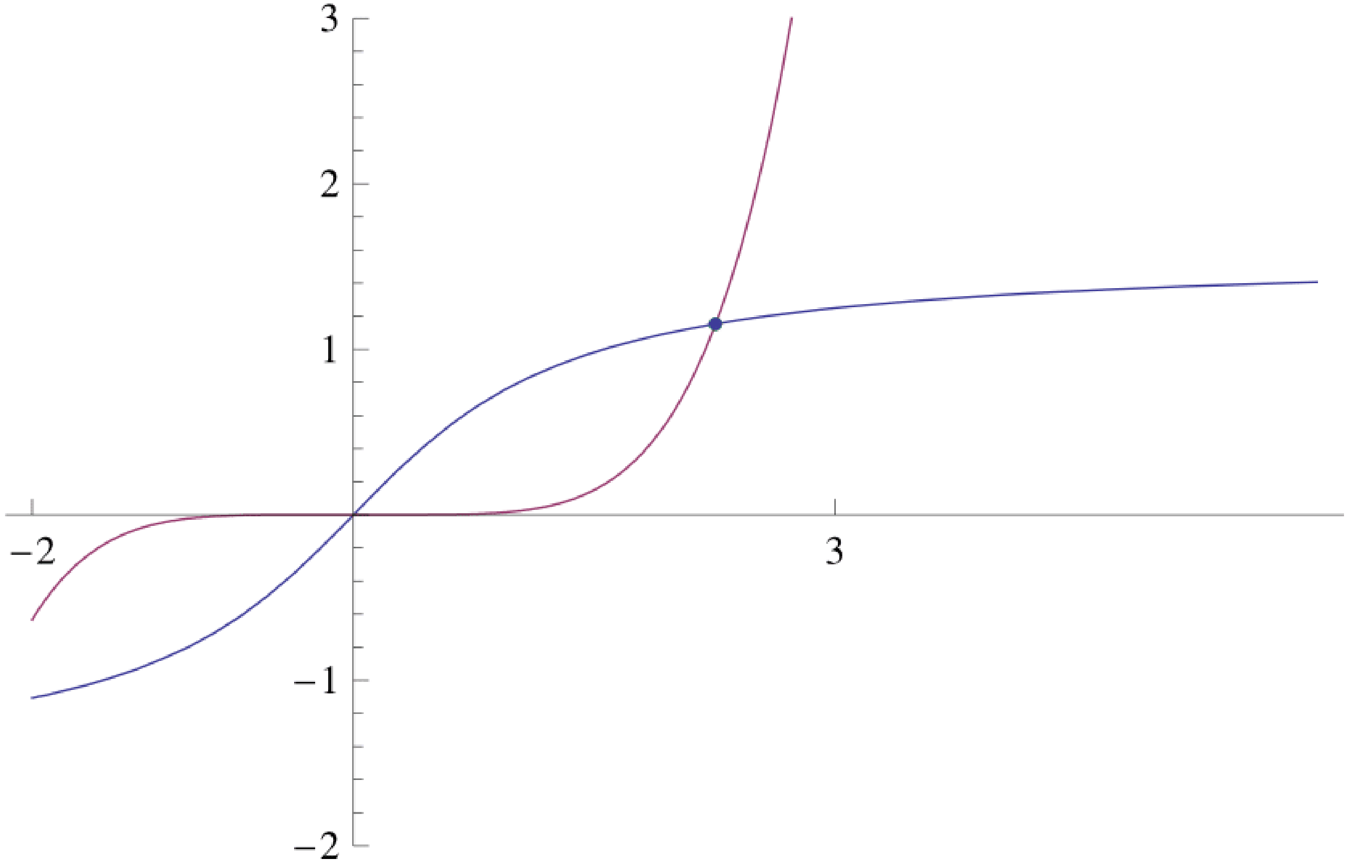
Active Learning of MLIP: Motivation

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



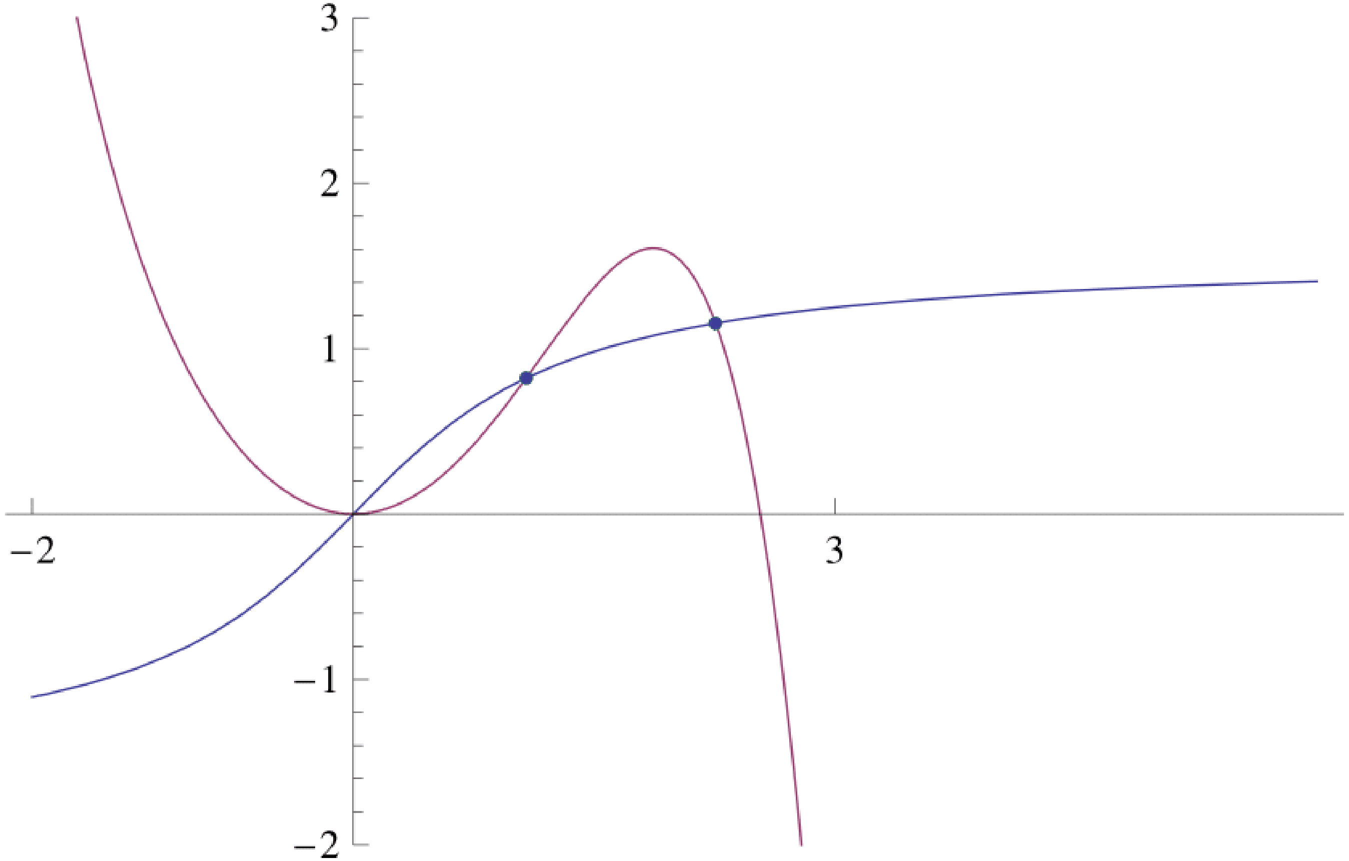
Active learning

Solution: detect when we are extrapolating and switch on learning



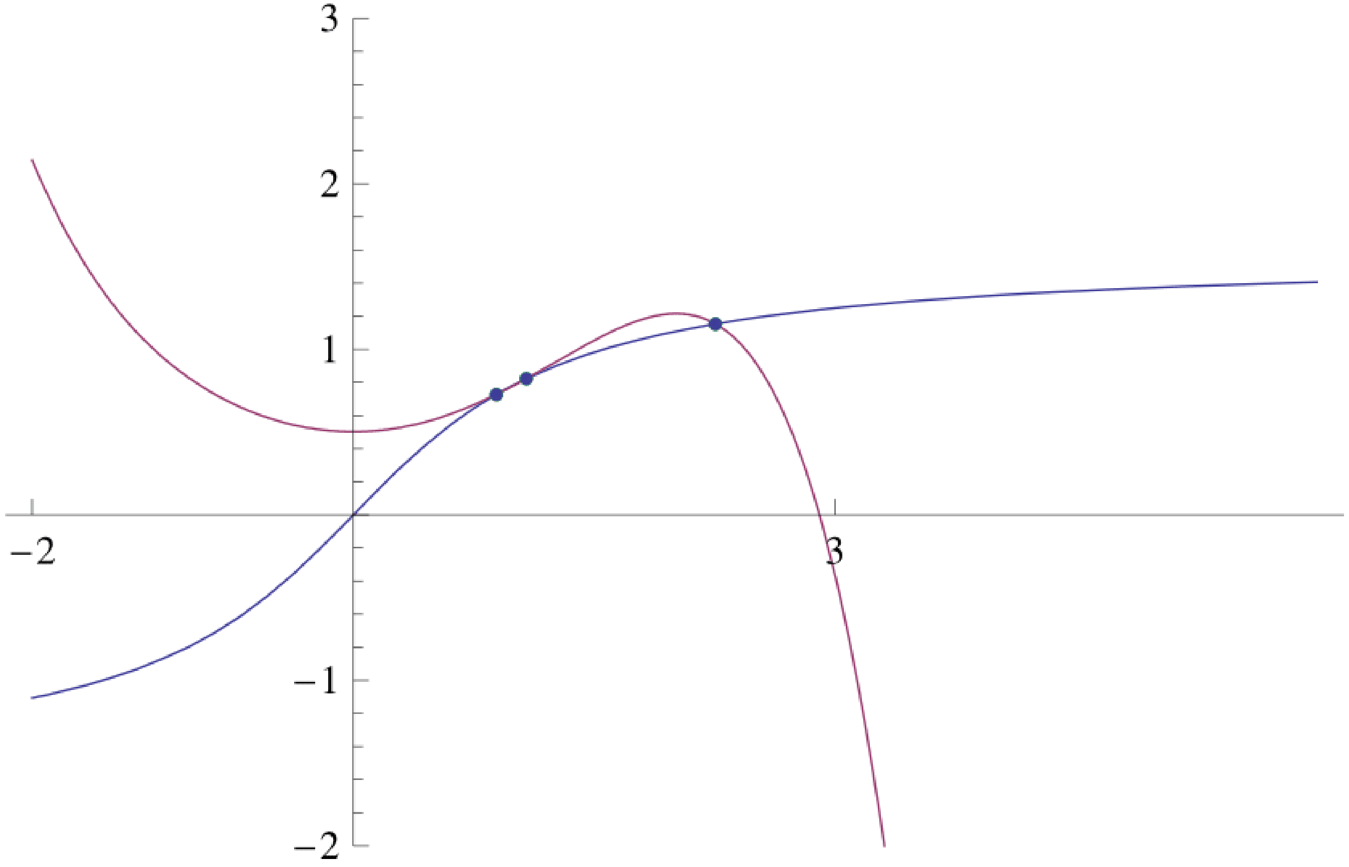
Active learning

Solution: detect when we are extrapolating and switch on learning



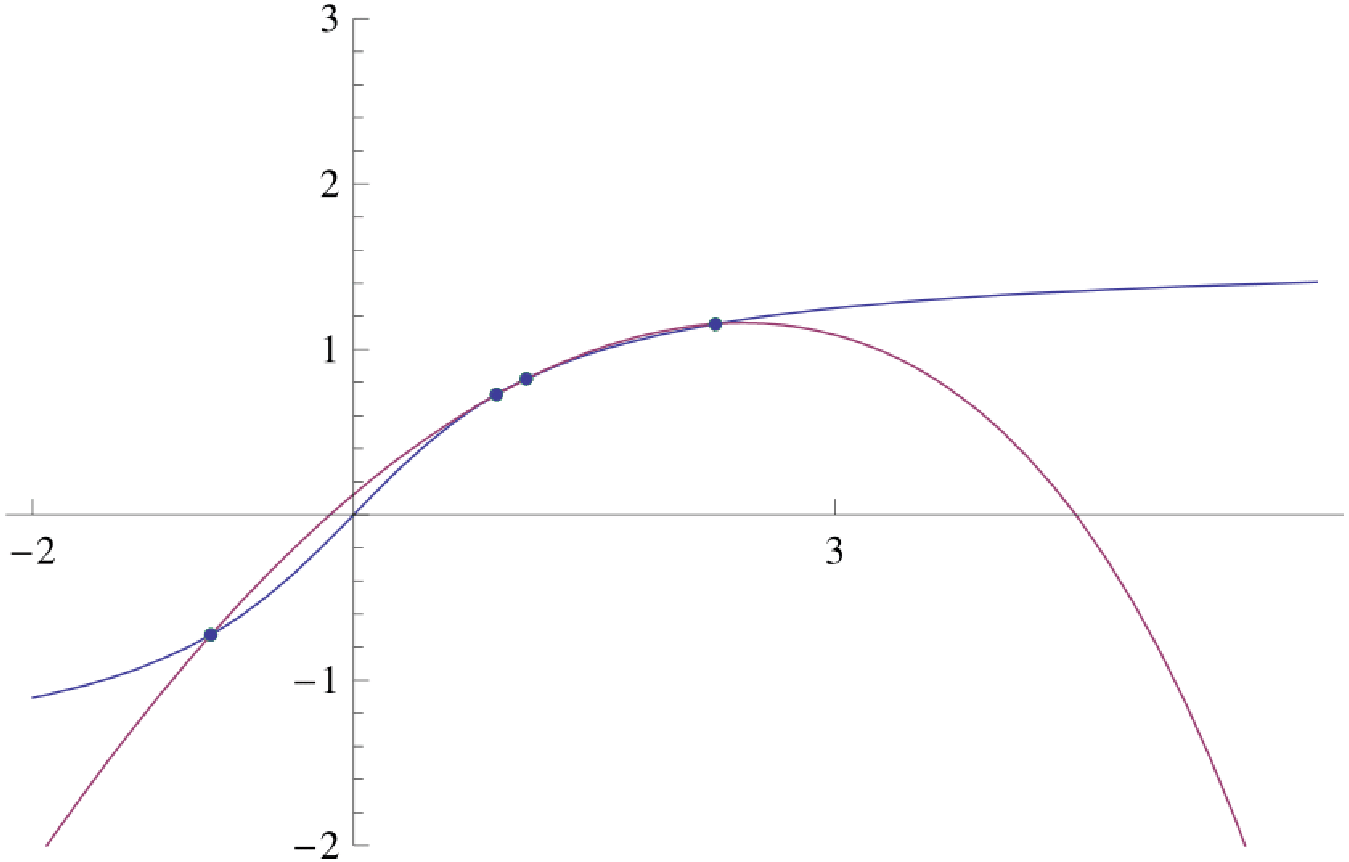
Active learning

Solution: detect when we are extrapolating and switch on learning



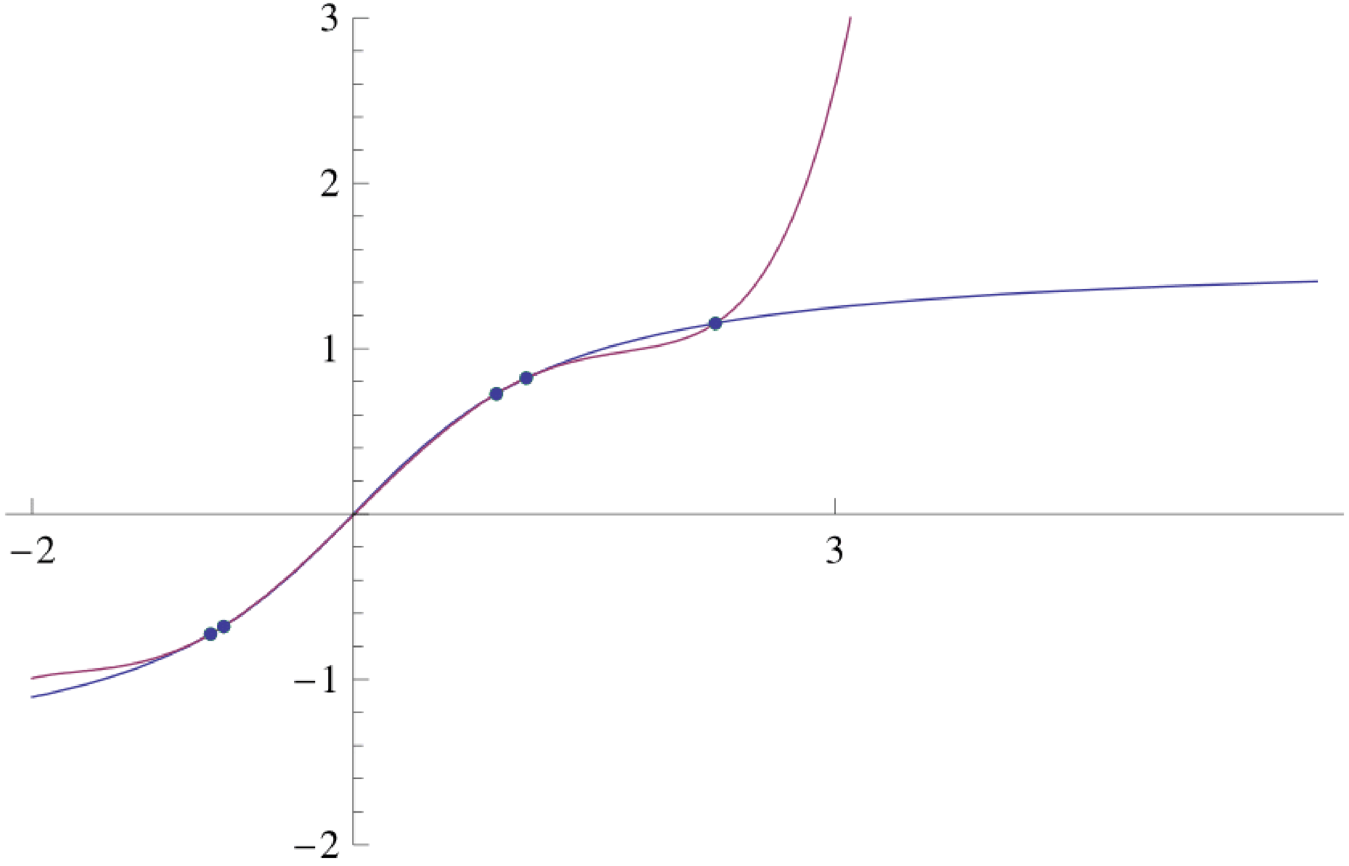
Active learning

