Gaussian Processes for Force Fields

Aldo Glielmo, Claudio Zeni

Physics Department, King's College London

08 May 2019 – Aalto University (FI)

KING'S College LONDON

EPSRC
Engineering and Physical Sciences Research Council

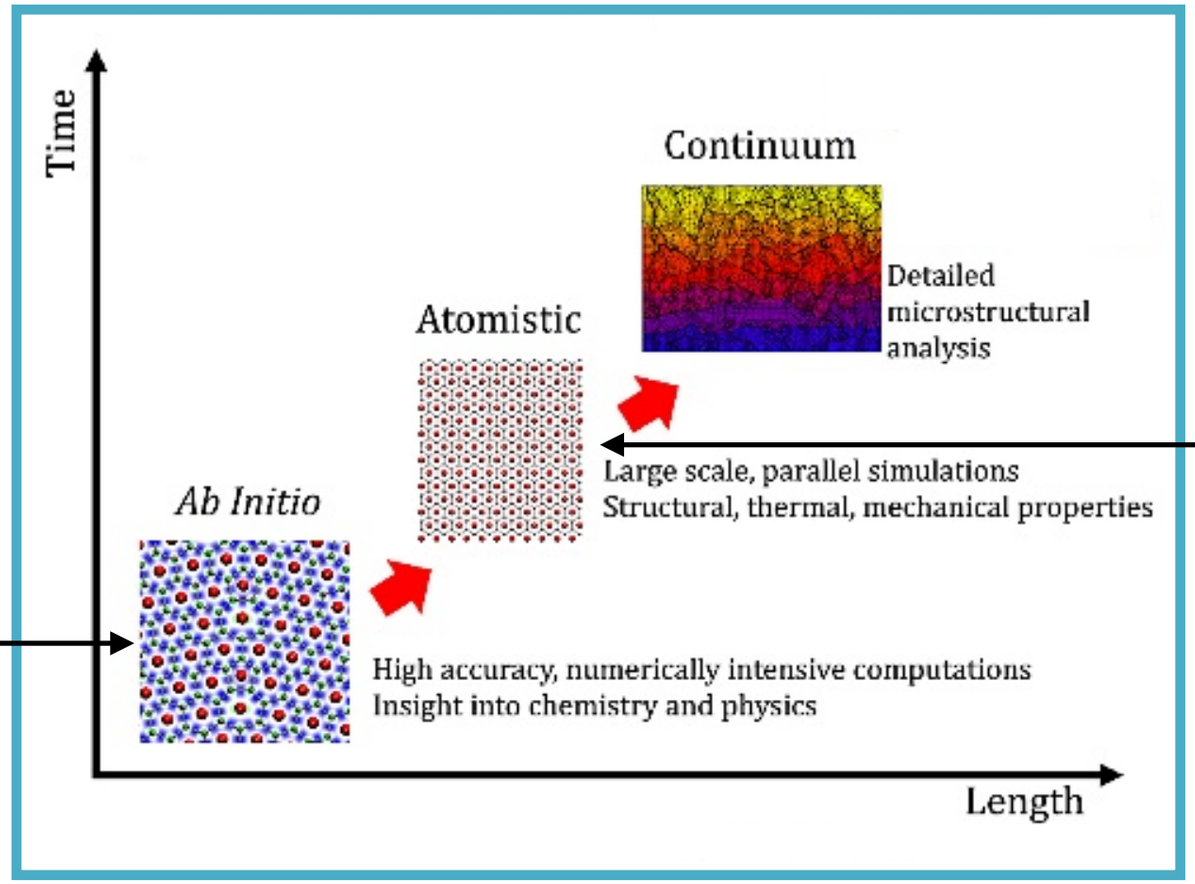
Part I

Bayesian inference and Gaussian process regression

Aldo Glielmo

Physics Department, King's College London

The aim of classical potentials



Quantum Hamiltonian

$\hat{H} \mid \phi_0\rangle = \varepsilon_0 \mid \phi_0\rangle$

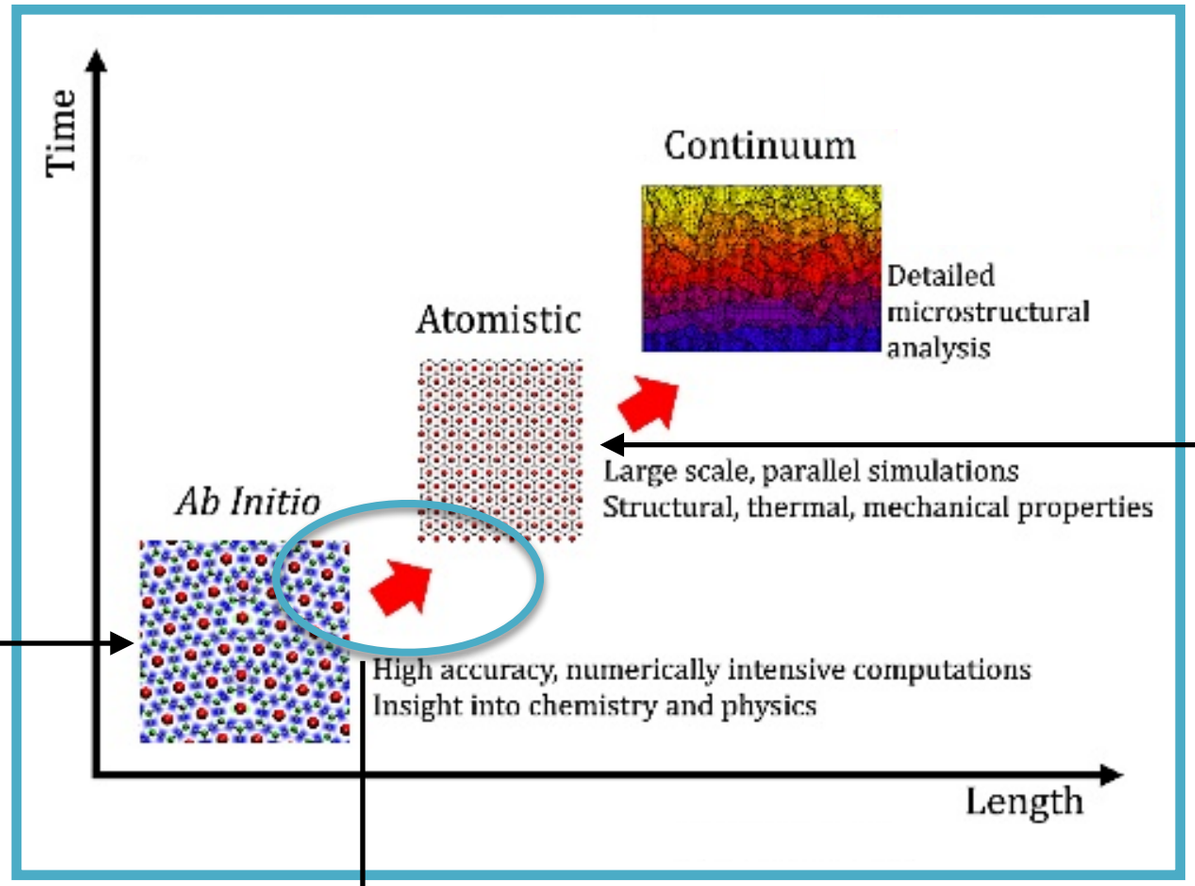
Classical Potential

$E = E(\{\mathbf{r}_i\})$

References:

- ***Skinnert et al. Mod. & Sim. in Materials (1994)***
- Behler, J., & Parrinello, M. PRL (2007)
- Bartók et al. PRL (2010)
- Li, Z., Kermode, J. R., & De Vita, A. *PRL* (2015)

The aim of classical potentials



Quantum Hamiltonian

$\hat{H} \mid \phi_0\rangle = \varepsilon_0 \mid \phi_0\rangle$

Classical Potential

$E = E(\{\mathbf{r}_i\})$

ML Potentials:

Systematic approaches to implicitly model the electronic degrees of freedom

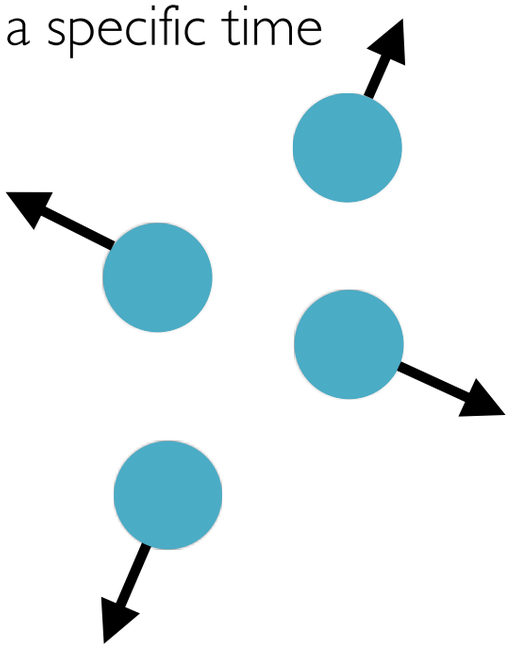
$E = E(\{\mathbf{r}_i\} \mid \varepsilon_0)$

References:

- ***Skinnert et al. Mod. & Sim. in Materials (1994)***
- Behler, J., & Parrinello, M. PRL (2007)
- Bartók et al. PRL (2010)
- Li, Z., Kermode, J. R., & De Vita, A. *PRL* (2015)

Learning force fields: problem setup

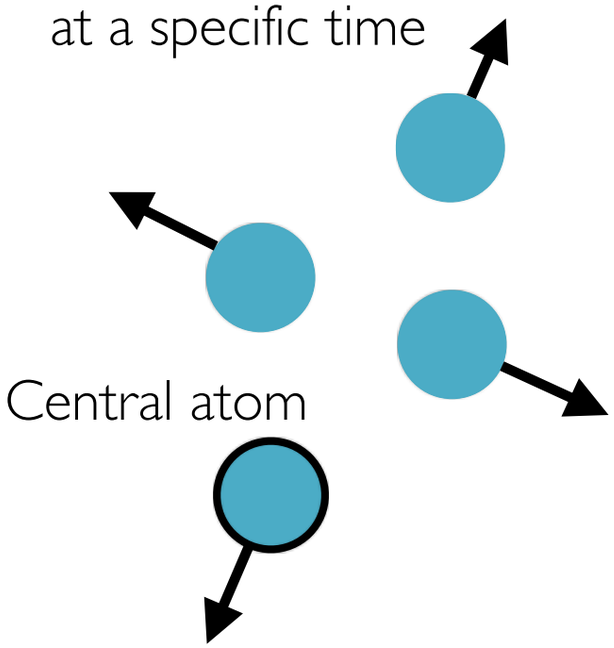
- Quantum system of *N* atoms at a specific time



- The computational cost of calculating energies and forces scales badly with N and also has high prefactors

Learning force fields: problem setup

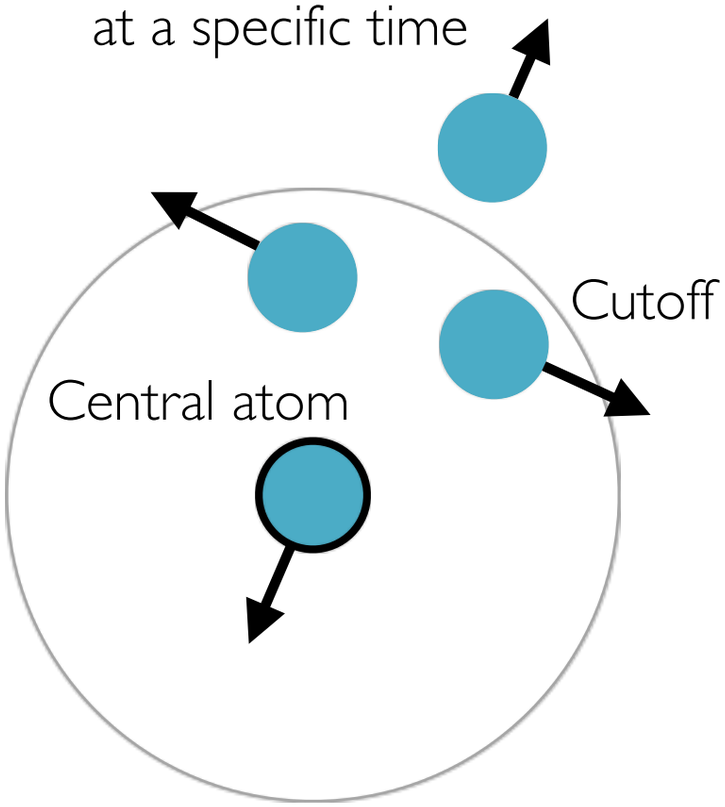
- Quantum system of *N* atoms at a specific time



- The computational cost of calculating energies and forces scales badly with N and also has high prefactors

Learning force fields: problem setup

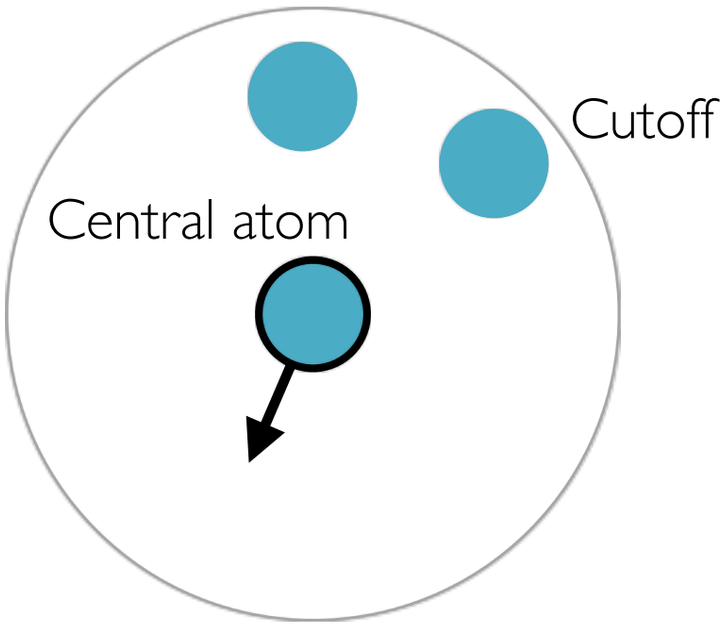
- Quantum system of *N* atoms at a specific time



- The computational cost of calculating energies and forces scales badly with N and also has high prefactors

Learning force fields: problem setup

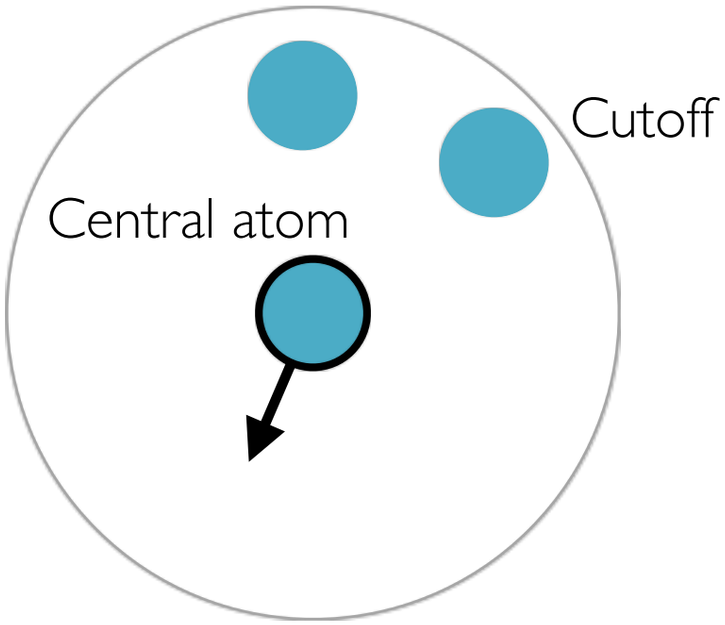
- Quantum system of *N* atoms at a specific time



- The computational cost of calculating energies and forces scales badly with N and also has high prefactors

Learning force fields: problem setup

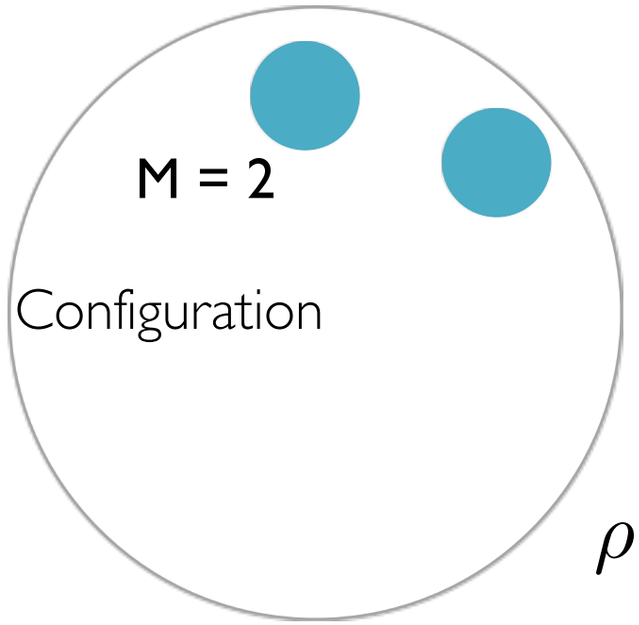
- Quantum system of *N* atoms at a specific time



- The computational cost of calculating energies and forces scales badly with N and also has high prefactors

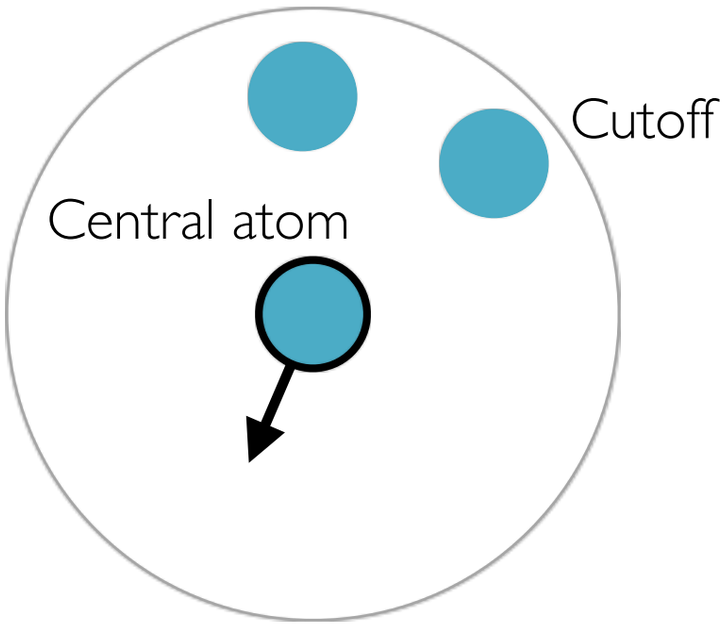
Input Space:

- **Configuration ρ**
- **3xM dimensional**
- **typically M~ 30-40**



Learning force fields: problem setup

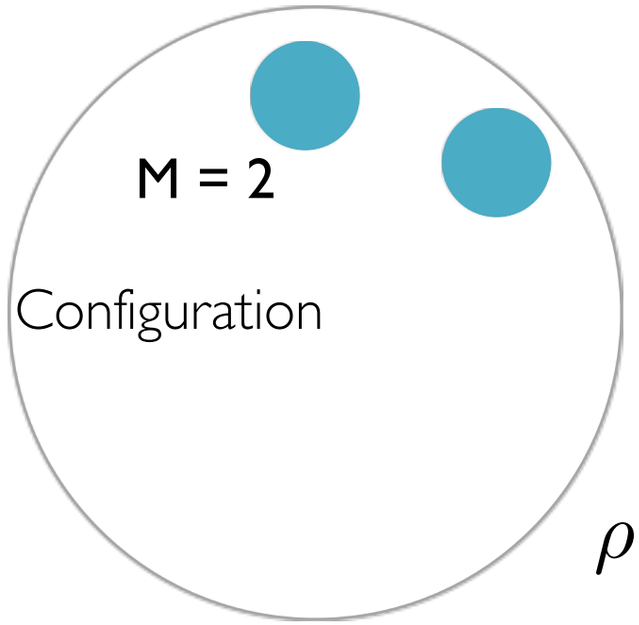
- Quantum system of *N* atoms at a specific time



- The computational cost of calculating energies and forces scales badly with N and also has high prefactors

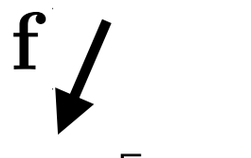
Input Space:

- **Configuration ρ**
- **3xM dimensional**
- **typically M~ 30-40**



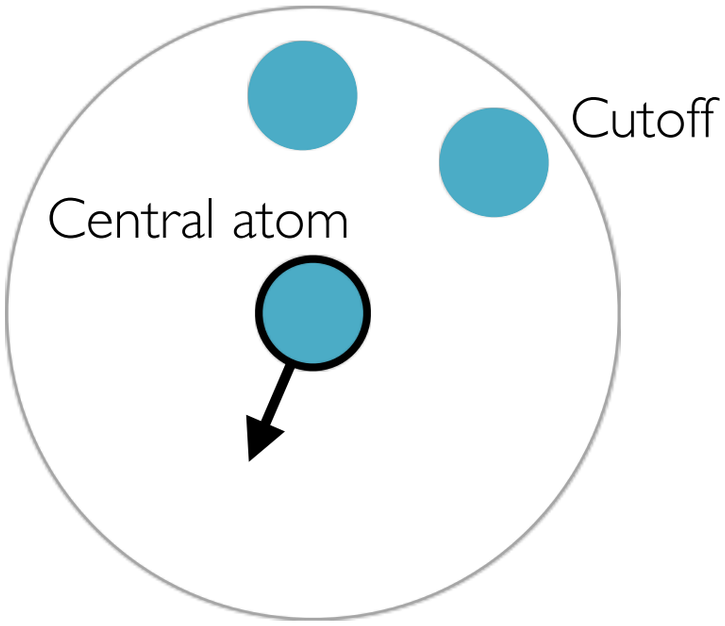
Output space:

- **Local energy ε of central atom**
- **The force f is obtained through differentiation**



Learning force fields: problem setup

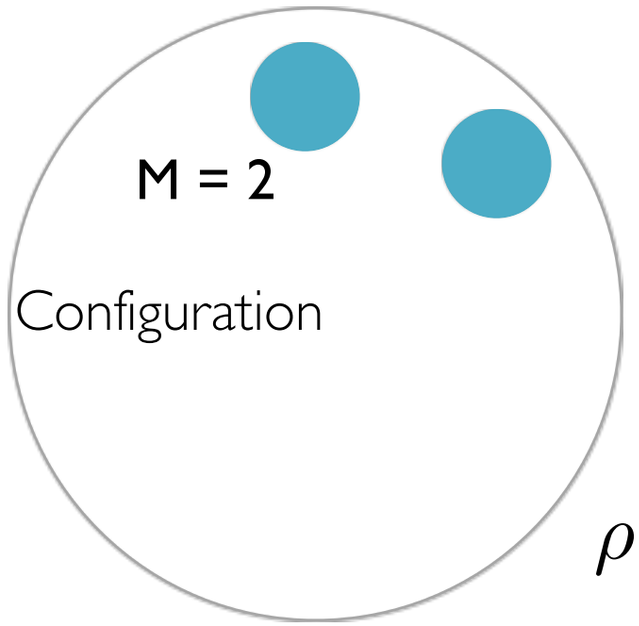
- Quantum system of *N* atoms at a specific time



- The computational cost of calculating energies and forces scales badly with N and also has high prefactors

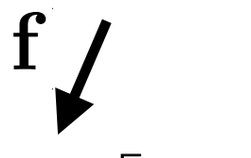
Input Space:

- **Configuration ρ**
- **3xM dimensional**
- **typically M~ 30-40**



Output space:

- **Local energy ε of central atom**
- **The force f is obtained through differentiation**



- The function **f** to be learned gives the force as a function of the local configuration ρ around a central atom
- The existence of such map can be assumed in most physical systems
- Linear scaling with *N* is guaranteed by the locality of the representation

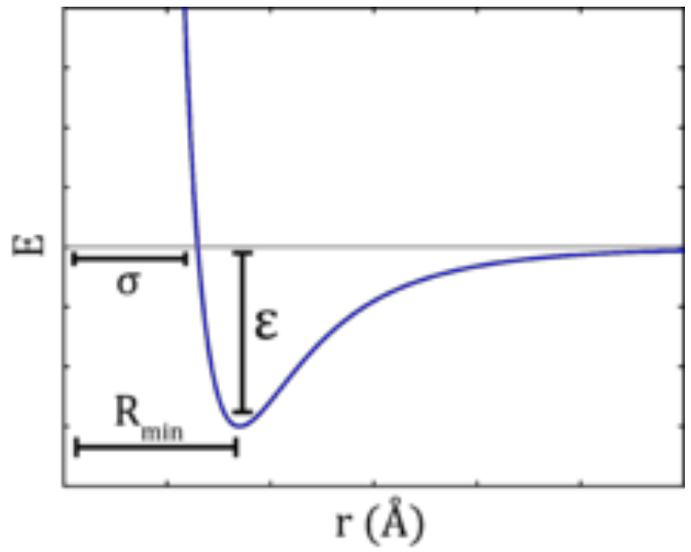
Traditional approach

- The traditional way of fitting potential energies and forces involves a lengthy **trial and error** procedure of careful selection of meaningful **parametric functional forms**

Traditional approach

- The traditional way of fitting potential energies and forces involves a lengthy **trial and error** procedure of careful selection of meaningful **parametric functional forms**

A Lennard-Jones parametric potential



Traditional approach

- The traditional way of fitting potential energies and forces involves a lengthy **trial and error** procedure of careful selection of meaningful **parametric functional forms**

A Lennard-Jones parametric potential

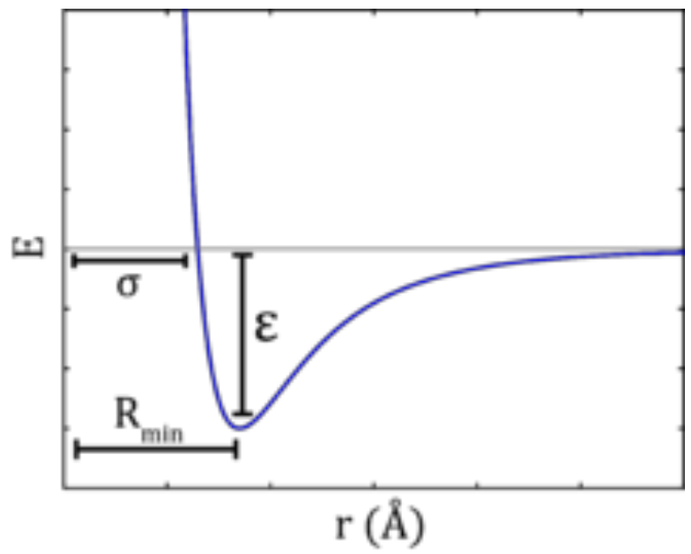


Table 1
Optimized values of fitting parameters of the EAM potentials for Ni, Al and Ni_3Al

Al		Ni		Ni_3Al	
Parameter	Value	Parameter	Value	Parameter	Value
r_c (nm)	0.6725	r_c (nm)	0.5168	r_c (nm)	0.6500
h_c (nm)	0.3294	h_c (nm)	0.3323	h_c (nm)	0.2658
V_0 (eV)	-3.5032×10^3	V_0 (eV)	-3.5126×10^3	V_0 (eV)	0.6068
r_1 (nm)	0.2858	r_1 (nm)	3.8673×10^{-5}	r_1 (nm)	0.4834
b_1	8.5951×10^{-2}	b_1	4.7067×10^{-3}	b_1	2.9013
b_2	5.0124×10^{-2}	b_2	0.15106	b_2	1.0001
δ (eV)	3.7503×10^3	δ (eV)	3.6046×10^3	δ (eV)	−3.4108
y	2.0080×10^1	y	1.9251×10^1	s_{Al}	0.9549
γ (1/nm)	4.2799×10^1	γ (1/nm)	1.6802×10^3	g_{Ni} (eV)	5.8549×10^1
B_0 (nm)	1.1927×10^4	B_0 (nm)	1.1914×10^4	g_{Al} (eV)	-1.8162×10^1
C_0 (1/nm^3)	8.6029×10^1	C_0 (1/nm^3)	2.0329×10^2		
r_0 (nm)	5.2755×10^{-2}	r_0 (nm)	−0.3138		
β	0.4890×10^{-2}	β	0.4890×10^{-2}		