Solution: detect when we are extrapolating and switch on learning



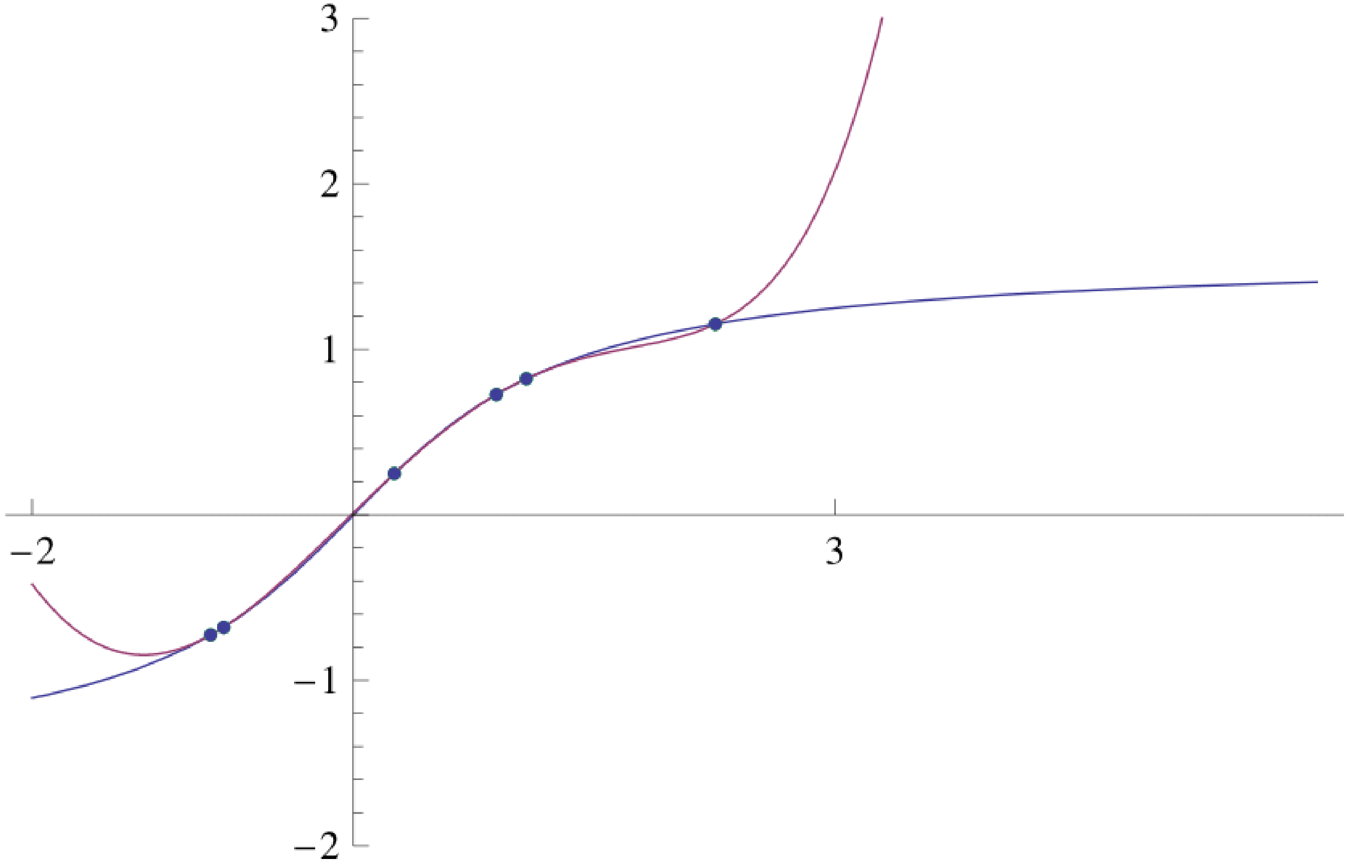
Active learning

Solution: detect when we are extrapolating and switch on learning



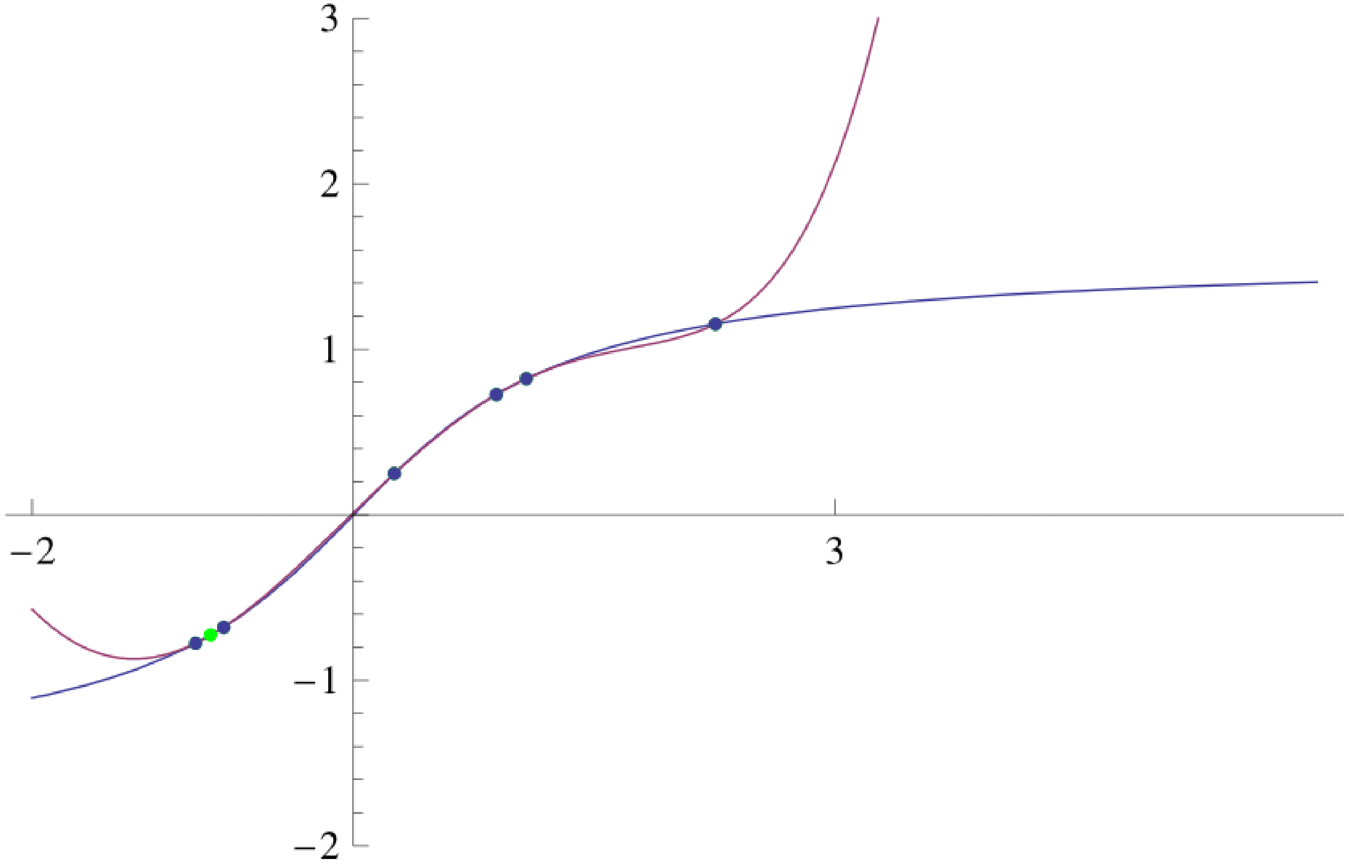
Active learning

Solution: detect when we are extrapolating and switch on learning



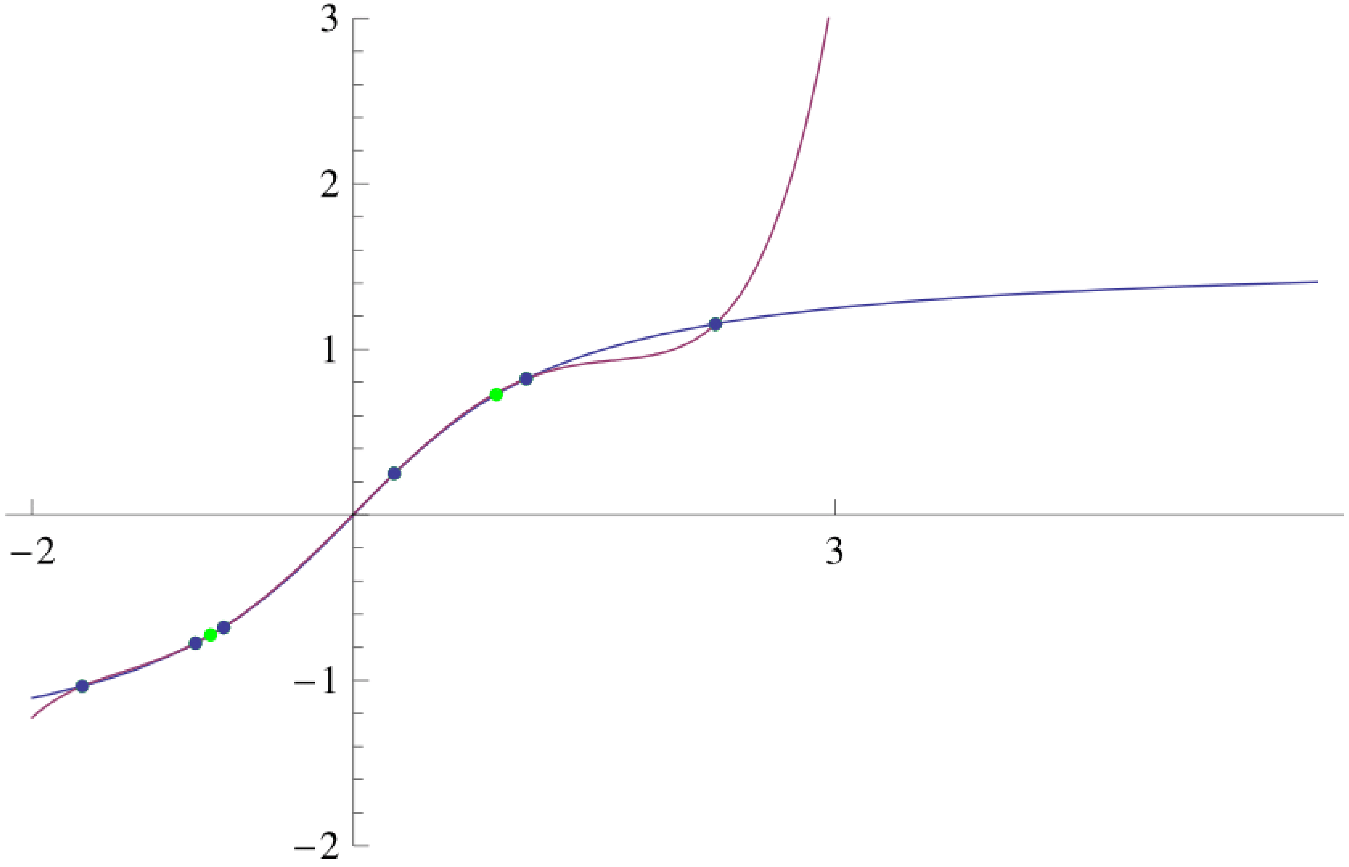
Active learning

Solution: detect when we are extrapolating and switch on learning



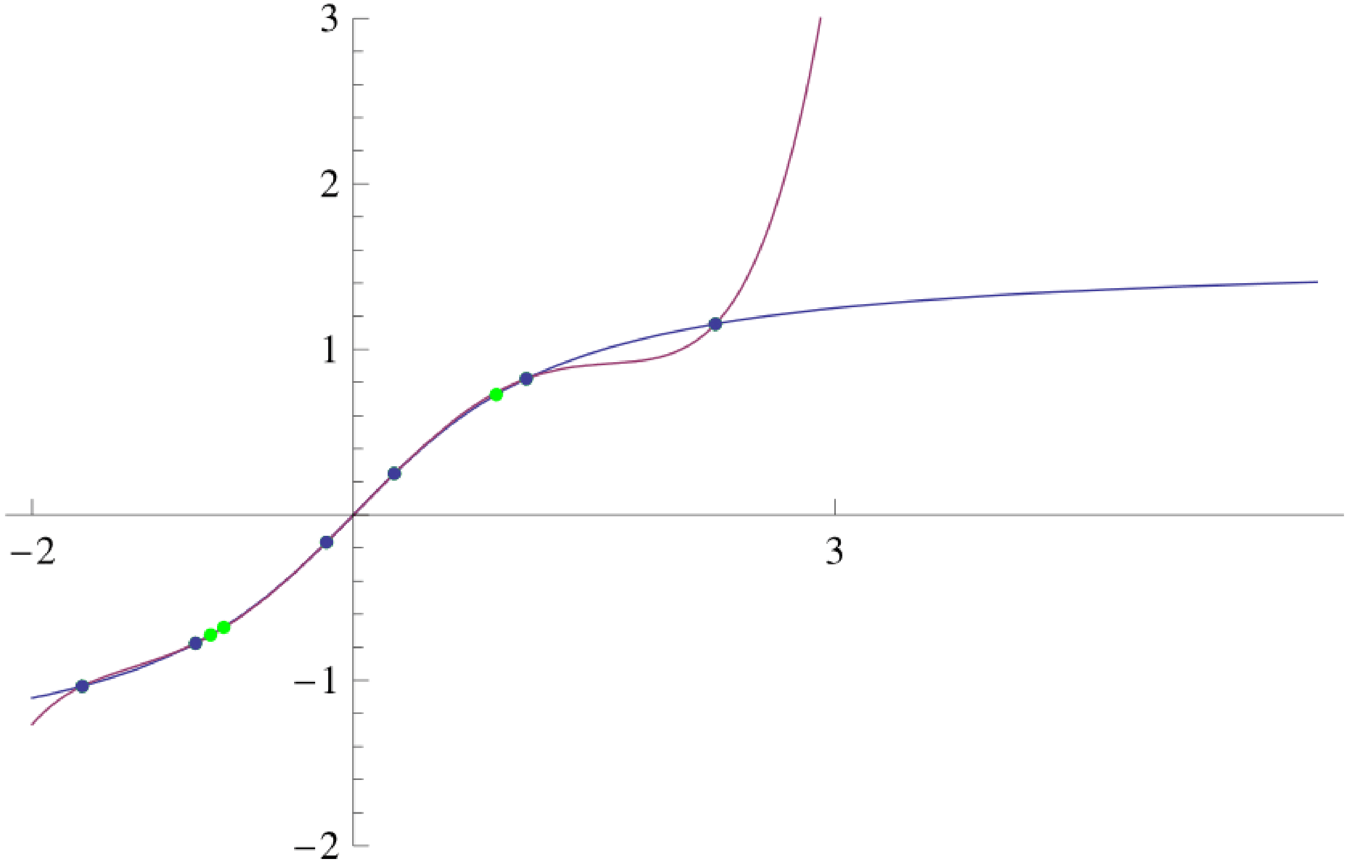
Active learning

Solution: detect when we are extrapolating and switch on learning



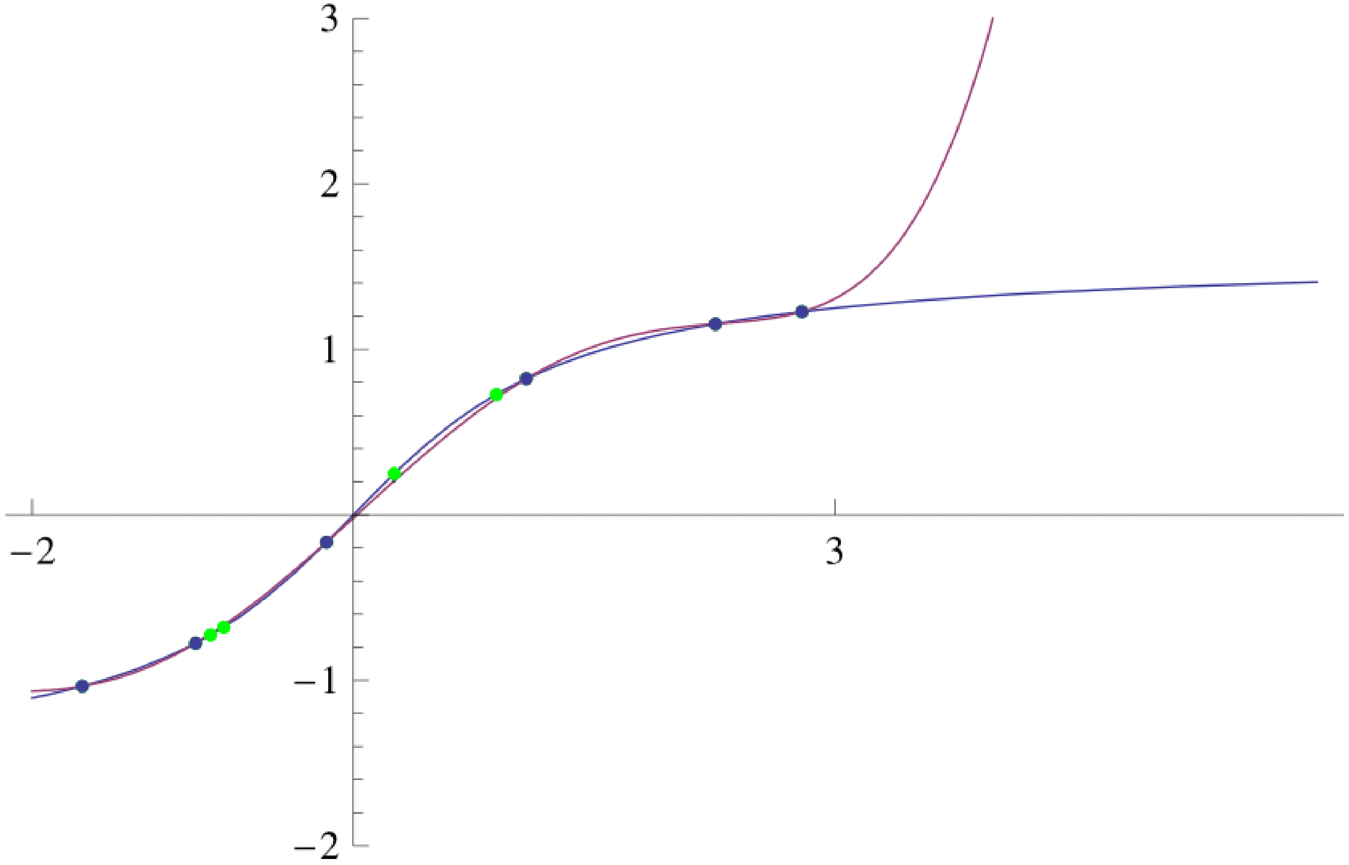
Active learning