Typical table of parameters of an EAM potential (from Mishin, Acta Materiali (2004))

The “new” data-driven approach

2007

PHYSICAL REVIEW LETTERS

Highlights Recent Accepted Collections Authors Referees Search Press About

Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces

Jörg Behler and Michele Parrinello
Phys. Rev. Lett. **98**, 146401 – Published 2 April 2007

2010

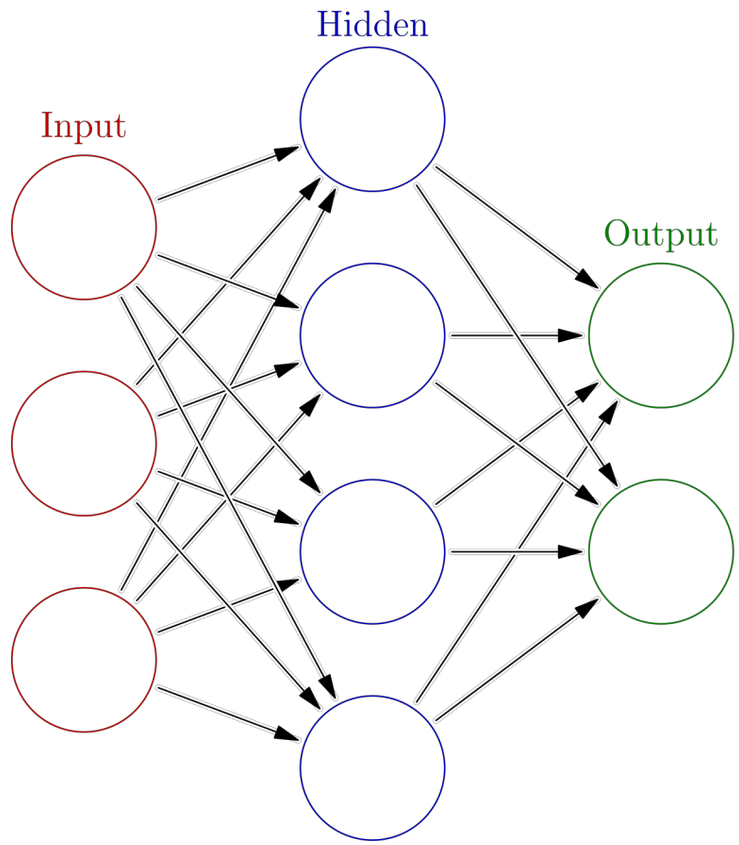
PHYSICAL REVIEW LETTERS

Highlights Recent Accepted Collections Authors Referees Search Press About

Featured in Physics

Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons

Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi
Phys. Rev. Lett. **104**, 136403 – Published 1 April 2010



- Very flexible models able to capture much more information from data

The “new” data-driven approach

2007

PHYSICAL REVIEW LETTERS

Highlights Recent Accepted Collections Authors Referees Search Press About

Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces

Jörg Behler and Michele Parrinello
Phys. Rev. Lett. **98**, 146401 – Published 2 April 2007

2010

PHYSICAL REVIEW LETTERS

Highlights Recent Accepted Collections Authors Referees Search Press About

Featured in Physics

Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons

Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi
Phys. Rev. Lett. **104**, 136403 – Published 1 April 2010

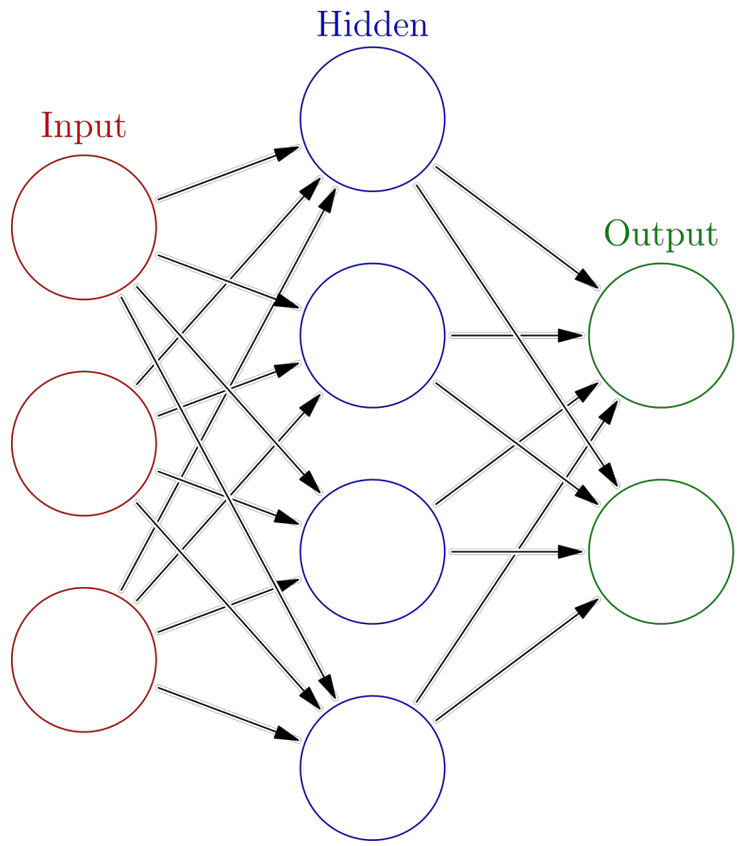
1994

IOPscience Journals Books Publishing Support Login Search IOPscience

Modelling and Simulation in Materials Science and Engineering

Neural networks in computational materials science: training algorithms

A J Skinner and J Q Broughton



“Our long term goal in applying neural networks is to demonstrate that they can be used to interpolate complex and expensive *ab initio* databases that describe interactions between atoms.”

- Very flexible models able to capture much more information from data

Bayesian inference 101

$\mathscr{D}$ Dataset (e.g. configurations and forces from a DFT calculation)

Bayesian inference 101

$\mathscr{D}$ Dataset (e.g. configurations and forces from a DFT calculation)

$p(f)$ Prior: Our beliefs about the function ***before*** we observe any data

(e.g., a Lennard Jones with Gaussian distributed parameters)

Bayesian inference 101

$\mathscr{D}$ Dataset (e.g. configurations and forces from a DFT calculation)

$p(f)$ **Prior**: Our beliefs about the function ***before*** we observe any data

(e.g., a Lennard Jones with Gaussian distributed parameters)

$p(\mathscr{D} \mid f)$ **Likelihood**: The probability to observe the data given a model prediction

Bayesian inference 101

$\mathscr{D}$ Dataset (e.g. configurations and forces from a DFT calculation)

$p(f)$ **Prior**: Our beliefs about the function ***before*** we observe any data

(e.g., a Lennard Jones with Gaussian distributed parameters)

$p(\mathscr{D} \mid f)$ **Likelihood**: The probability to observe the data given a model prediction

$p(f \mid \mathscr{D})$ **Posterior**: Our beliefs about the function ***after*** we have observe any data

Bayesian inference 101

$\mathscr{D}$ Dataset (e.g. configurations and forces from a DFT calculation)

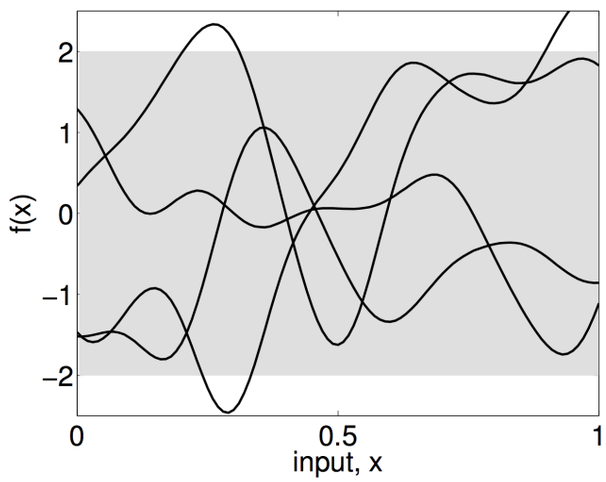
$p(f)$ **Prior**: Our beliefs about the function ***before*** we observe any data

(e.g., a Lennard Jones with Gaussian distributed parameters)

$p(\mathscr{D} \mid f)$ **Likelihood**: The probability to observe the data given a model prediction

$p(f \mid \mathscr{D})$ **Posterior**: Our beliefs about the function ***after*** we have observe any data

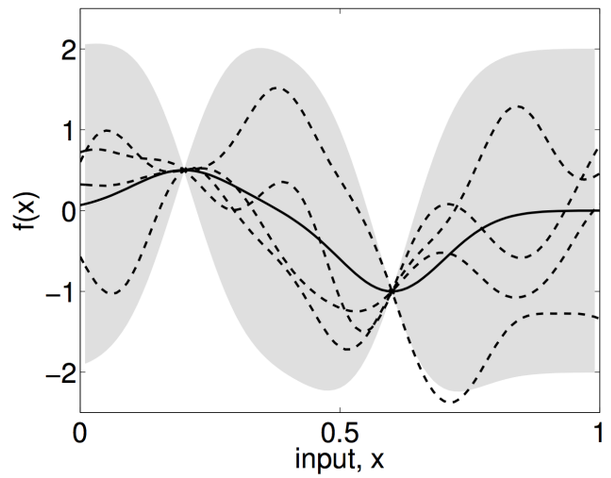
Prior



Bayes' Theorem

$$p(f \mid \mathscr{D}) = \frac{p(\mathscr{D} \mid f)p(f)}{p(\mathscr{D})}$$

Posterior



[Rasmussen and Williams 2006]

Probabilistic interpretation of least squares

Bayes' Theorem

$$p(\mathbf{w} \mid \mathscr{D}) \propto \frac{p(\mathscr{D} \mid \mathbf{w})p(\mathbf{w})}{p(\mathscr{D})}$$

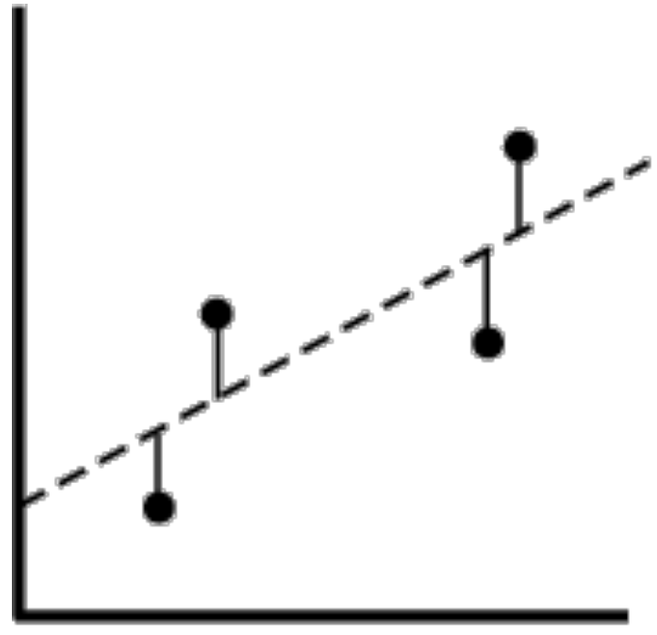
Probabilistic interpretation of least squares

Bayes' Theorem

$$p(\mathbf{w} \mid \mathscr{D}) \propto \frac{p(\mathscr{D} \mid \mathbf{w})p(\mathbf{w})}{p(\mathscr{D})}$$

Regularised squared error loss

$$L(\mathbf{w}) = \sum_i \|f_i - f_{\mathbf{w}}(\rho_i)\|^2 + \alpha\|\mathbf{w}\|^2$$



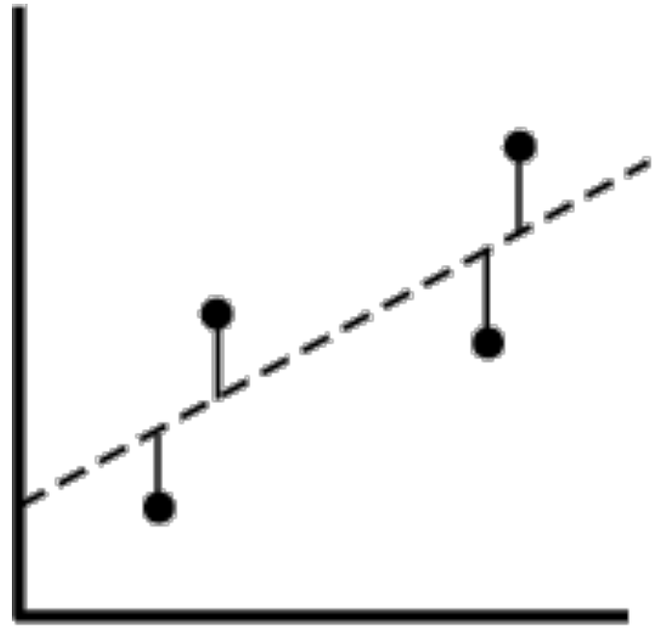
Probabilistic interpretation of least squares

Bayes' Theorem

$$p(\mathbf{w} \mid \mathscr{D}) \propto \frac{p(\mathscr{D} \mid \mathbf{w})p(\mathbf{w})}{p(\mathscr{D})}$$

Regularised squared error loss

$$L(\mathbf{w}) = \sum_i \|f_i - f_{\mathbf{w}}(\rho_i)\|^2 + \alpha\|\mathbf{w}\|^2$$



Gaussian prior and Gaussian likelihood

$$p(\mathbf{w}) \propto e^{-\|\mathbf{w}\|^2/2\sigma_{\mathbf{w}}^2}$$

$$p(f_i \mid \mathbf{w}) \propto e^{-\|f_i - f_{\mathbf{w}}(\rho_i)\|^2/2\sigma_f^2}$$

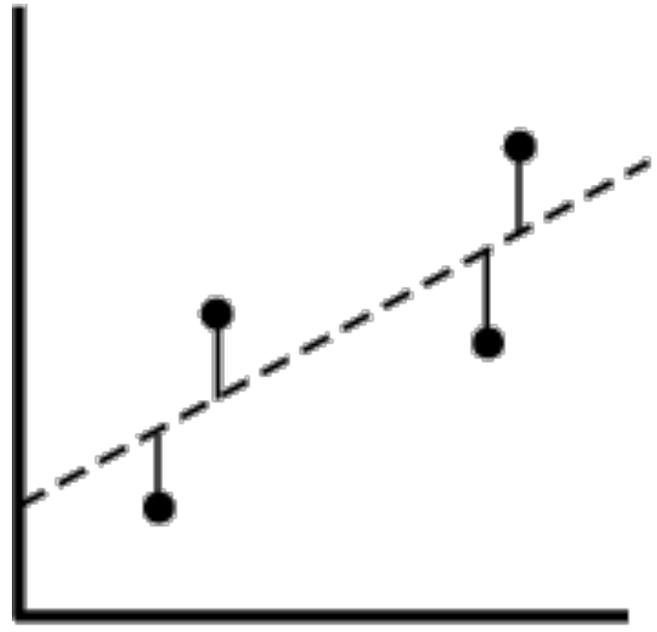
Probabilistic interpretation of least squares

Bayes' Theorem

$$p(\mathbf{w} \mid \mathscr{D}) \propto \frac{p(\mathscr{D} \mid \mathbf{w})p(\mathbf{w})}{p(\mathscr{D})}$$

Regularised squared error loss

$$L(\mathbf{w}) = \sum_i \|f_i - f_{\mathbf{w}}(\rho_i)\|^2 + \alpha\|\mathbf{w}\|^2$$



Gaussian prior and Gaussian likelihood

$$p(\mathbf{w}) \propto e^{-\|\mathbf{w}\|^2/2\sigma_{\mathbf{w}}^2}$$

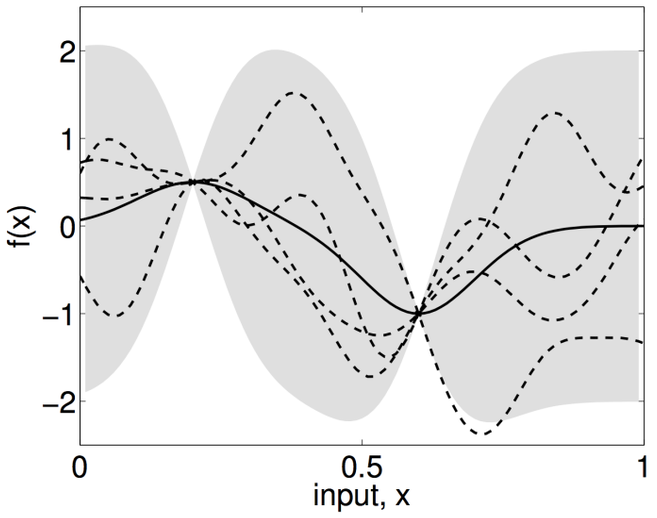
$$p(f_i \mid \mathbf{w}) \propto e^{-\|f_i - f_{\mathbf{w}}(\rho_i)\|^2/2\sigma_f^2}$$

$$-\ln p(\mathbf{w} \mid \mathscr{D}) \propto \sum_i \|f_i - f_{\mathbf{w}}(\rho_i)\|^2 + \frac{\sigma_f^2}{\sigma_{\mathbf{w}}^2}\|\mathbf{w}\|^2$$

Minimising a regularised squared loss corresponds to maximising a posterior distribution!

Prior information

Posterior

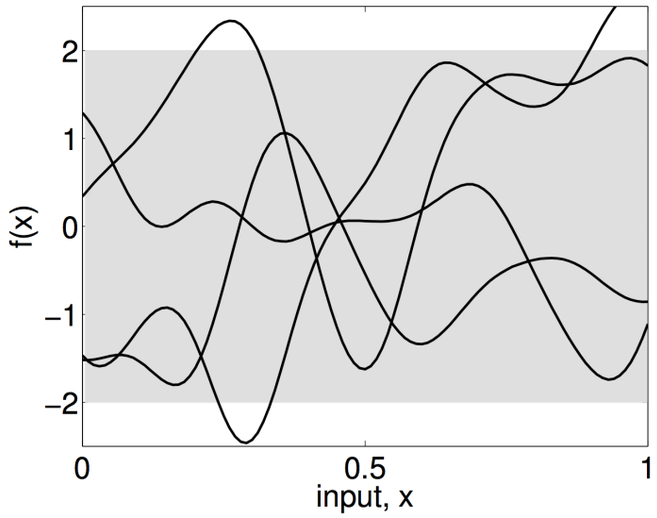


$$p(f \mid \mathcal{D}) \propto p(\mathcal{D} \mid f)p(f)$$

Likelihood

$$\mathcal{D} = (\rho, \mathbf{f})_1, (\rho, \mathbf{f})_2, \ldots$$

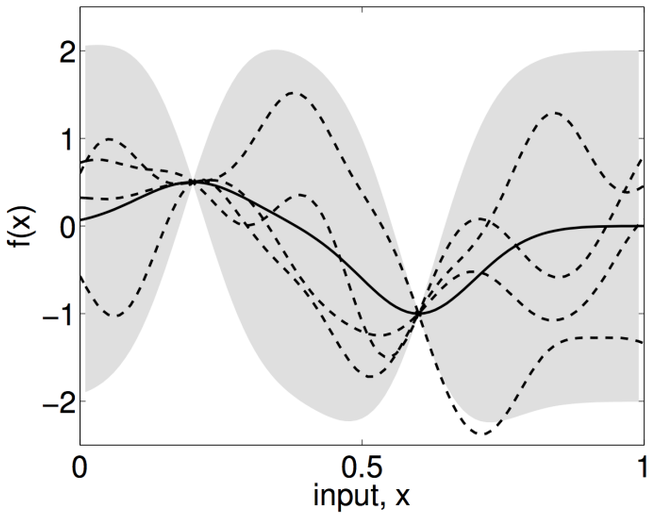
Prior



[Rasmussen and Williams 2006]