Solution: detect when we are extrapolating and switch on learning



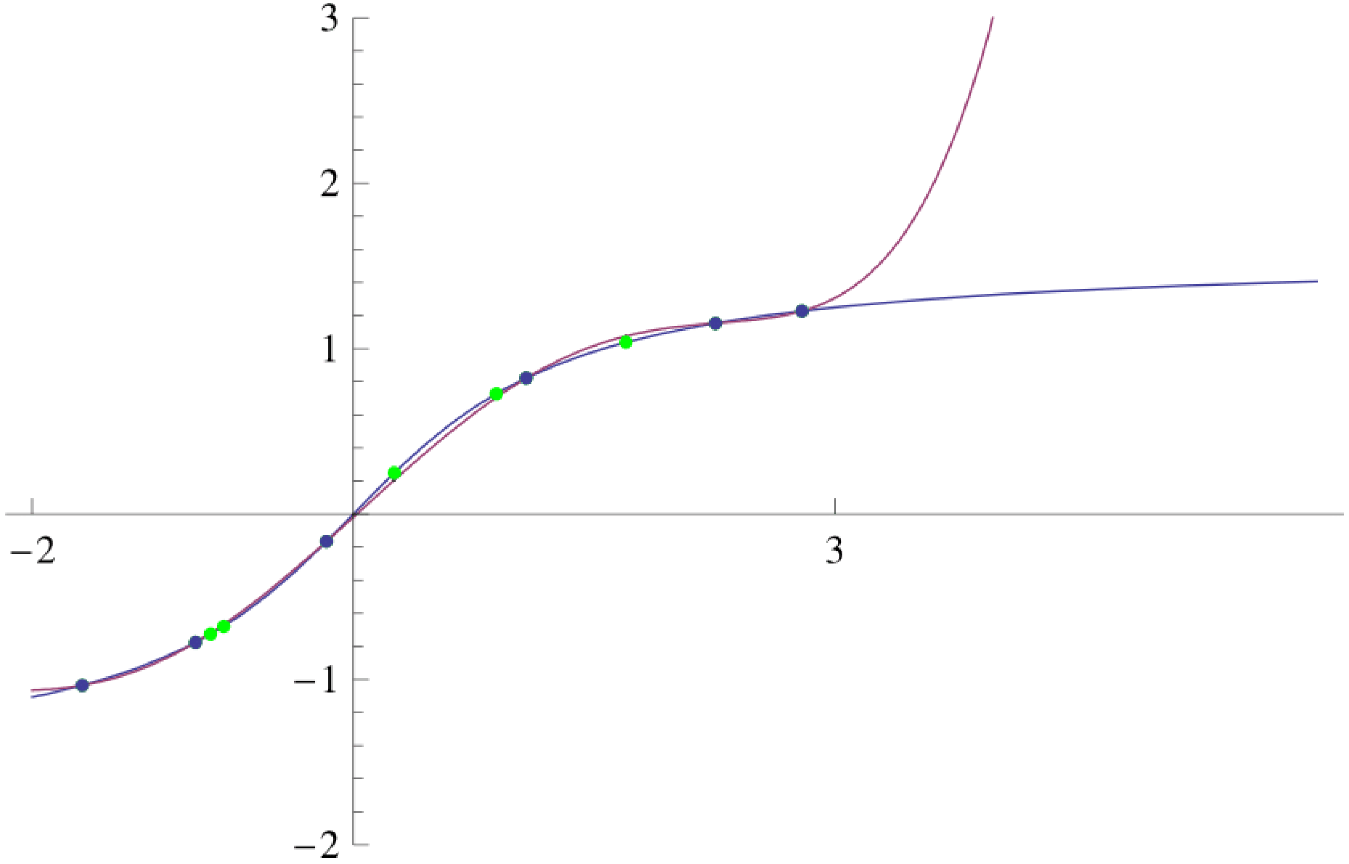
Active learning

Solution: detect when we are extrapolating and switch on learning



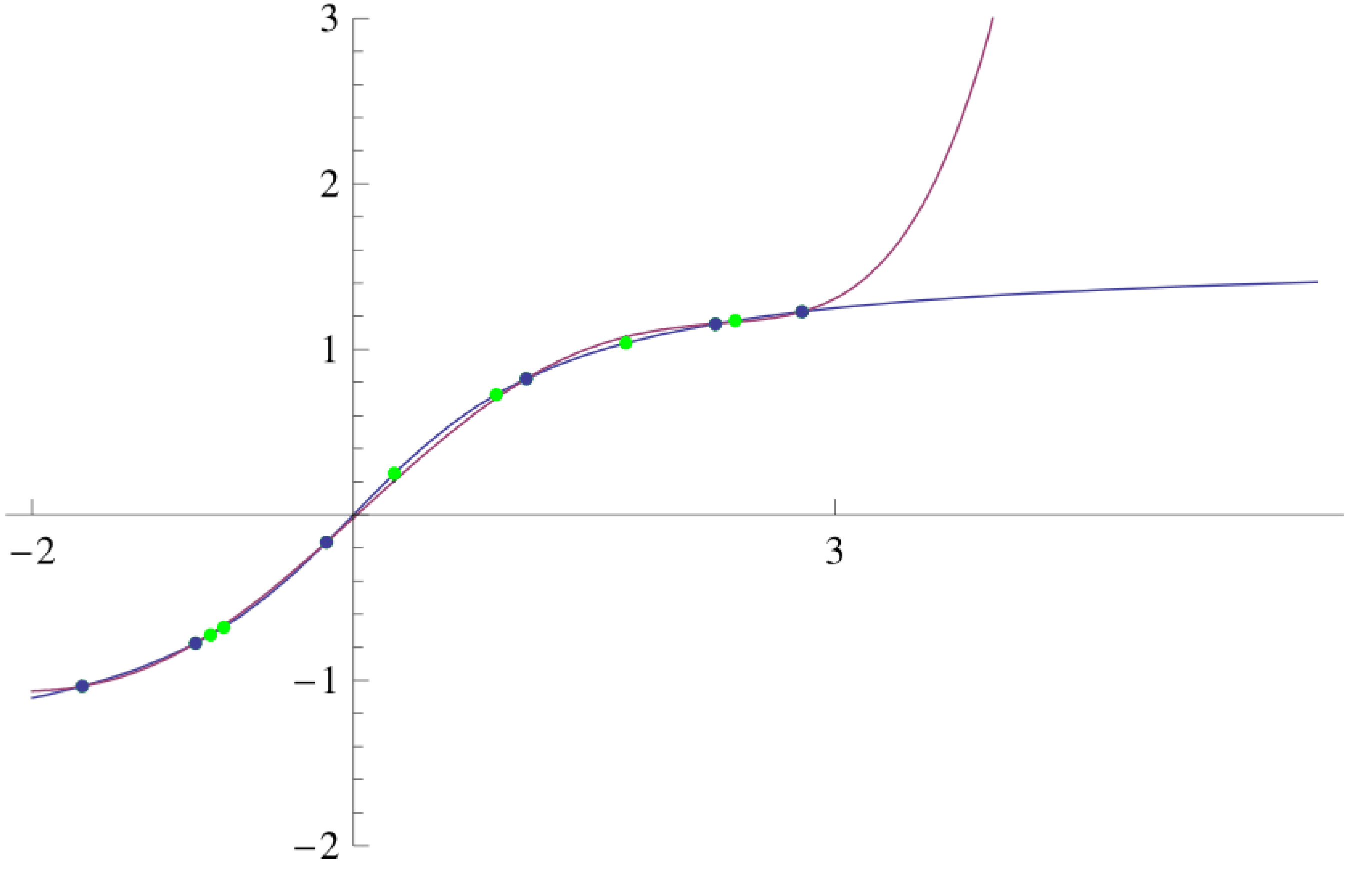
Active learning

Solution: detect when we are extrapolating and switch on learning



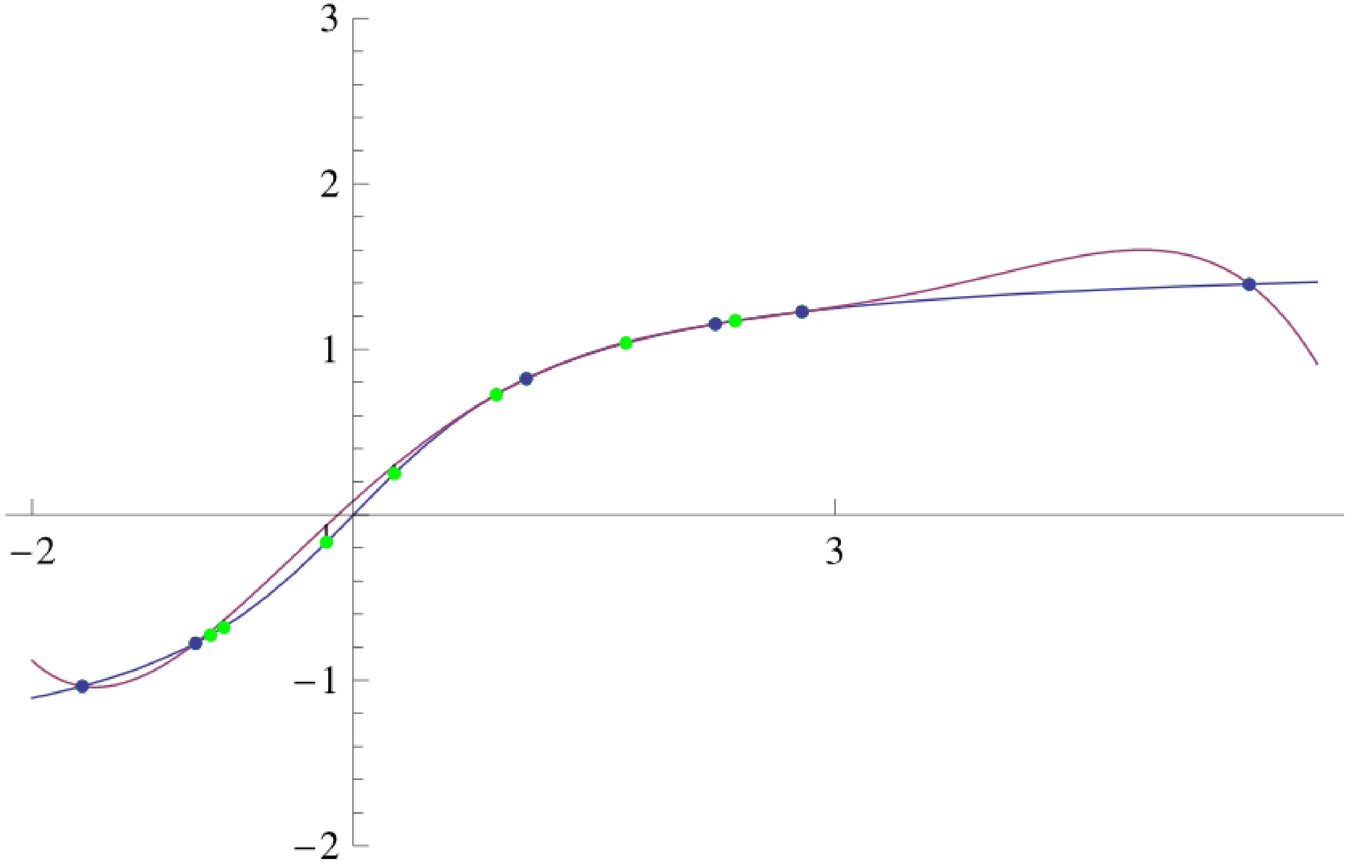
Active learning

Solution: detect when we are extrapolating and switch on learning



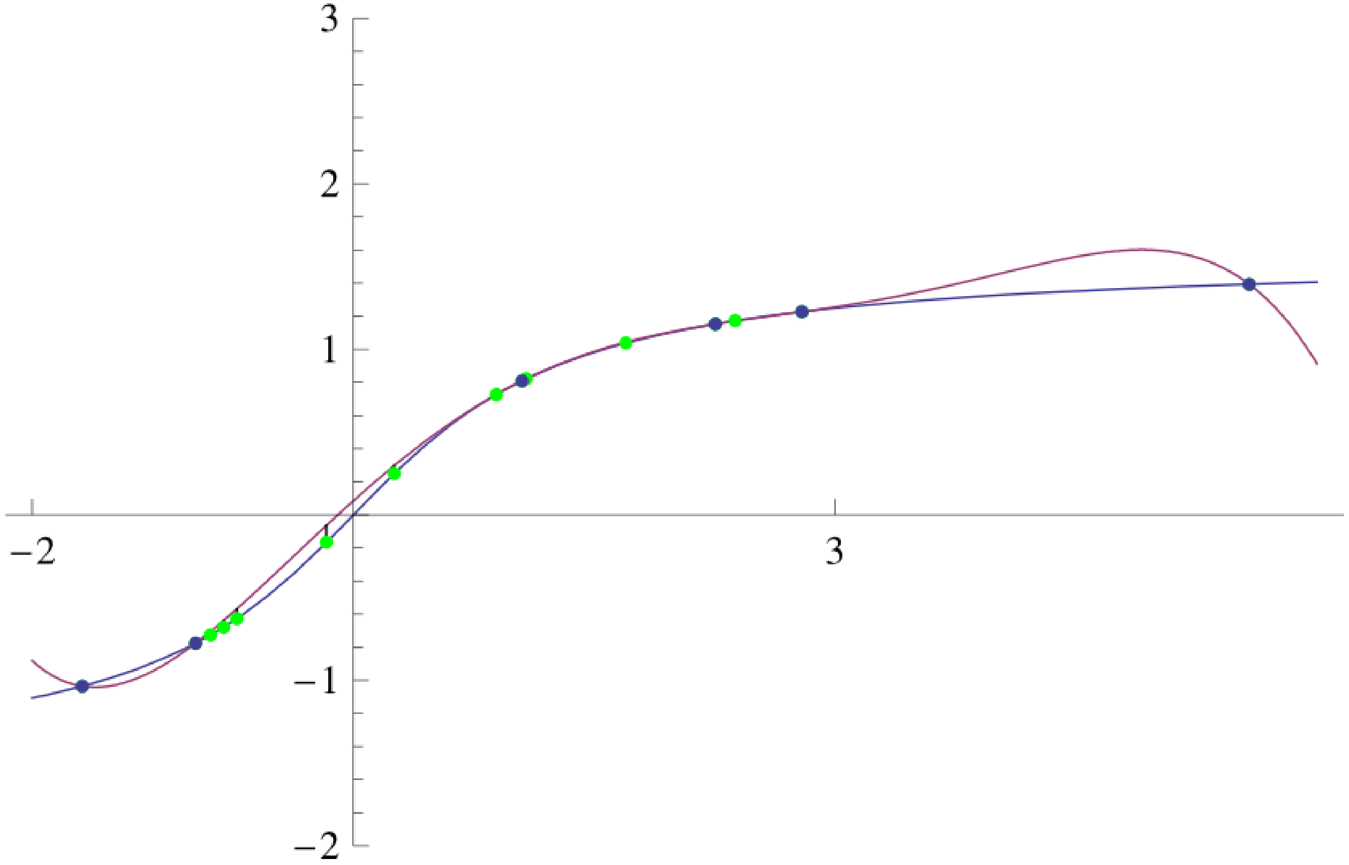
Active learning

Solution: detect when we are extrapolating and switch on learning



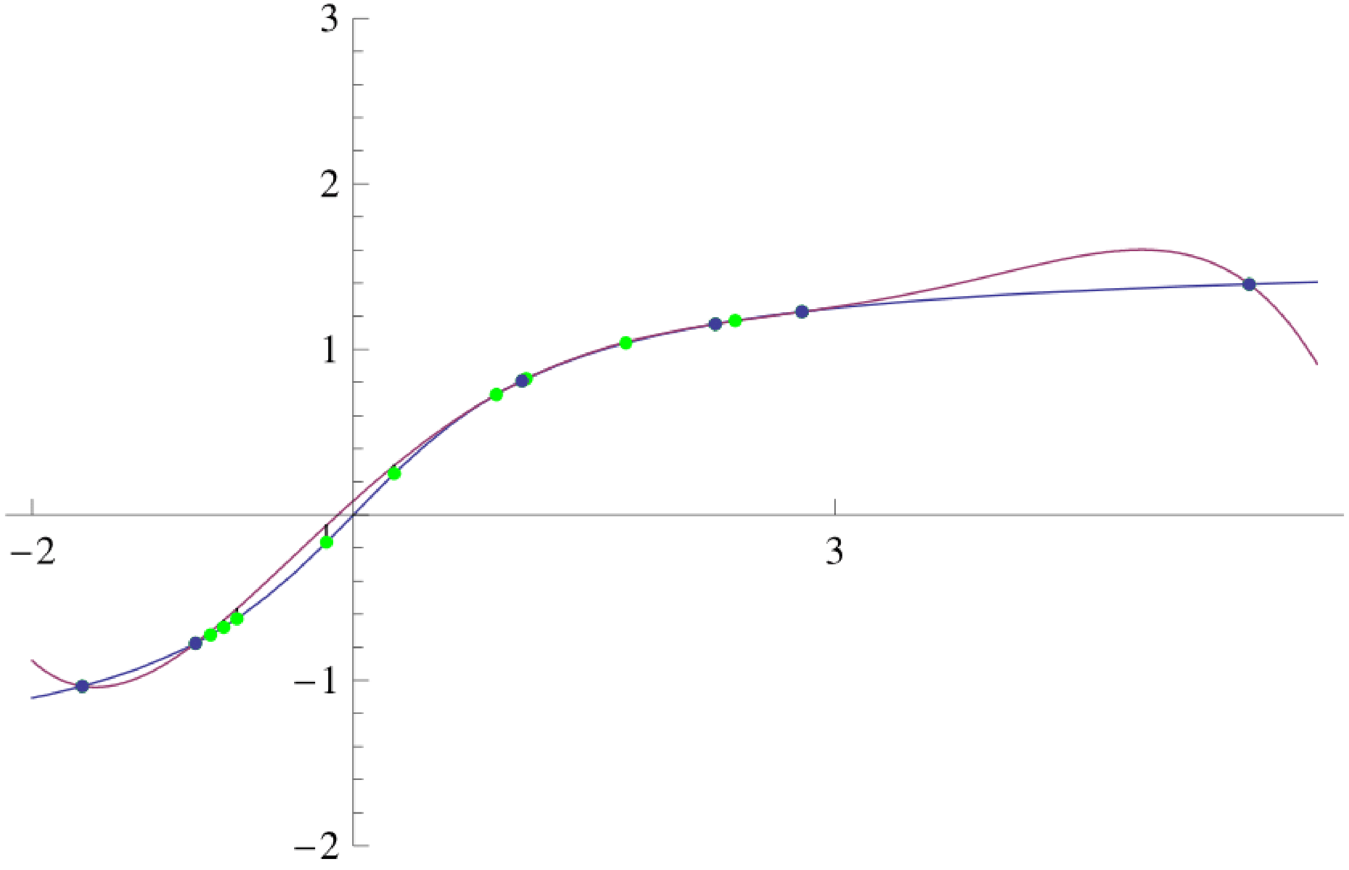
Active learning

Solution: detect when we are extrapolating and switch on learning



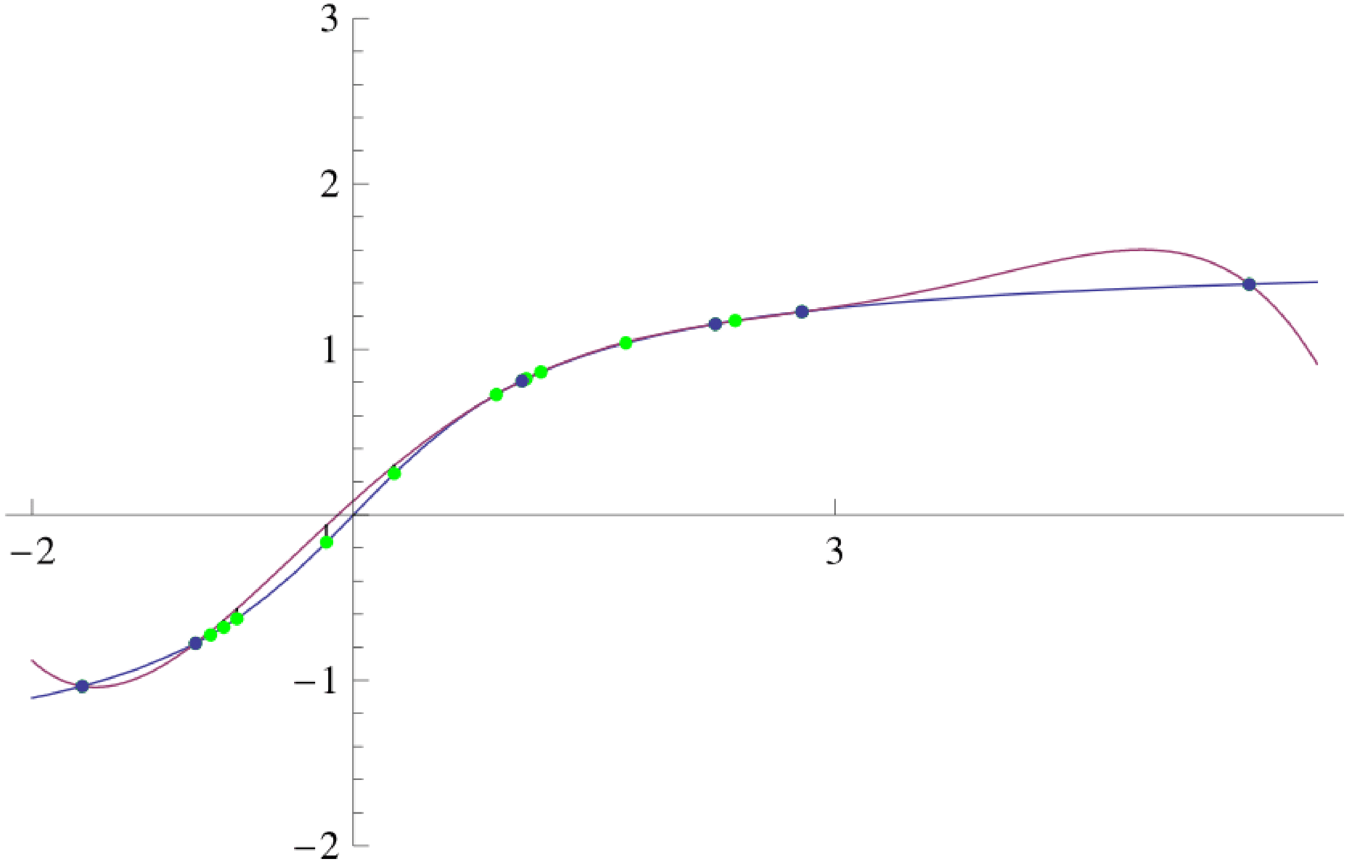
Active learning

Solution: detect when we are extrapolating and switch on learning



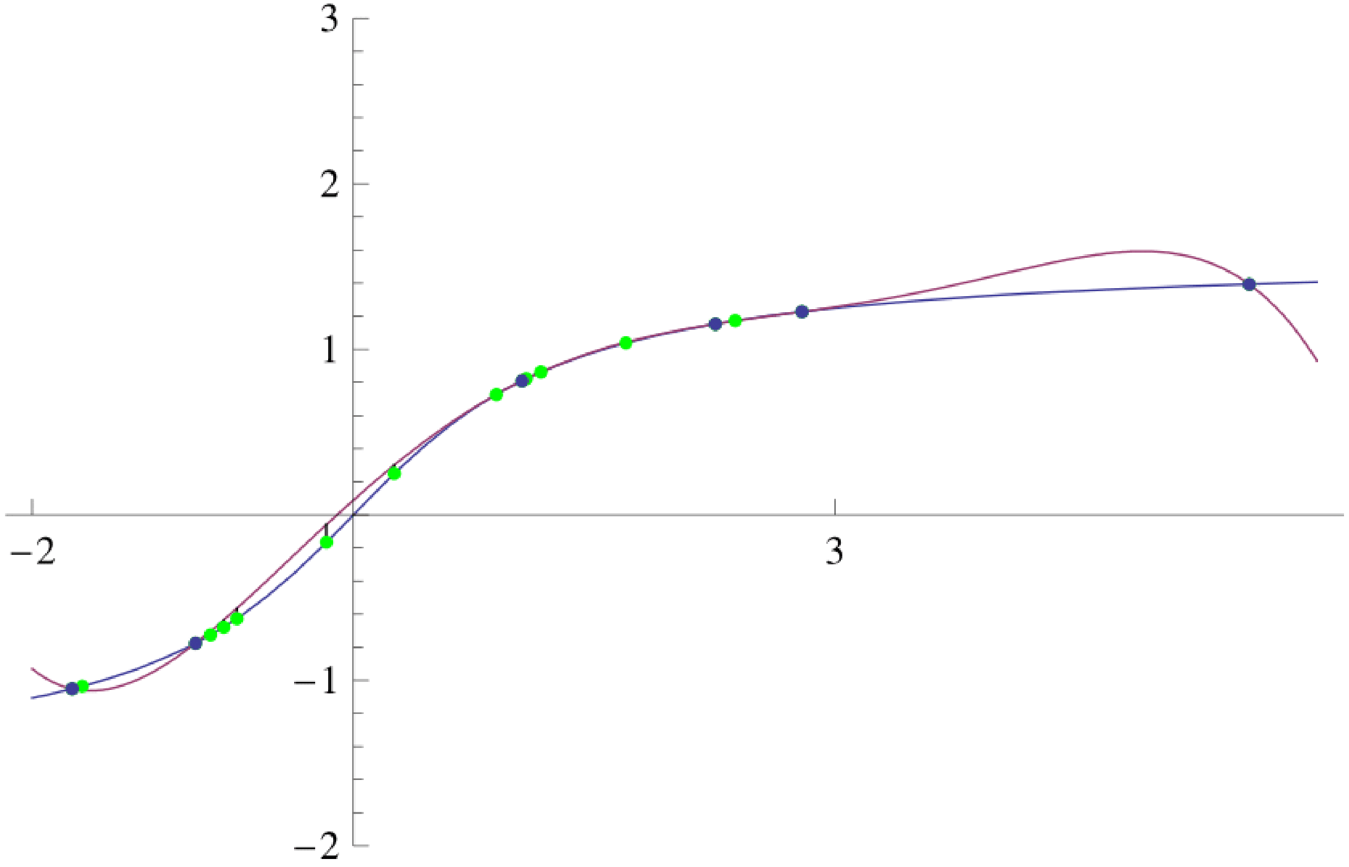
Active learning

Solution: detect when we are extrapolating and switch on learning



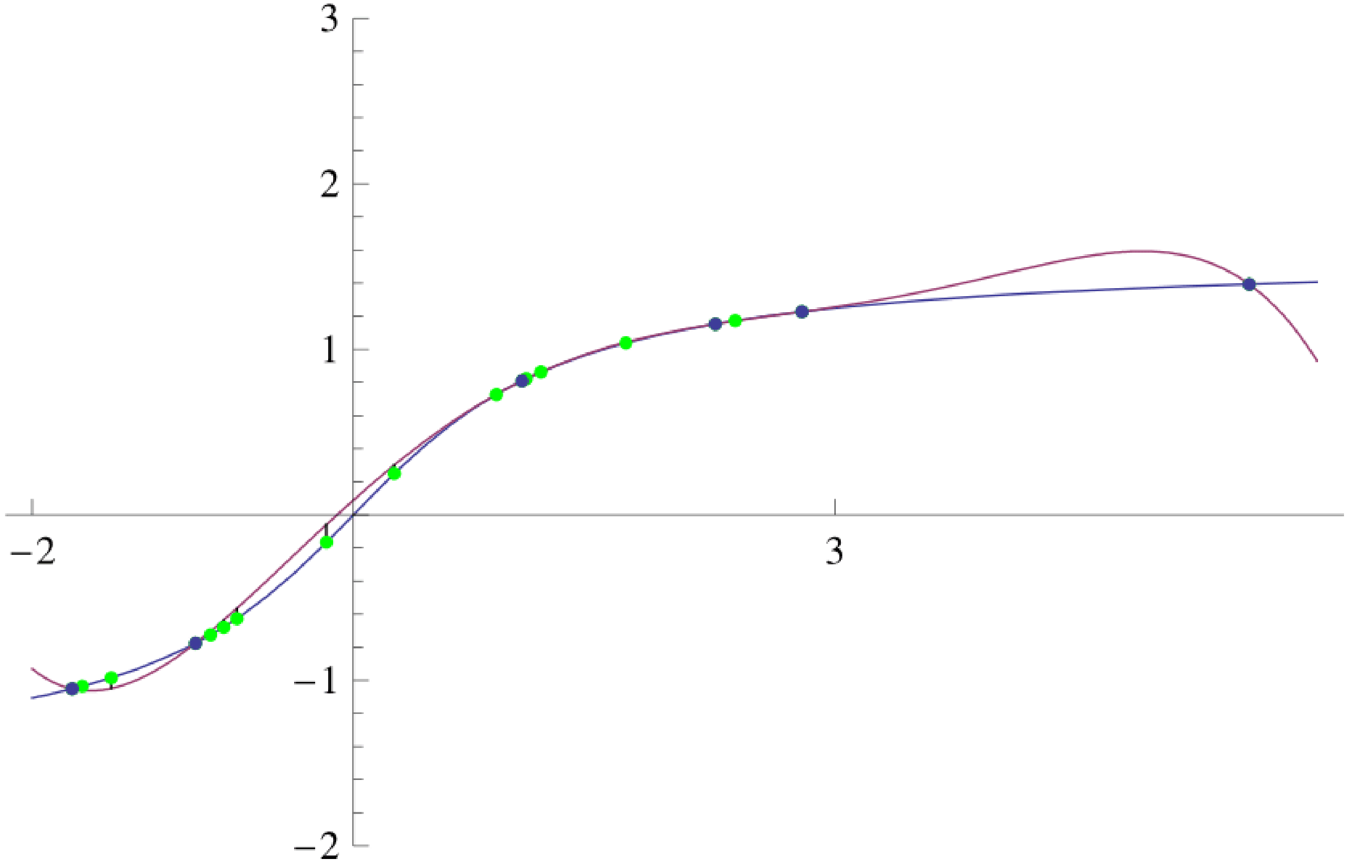
Active learning

Solution: detect when we are extrapolating and switch on learning



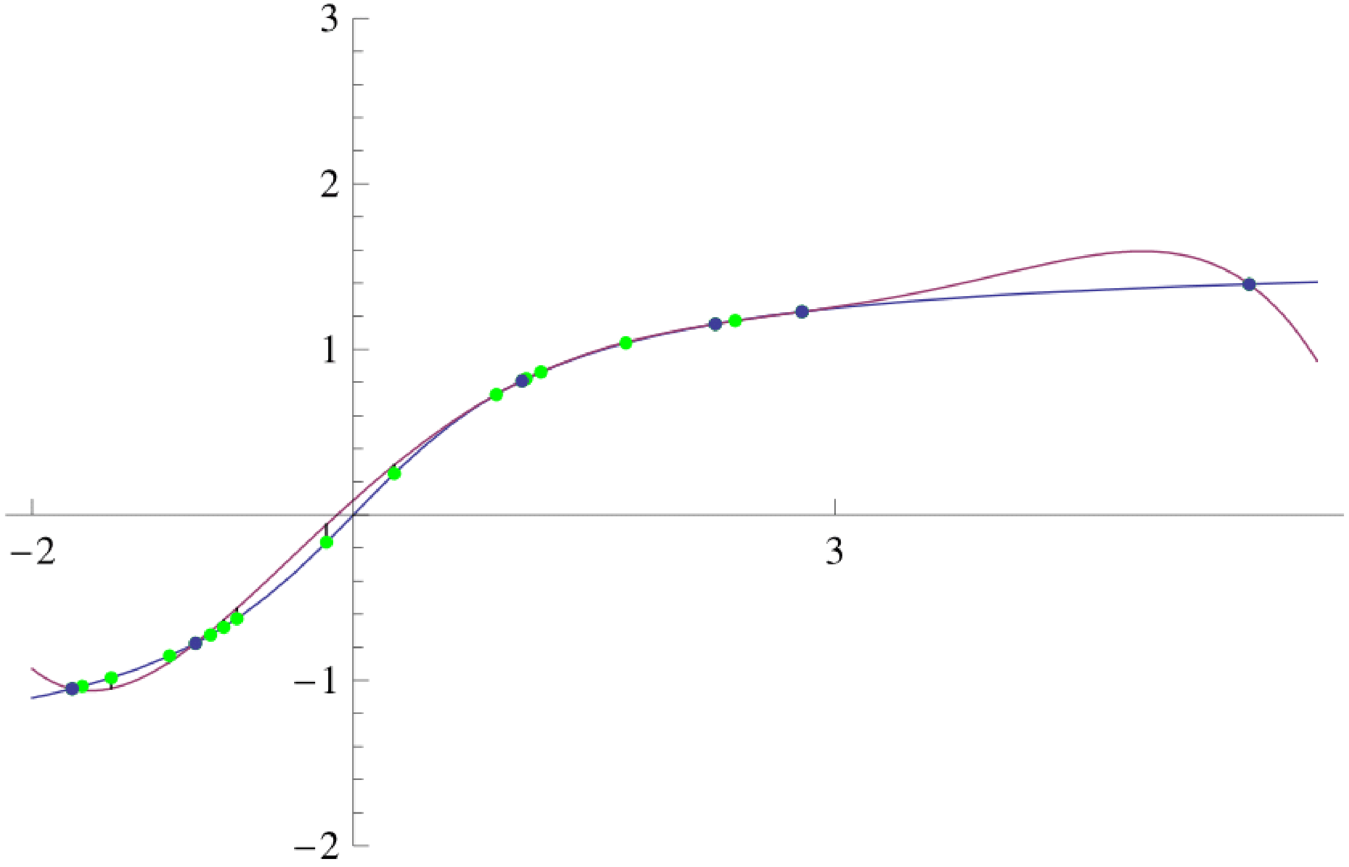
Active learning

Solution: detect when we are extrapolating and switch on learning