Prior information

Posterior

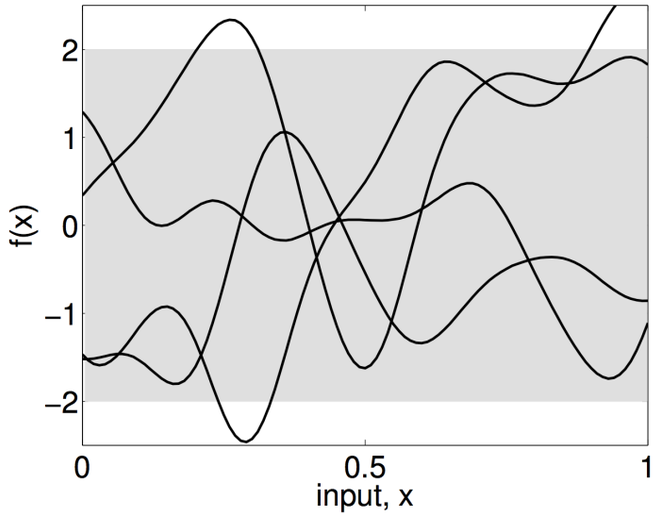


$$p(f \mid \mathcal{D}) \propto p(\mathcal{D} \mid f)p(f)$$

Likelihood

$$\mathcal{D} = (\rho, \mathbf{f})_1, (\rho, \mathbf{f})_2, \ldots$$

Prior



[Rasmussen and Williams 2006]

Traditional approaches:

- High prior information (parametric functional forms)
- Hence:
  - ✗ limited in their accuracy if prior is wrong
  - ✓ *simple to interpret*
  - ✓ *very fast*
  - ✓ *tend to be more transferable*

New ML approaches:

- Low prior information (Neural Networks, non parametric regression)
- Hence:
  - ✓ very flexible (potentially very accurate)
  - ✗ difficult to interpret
  - ✗ relatively slow
  - ✗ transferability can be problematic

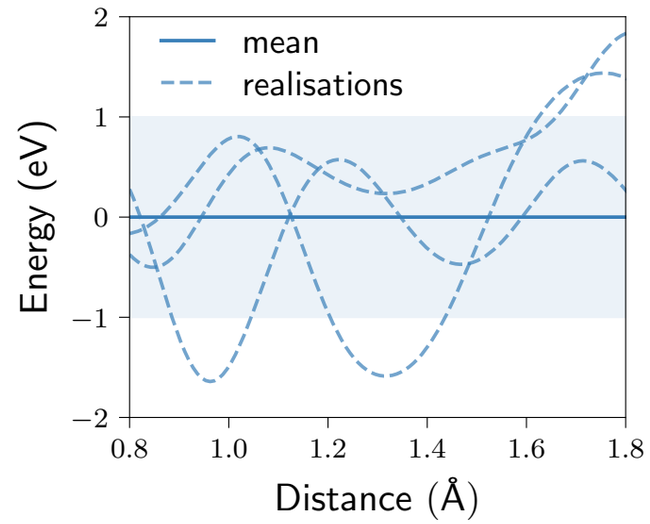
For robustness, for interpretability, and since we will never have infinite data:

We need a way to select the appropriate model complexity

Gaussian process prior

A Gaussian process can be used as prior distribution

$$p(f(\rho)) = \mathcal{GP}(0, k(x, x'))$$



Gaussian process prior

A Gaussian process can be used as prior distribution

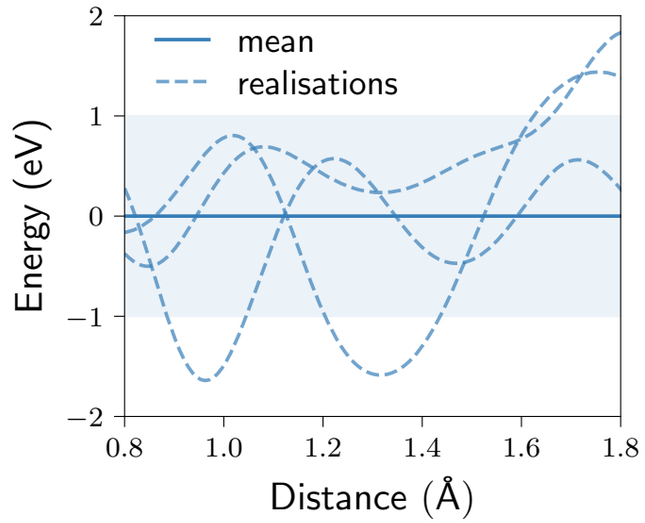
$$p(f(\rho)) = \mathcal{GP}(0, k(x, x'))$$

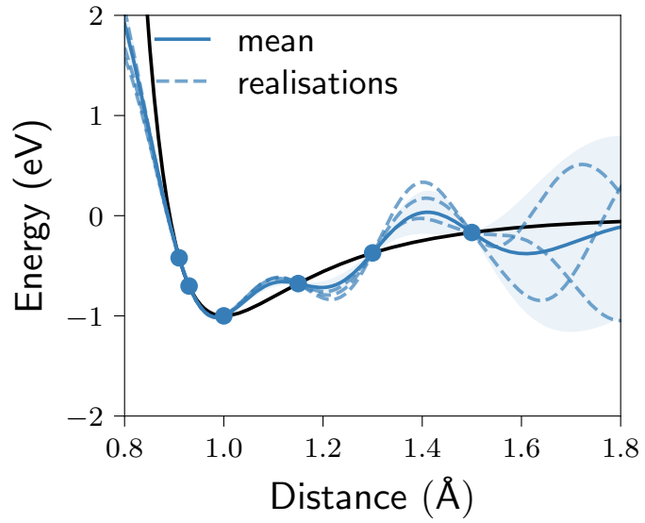
Using a Gaussian likelihood,
the posterior Gaussian process can be found analytically

$$p(f(x) \mid \mathcal{D}) = \mathcal{GP}(\hat{f}(x), \hat{k}(x, x'))$$

$$\hat{f}(x) = \sum_{i=1}^{N} k(x, x_i)\alpha_i$$ Predictions

$$\hat{\sigma}^2(x) = \hat{k}(x, x)$$ Predicted variance (measure of uncertainty)





Gaussian process prior

A Gaussian process can be used as prior distribution

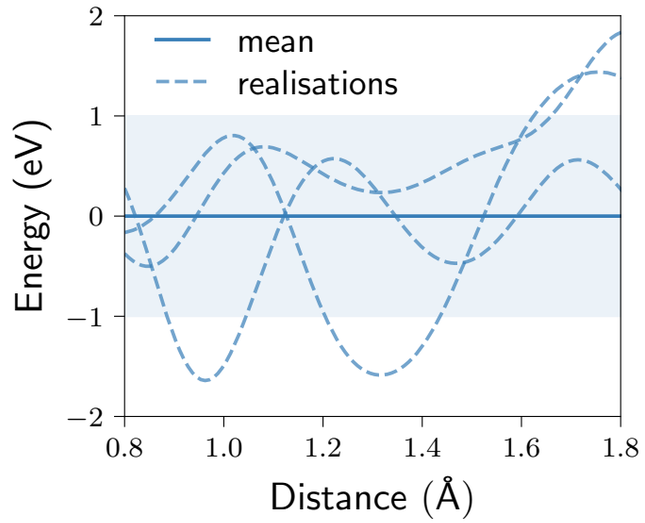
$$p(f(\rho)) = \mathcal{GP}(0, k(x, x'))$$

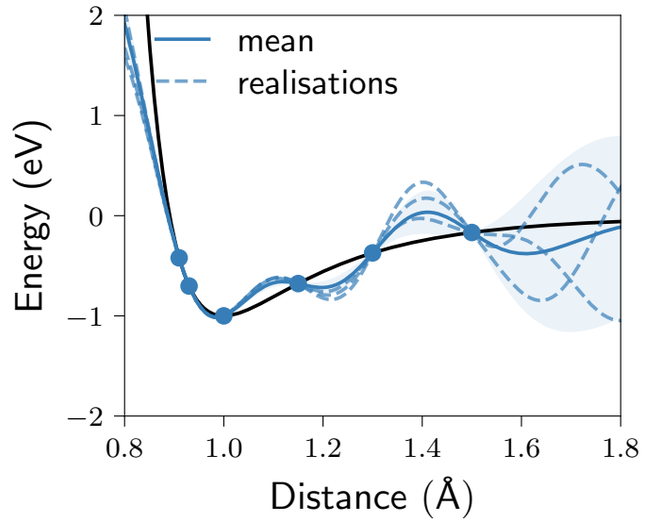
Using a Gaussian likelihood,
the posterior Gaussian process can be found analytically

$$p(f(x) \mid \mathcal{D}) = \mathcal{GP}(\hat{f}(x), \hat{k}(x, x'))$$

$$\hat{f}(x) = \sum_{i=1}^{N} k(x, x_i)\alpha_i \qquad \text{Predictions}$$

$$\hat{\sigma}^2(x) = \hat{k}(x, x) \qquad \text{Predicted variance (measure of uncertainty)}$$





- Very flexible regression model (provably equivalent to a NN with a single, infinite hidden layer)
- Principled uncertainty predictions (crucial for transferability and validation)
- Prior knowledge can be included into the kernel function

GP visualisation

Squared exponential kernel with length scale ℓ

$$k(x, x) = e^{-(x-x')^2/2\ell^2}$$

Gaussian likelihood with "noise" parameter σ_n^2

$$p(f_i \mid f(x_i)) \propto e^{-(f_i - f(x_i))^2/2\sigma_n^2}$$

Animation by

Dr. Ádám Fekete
King's College London

https://mybinder.org/v2/gh/fekad/gp-visualisation.git/master?filepath=gp-visualisation.ipynb

GP visualisation

Squared exponential kernel with length scale ℓ

$$k(x, x) = e^{-(x-x')^2/2\ell^2}$$

Gaussian likelihood with "noise" parameter σ_n^2

$$p(f_i \mid f(x_i)) \propto e^{-(f_i - f(x_i))^2/2\sigma_n^2}$$

Posterior Gaussian process **?**

Animation by

Dr. Ádám Fekete
King's College London

https://mybinder.org/v2/gh/fekad/gp-visualisation.git/master?filepath=gp-visualisation.ipynb

GP learning: design of kernel function

- A Gaussian Process regression assumes a normal distribution for $p(\varepsilon(\rho))$, a GP

$$p(\varepsilon(\rho)) = \mathcal{GP}(0, k(\rho, \rho'))$$

- Kernels should respects all the symmetries of the force and possess a controllable degree of prior information

GP learning: design of kernel function

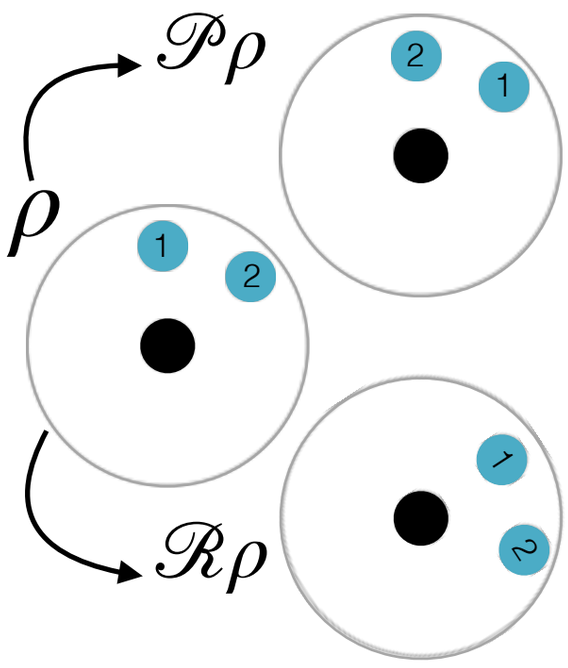
- A Gaussian Process regression assumes a normal distribution for $p(\varepsilon(\rho))$, a GP

$$p(\varepsilon(\rho)) = \mathcal{GP}(0, k(\rho, \rho'))$$

- Kernels should respects all the symmetries of the force and possess a controllable degree of prior information

Symmetries

- Symmetry properties should be encoded into the kernel function



Permutation invariance

$$k(\mathscr{P}\rho, \mathscr{P}'\rho') = k(\rho, \rho')$$

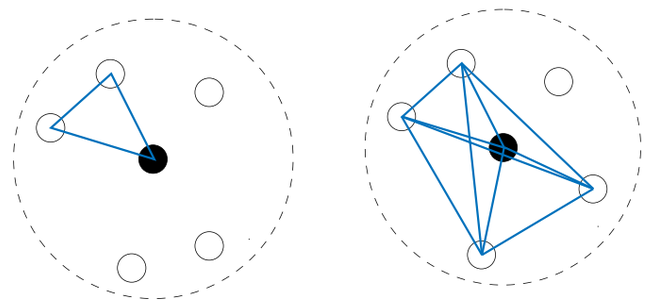
Rotation invariance

$$k(\mathscr{R}\rho, \mathscr{R}'\rho') = k(\rho, \rho')$$

Interaction Order

- Interaction order can also be encoded into the kernel, controlling the degree of complexity

$$\frac{\partial^n k_n(\rho, \rho')}{\partial \mathbf{r}_1 \cdots \partial \mathbf{r}_n} = 0$$



3- and 5-body interactions

Part II

Choice of kernel function and computational speedups

Claudio Zeni

Physics Department, King's College London

GP learning: design of kernel function

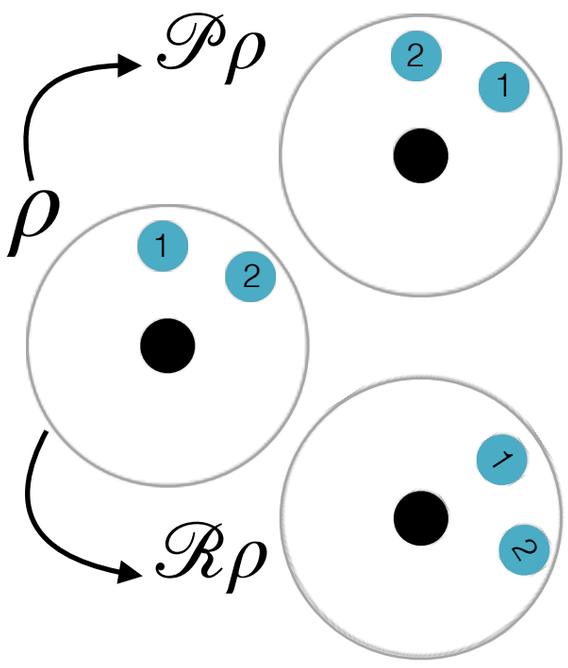
- A Gaussian Process regression assumes a normal distribution for $p(\varepsilon(\rho))$, a GP

$$p(\varepsilon(\rho)) = \mathcal{GP}(0, k(\rho, \rho'))$$

- Kernels should respects all the symmetries of the force and possess a controllable degree of prior information

Symmetries

- Symmetry properties should be encoded into the kernel function



Permutation invariance

$$k(\mathscr{P}\rho, \mathscr{P}'\rho') = k(\rho, \rho')$$

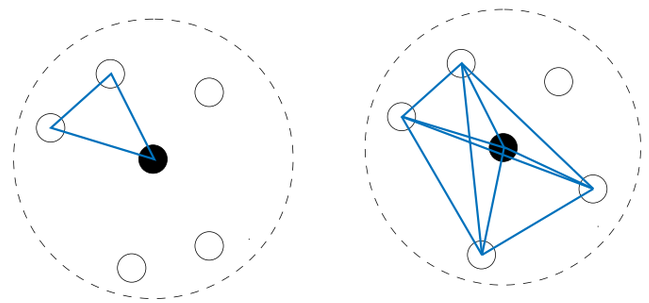
Rotation invariance

$$k(\mathscr{R}\rho, \mathscr{R}'\rho') = k(\rho, \rho')$$

Interaction Order

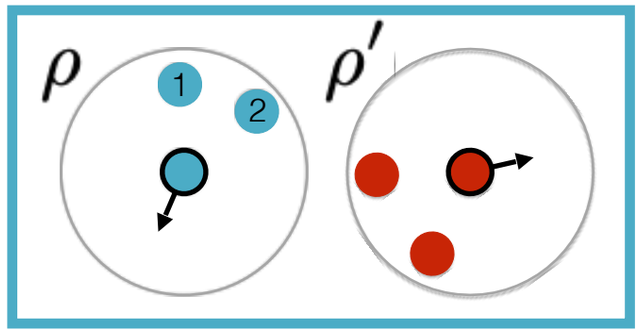
- Interaction order can also be encoded into the kernel, controlling the degree of complexity

$$\frac{\partial^n k_n(\rho, \rho')}{\partial \mathbf{r}_1 \cdots \partial \mathbf{r}_n} = 0$$



3- and 5-body interactions

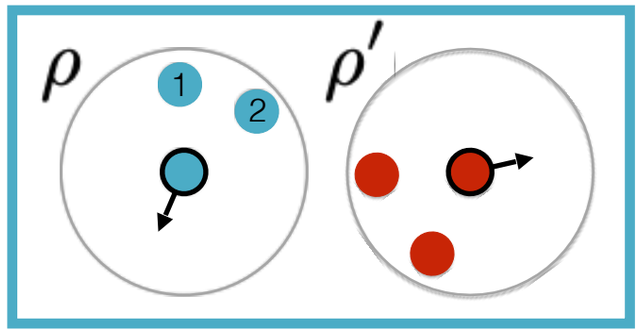
Building *n*-body kernels



Building *n*-body kernels

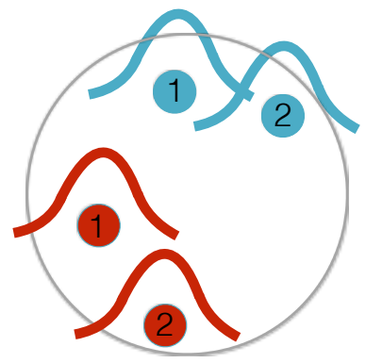
- Permutation invariant representation of a local environment

$$\rho(\mathbf{r}, \{\mathbf{r}_i\}) = \sum_{i=1}^{M} \mathcal{N}(\mathbf{r}_i, \sigma^2) \quad =$$