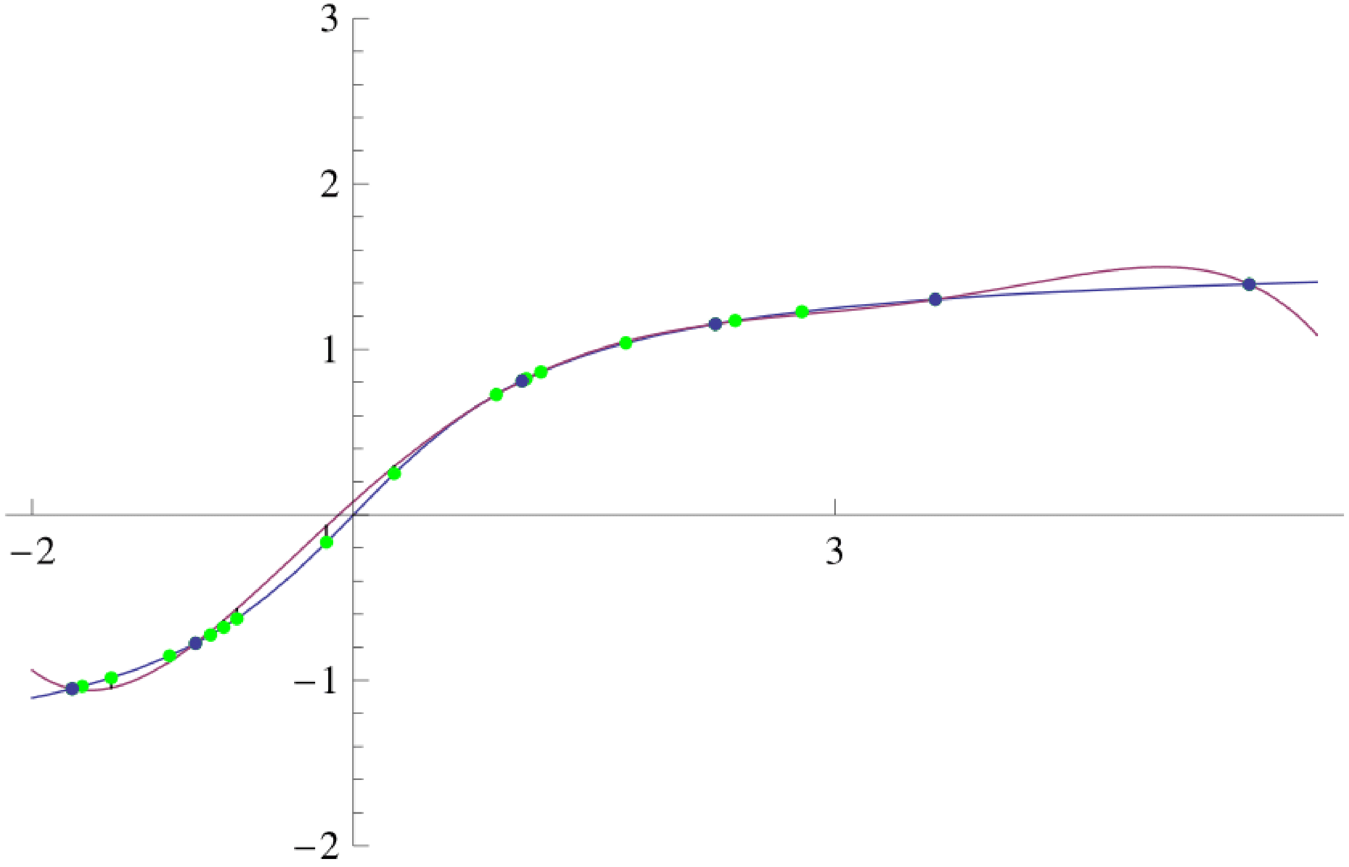
Active learning

Solution: detect when we are extrapolating and switch on learning



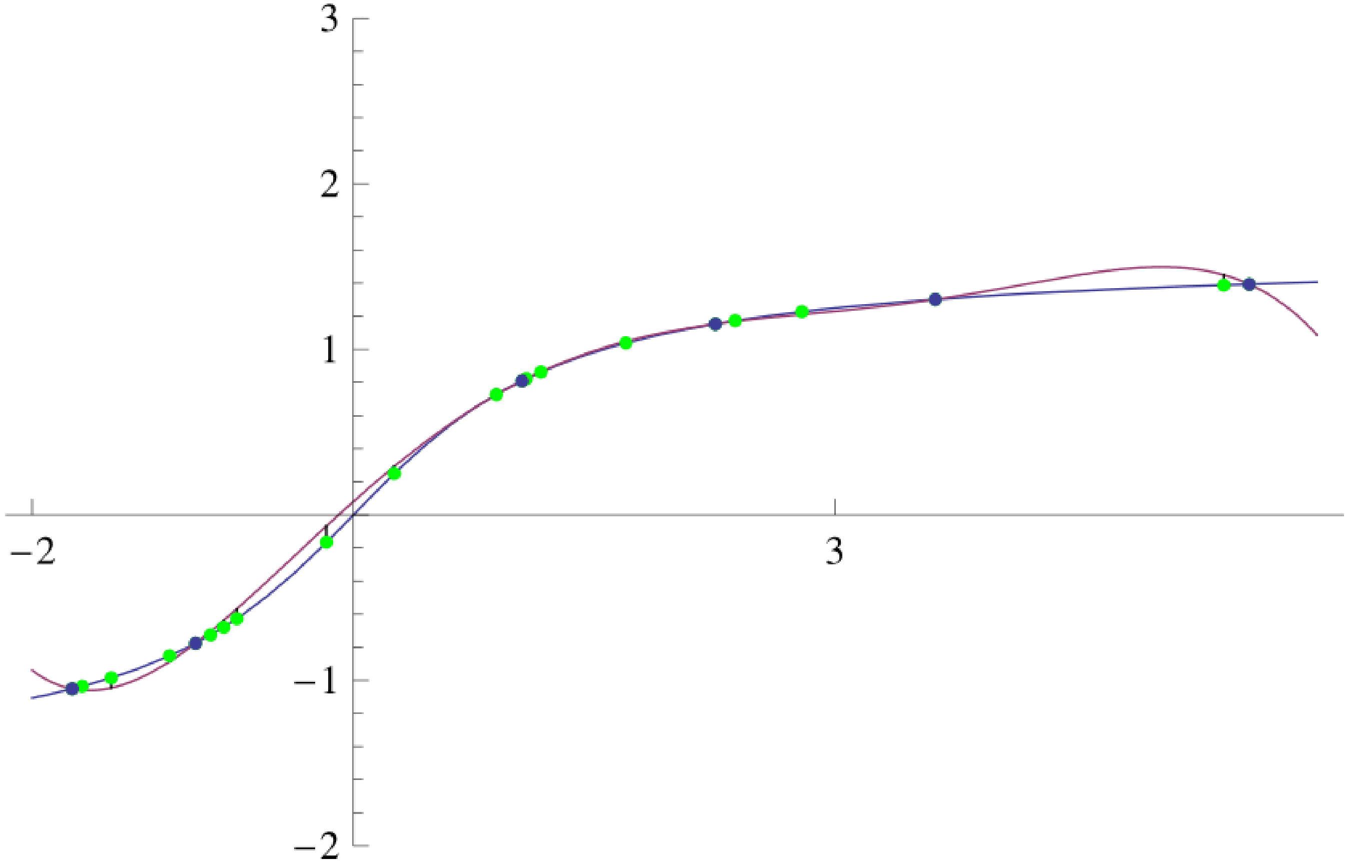
Active learning

Solution: detect when we are extrapolating and switch on learning



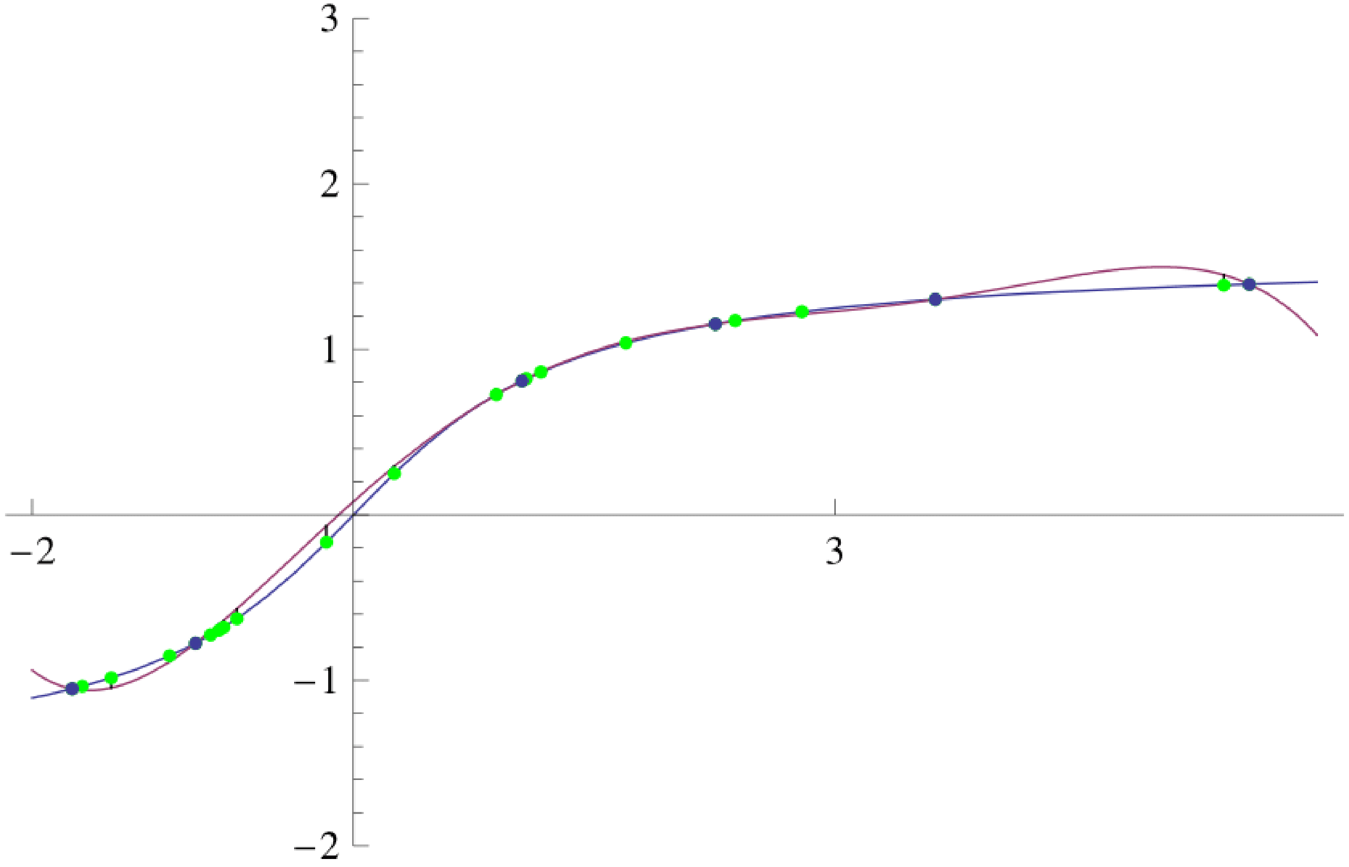
Active learning

Solution: detect when we are extrapolating and switch on learning



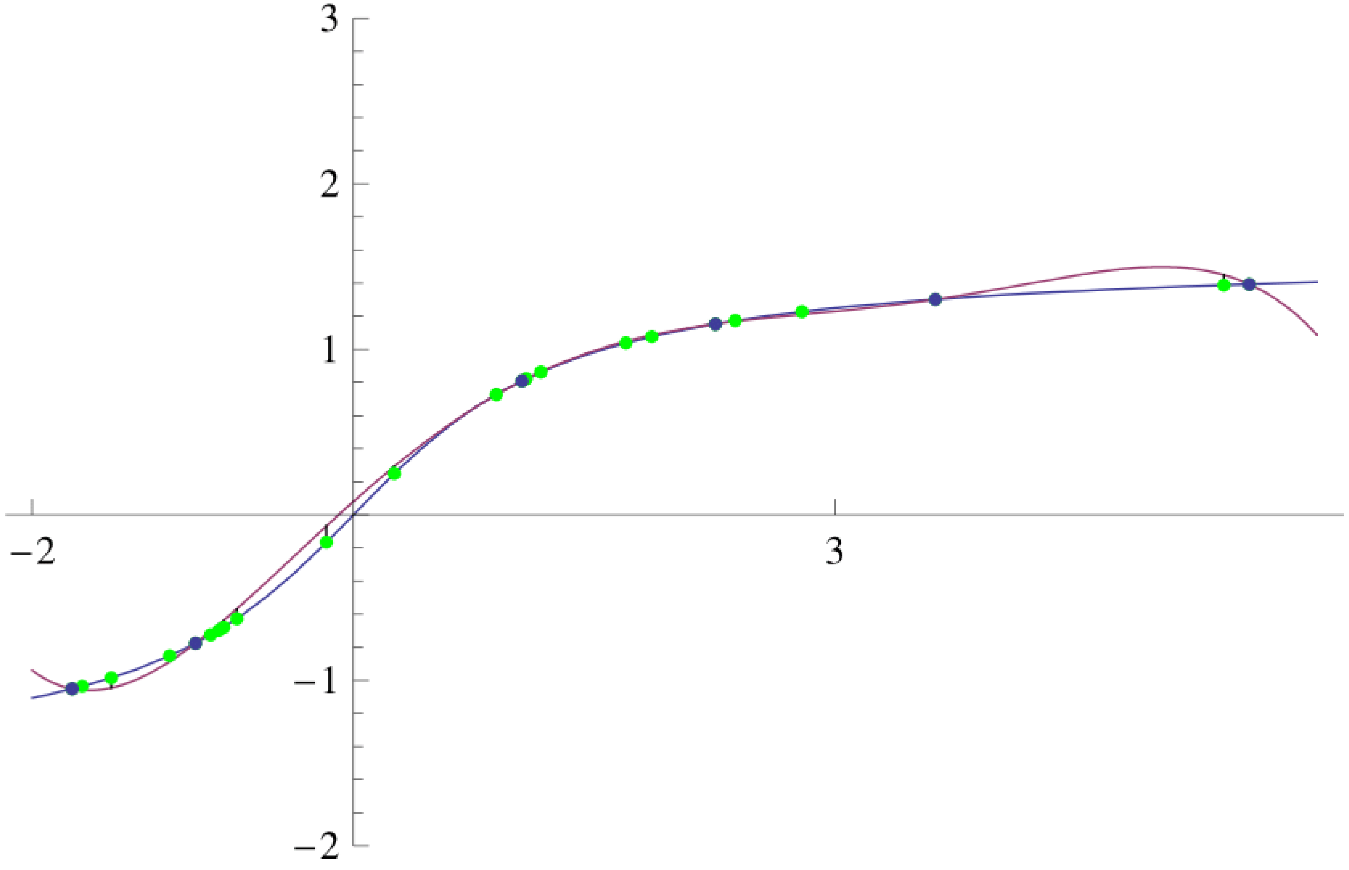
Active learning

Solution: detect when we are extrapolating and switch on learning



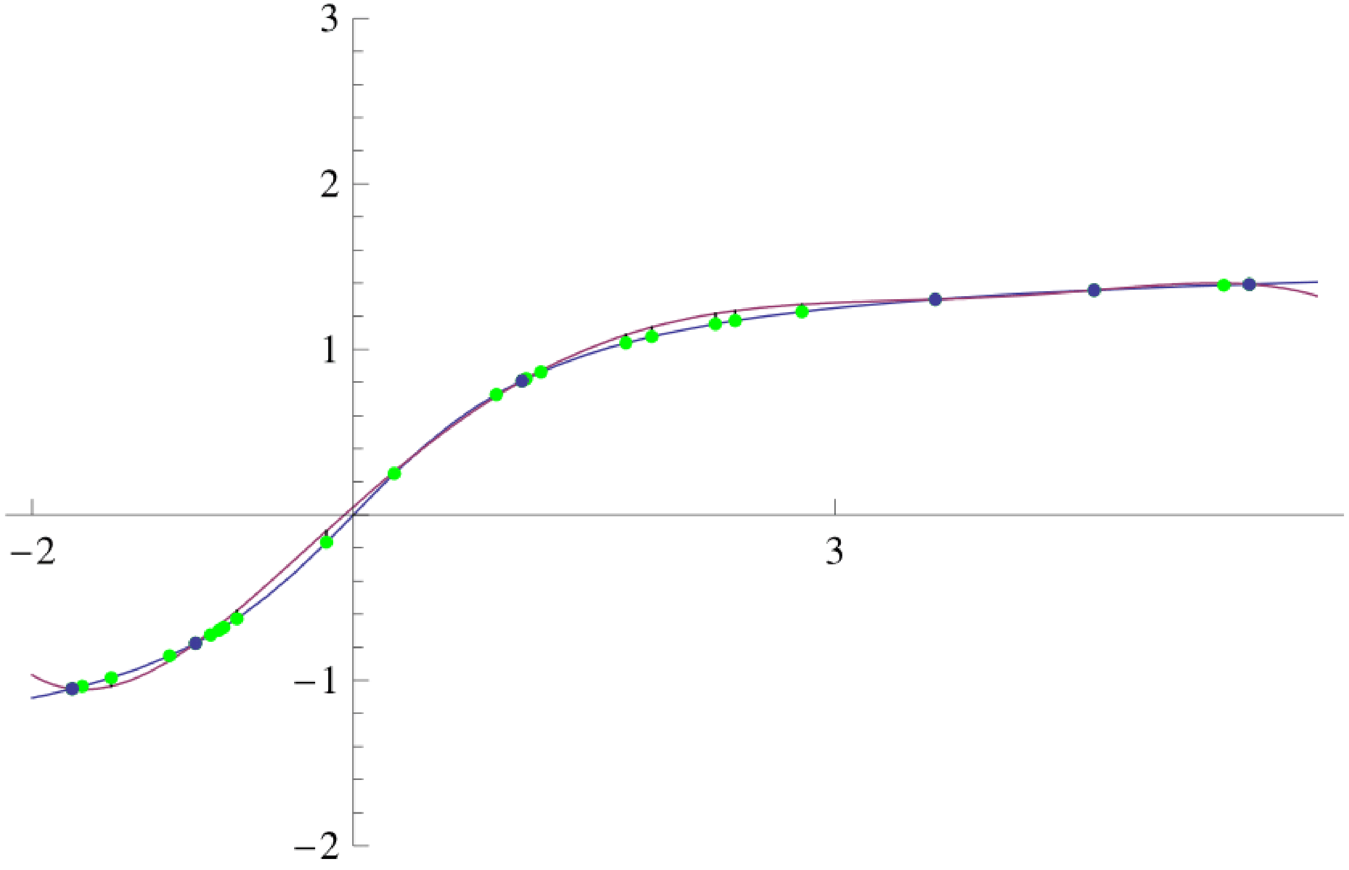
Active learning

Solution: detect when we are extrapolating and switch on learning



Active learning

Solution: detect when we are extrapolating and switch on learning



Existing developments (known to me)

- J. Behler;
  Smith, Nebgen, Lubbers, Isayev, Roitberg;
  Zhang, Lin, Wang, Car, E
  query by committee
- R. Ramprasad:
  train a **2^{nd} ML model** to predict the degree of uncertainty
- Jinnouchi, Lahnsteiner, Karsai, Kresse, Bokdam:
  Gaussian process **predictive variance**
- Noam, Csanyi, Deringer:
  a **metric**-based criterion
- A.S.:
  D-optimality

How we do it?

D-optimality

Skip to Applications

D-optimality

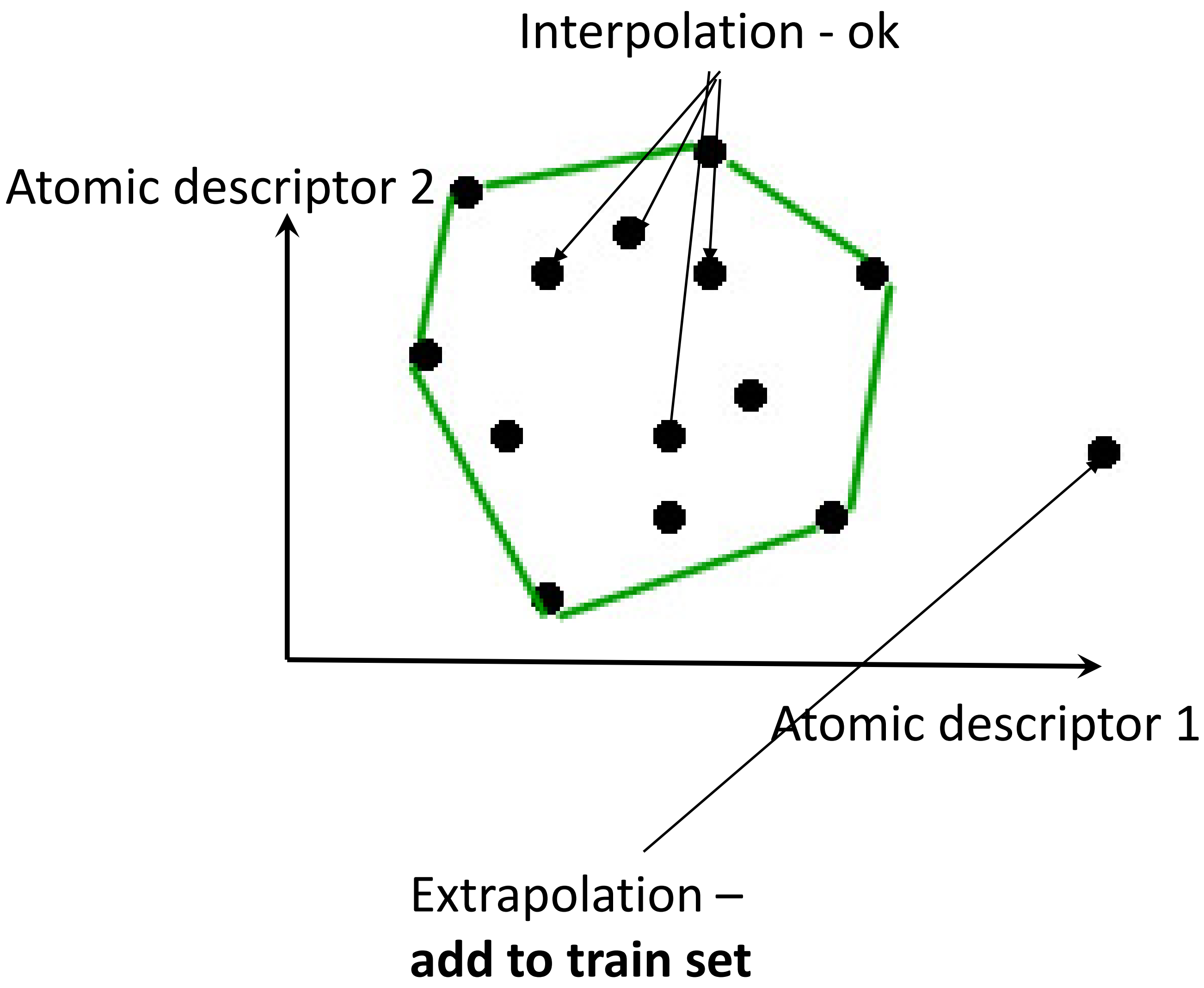
essentially

- detects hitting outside a convex hull,

but for linear models

(convex hull -> simplex)

Algorithm: $O(N^2)$

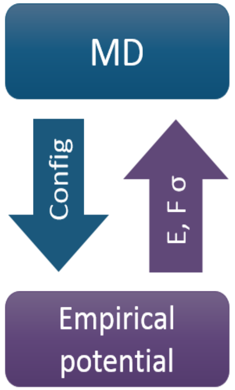


E. Podryabinkin, A. Shapeev (2017)

Applications

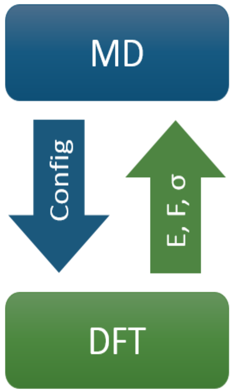
Application #1: Learning on the fly

MD with empirical potential



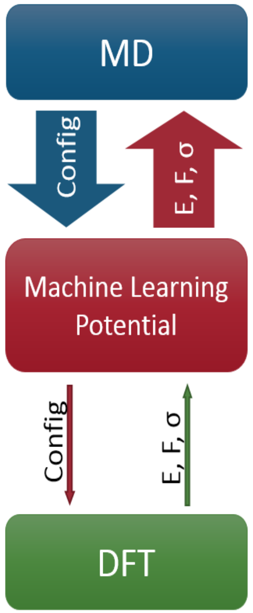
+ Fast
– Qualitative accuracy only

Ab-initio MD



– Time consuming
+ Accurate

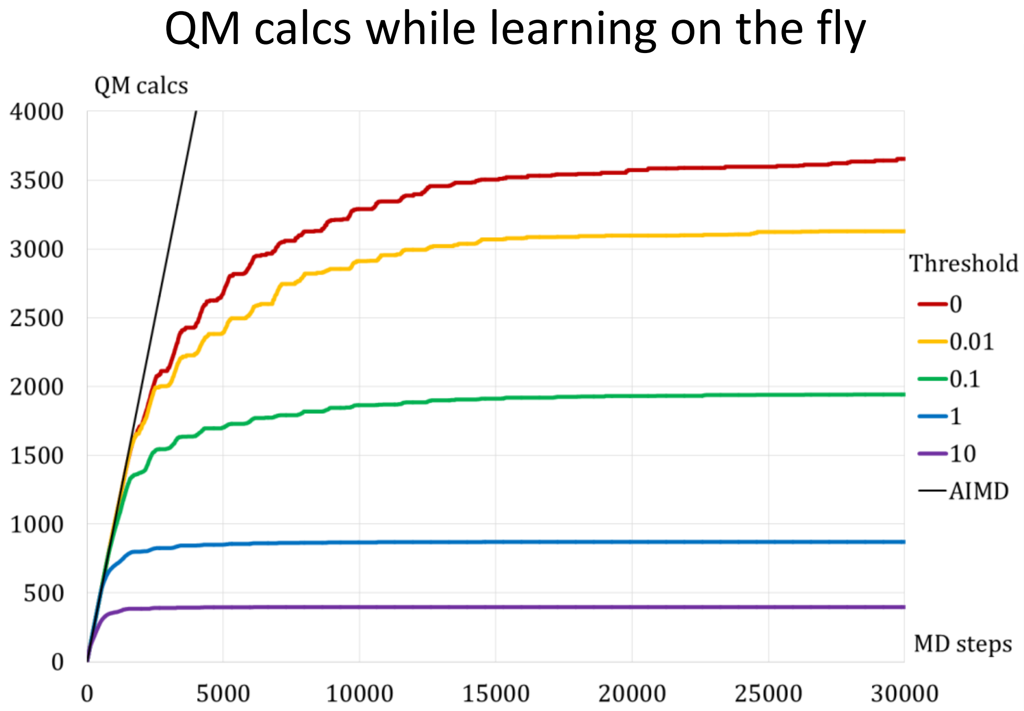
Hybrid MD



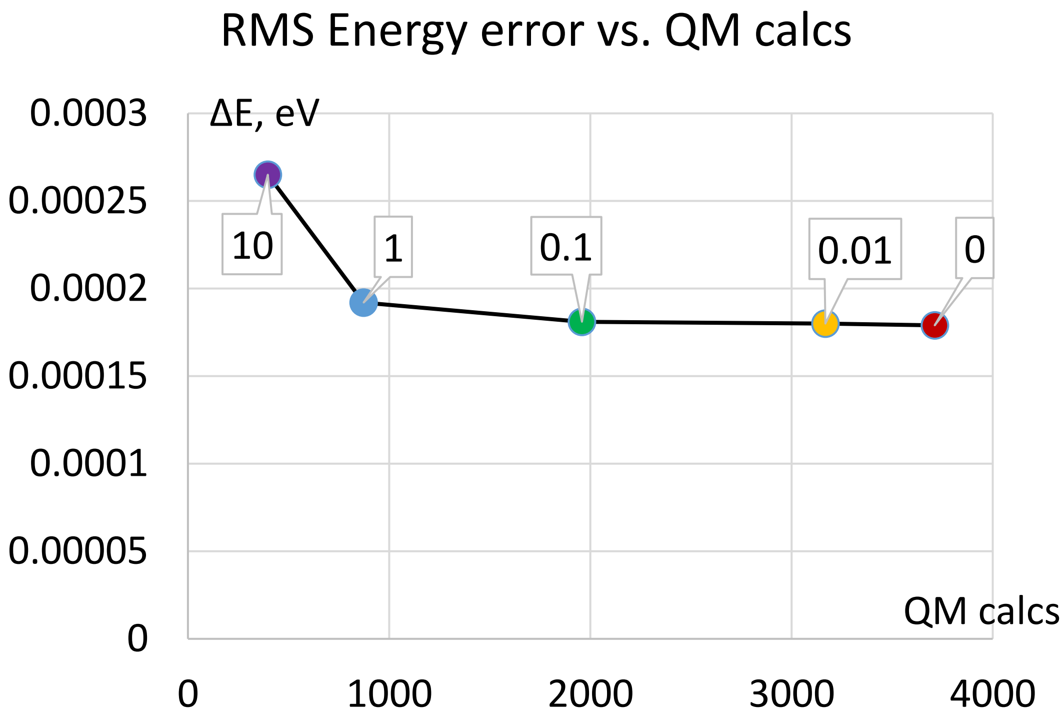
+ Fast
+ Accurate (hopefully)

- 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

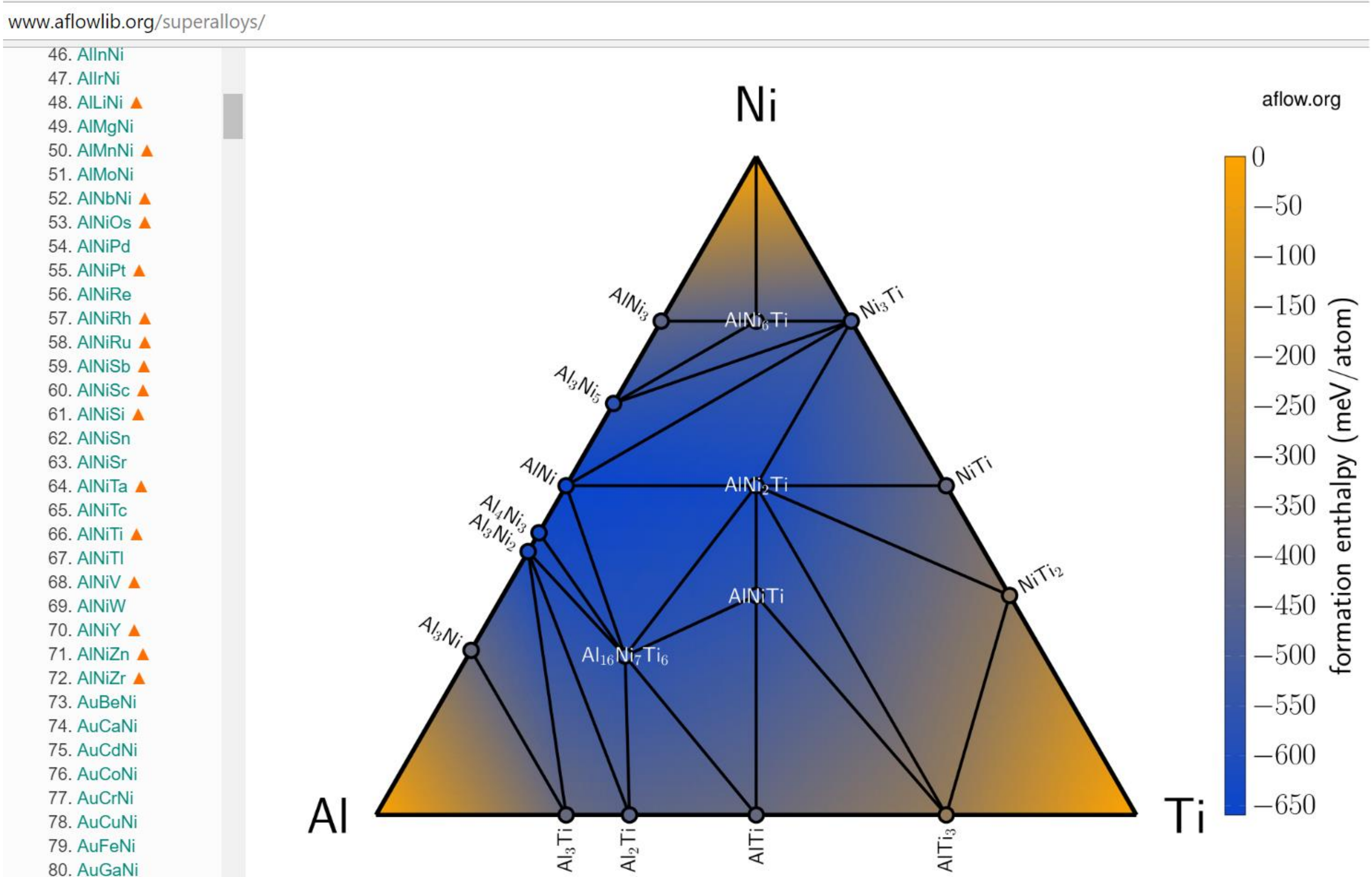


QM calcs while learning on the fly



RMS Energy error vs. QM calcs

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



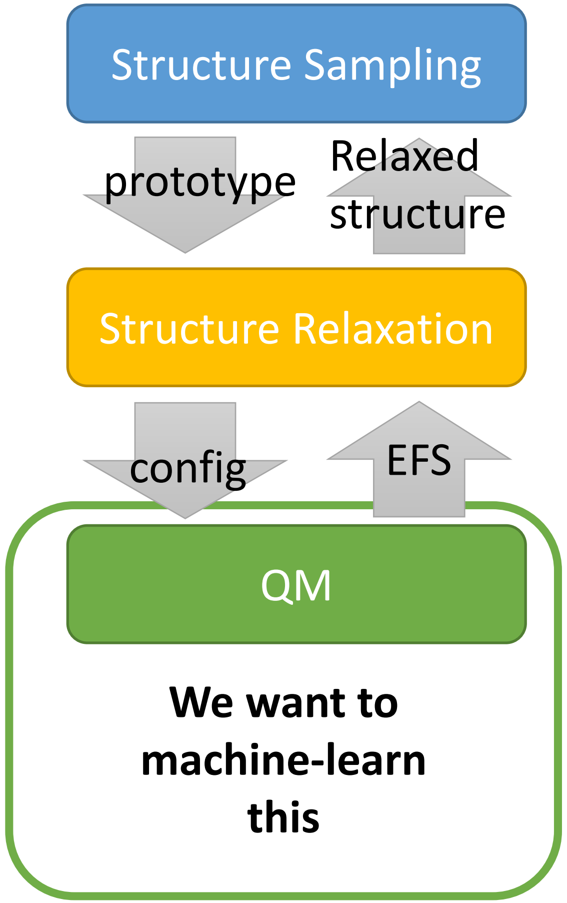
www.aflowlib.org/superalloys/

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

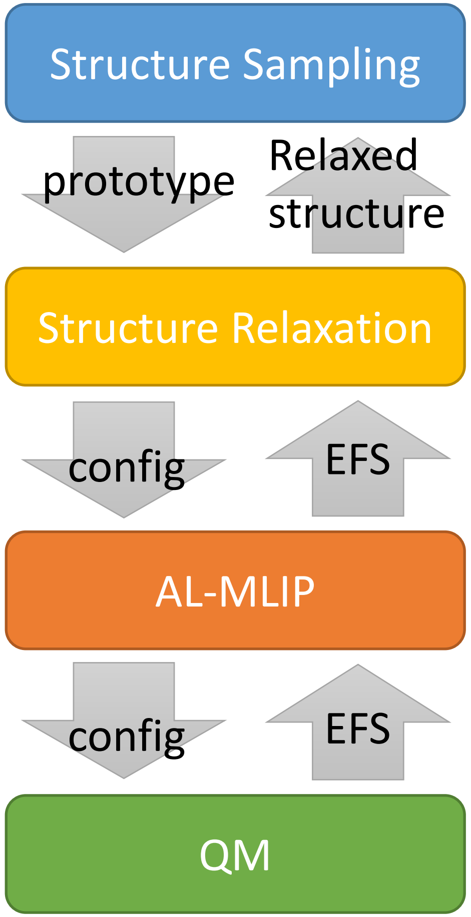


Convex hulls now

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

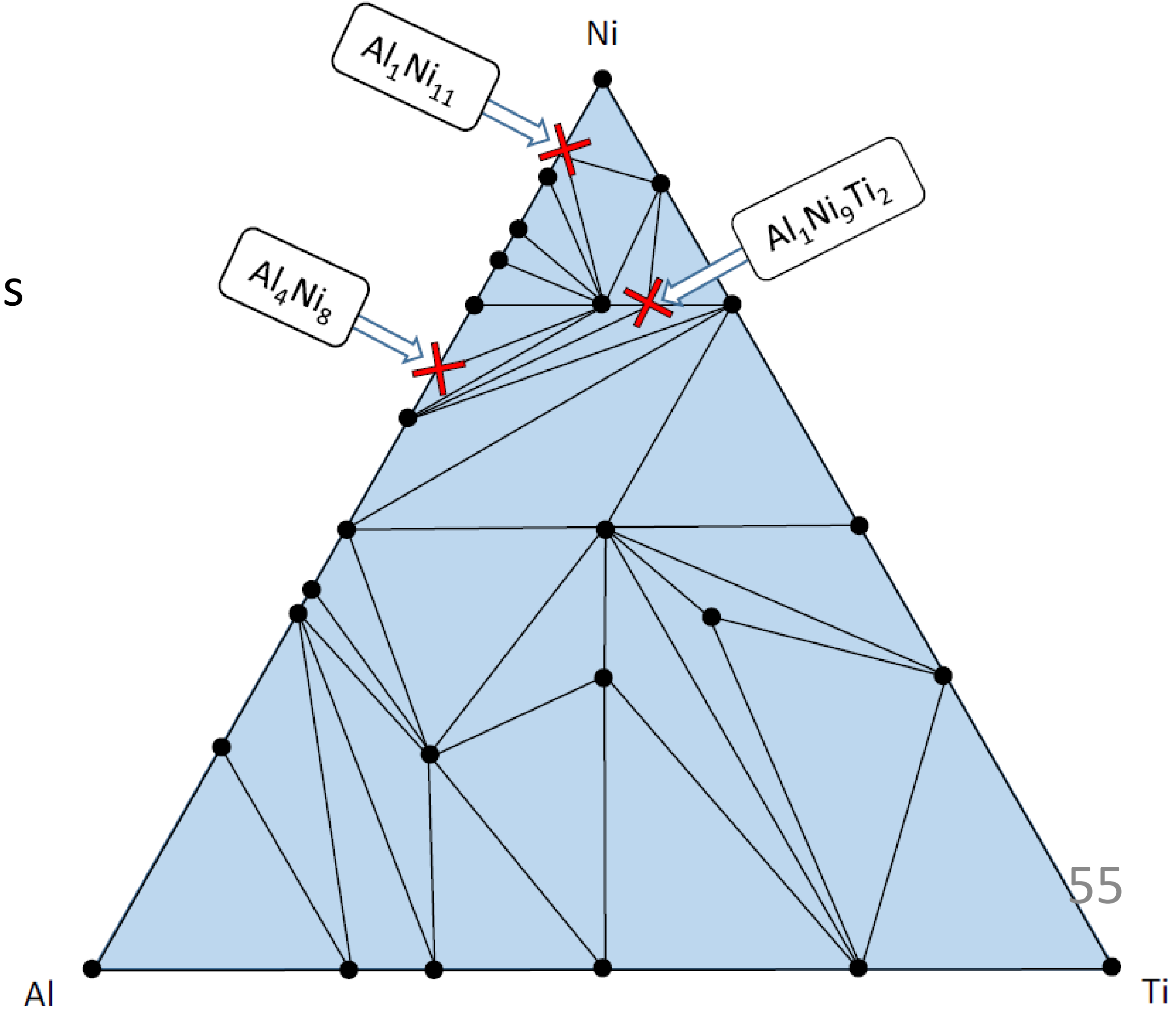
How it is done:

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



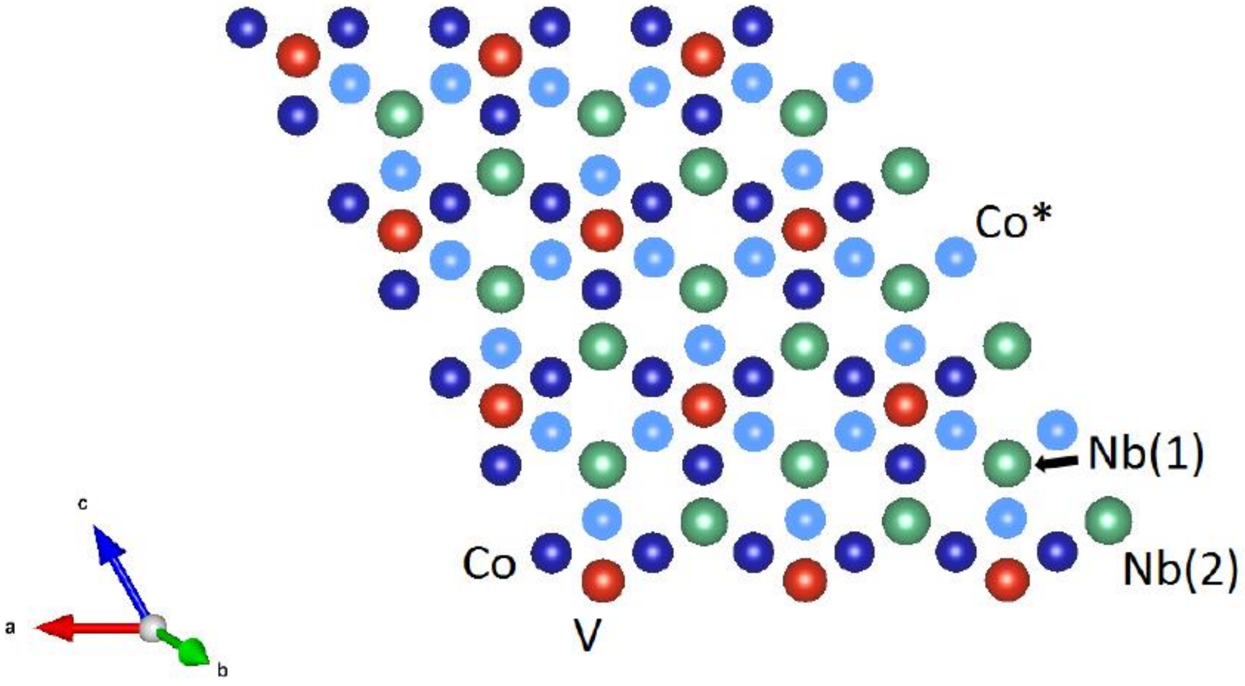
Convex hulls now: details

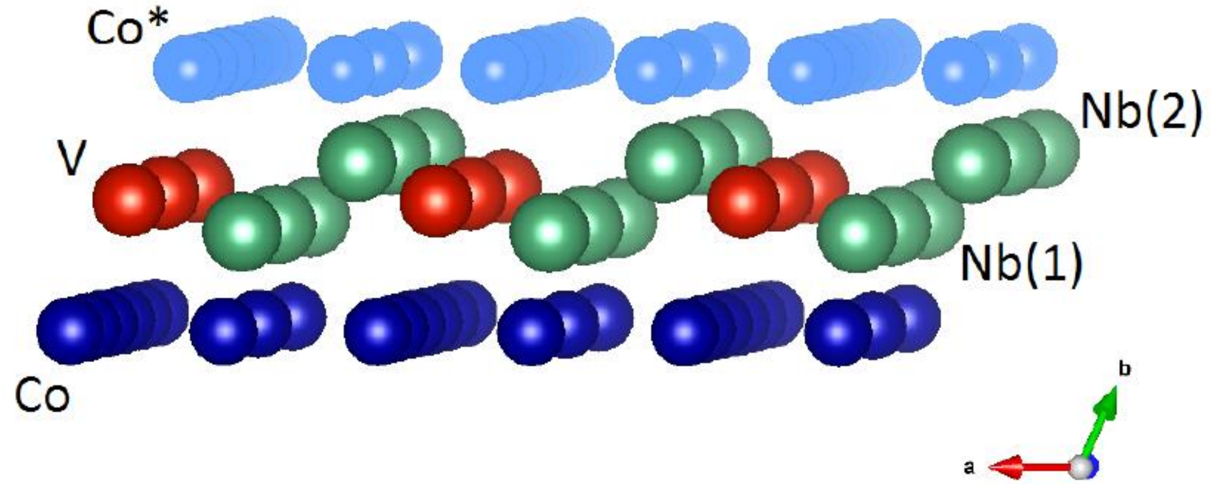
1. Screen-1:
   1. Start with **400K** structures
   2. Obtain **400K** relaxed structures, with RMSE = **25** meV/atom
   3. Retain **40K** low-energy structures (within 4-σ)
2. Screen-2:
   1. Start with **60K** structures
   2. Obtain **60K** relaxed structures, with RMSE = **9** meV/atom
   3. Retain **7K** low-energy structures (within 4-σ)
3. Final relaxation:
   1. Relax **7K** structures on DFT



Results

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





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 (9 meV/atom):
- Sampling is now the bottleneck, not DFT