- Dot product induces a 2-body kernel

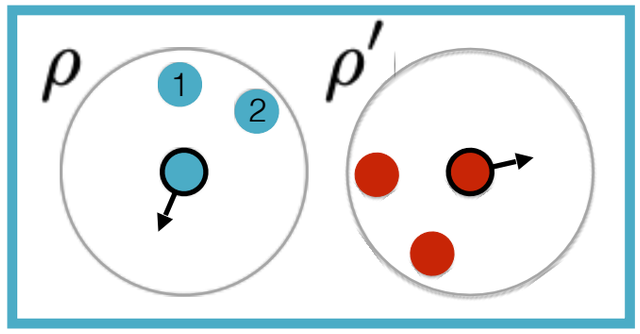
$$k_2(\rho, \rho') = \int d\mathbf{r}\rho(\mathbf{r})\rho'(\mathbf{r}) \quad =$$



Building *n*-body kernels

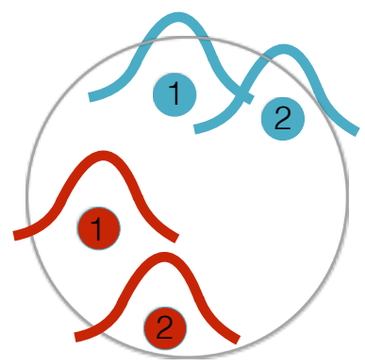
- Permutation invariant representation of a local environment

$$\rho(\mathbf{r}, \{\mathbf{r}_i\}) = \sum_{i=1}^{M} \mathcal{N}(\mathbf{r}_i, \sigma^2)$$



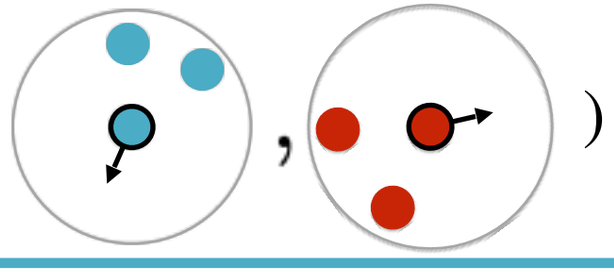
- Dot product induces a 2-body kernel

$$k_2(\rho, \rho') = \int d\mathbf{r}\, \rho(\mathbf{r}) \rho'(\mathbf{r})$$



- Haar integration imposes rotational symmetry

$$k_n^s(\rho, \rho') = \int_{O(3)} d\mathscr{R}\, k_n(\rho, \mathscr{R}\rho') \sim \int k_n($$

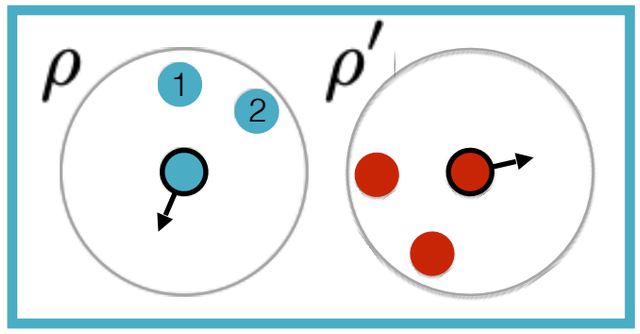


$$)$$

Building *n*-body kernels

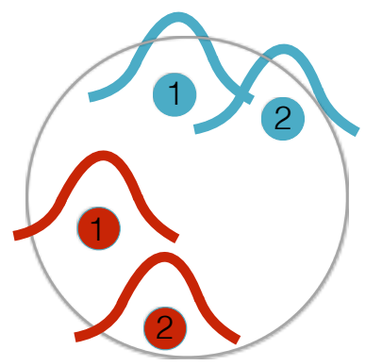
- Permutation invariant representation of a local environment

$$\rho(\mathbf{r}, \{\mathbf{r}_i\}) = \sum_{i=1}^{M} \mathcal{N}(\mathbf{r}_i, \sigma^2)$$



- Dot product induces a 2-body kernel

$$k_2(\rho, \rho') = \int d\mathbf{r}\, \rho(\mathbf{r}) \rho'(\mathbf{r})$$



- Haar integration imposes rotational symmetry

$$k_n^s(\rho, \rho') = \int_{O(3)} d\mathscr{R}\, k_n(\rho, \mathscr{R}\rho') \sim \int k_n(\quad , \quad)$$

SOAP kernel

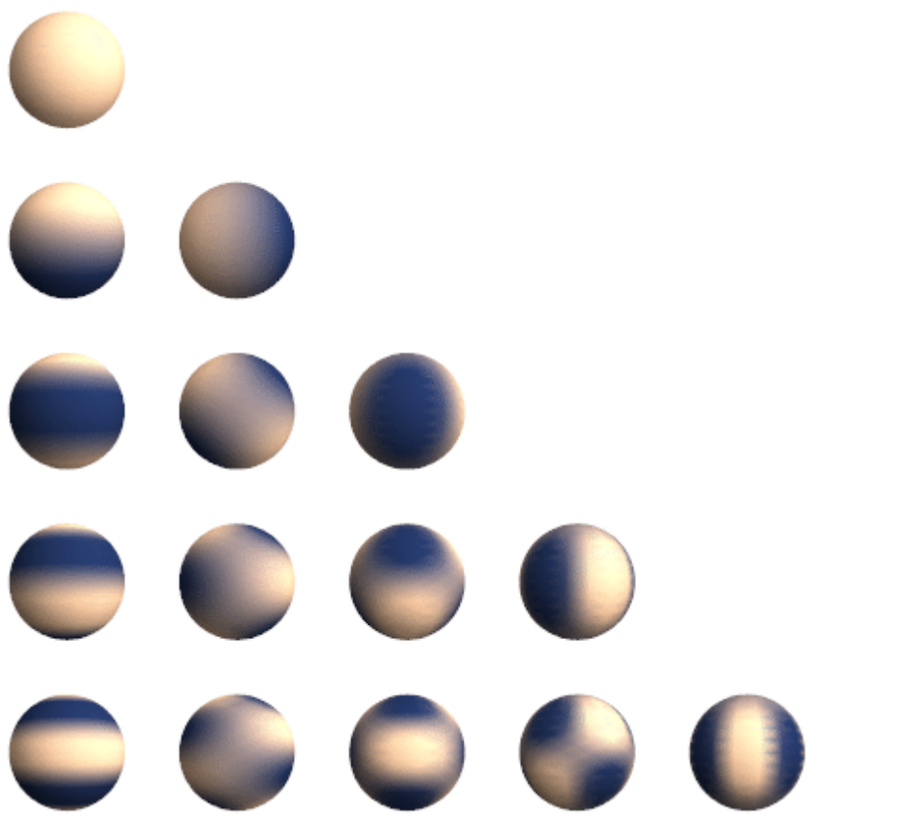
- Start by representing the environment of an atom as its neighbour density

$$\rho(\mathbf{r}) = \sum_{i=1}^{M} \mathcal{N}\left(\mathbf{r}_i, \sigma^2\right) f_{cut}(|\mathbf{r}_i|)$$

SOAP kernel

- Start by representing the environment of an atom as its neighbour density

$$\rho(\mathbf{r}) = \sum_{i=1}^{M} \mathcal{N}\left(\mathbf{r}_i, \sigma^2\right) f_{cut}(|\mathbf{r}_i|)$$



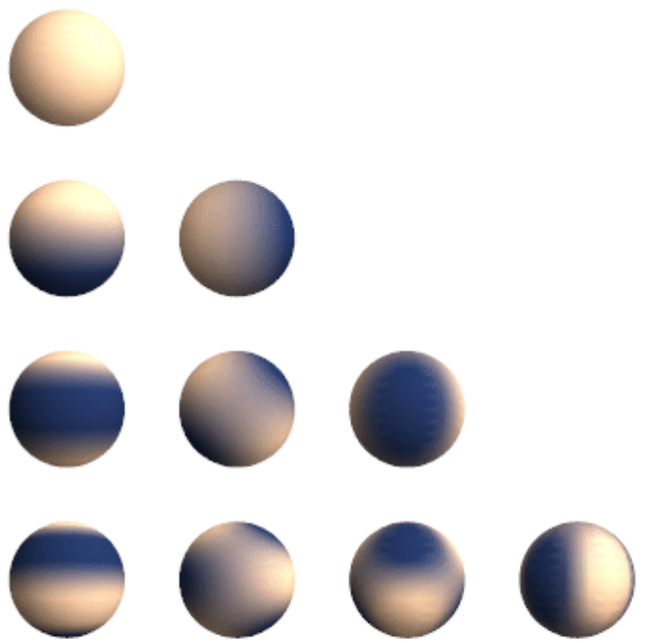
- Expand the neighbour density in a basis of spherical harmonics and radial functions:

$$\rho(\mathbf{r}) = \sum_{nlm} c^i_{lmn} Y_{lm}\left(\hat{\mathbf{r}}\right) g_n(r)$$

SOAP kernel

- Start by representing the environment of an atom as its neighbour density

$$\rho(\mathbf{r}) = \sum_{i=1}^{M} \mathcal{N}\left(\mathbf{r}_i, \sigma^2\right) f_{cut}(|\mathbf{r}_i|)$$

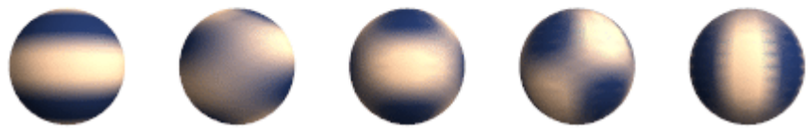


- Expand the neighbour density in a basis of spherical harmonics and radial functions:

$$\rho(\mathbf{r}) = \sum_{nlm} c^i_{lmn} Y_{lm}\left(\hat{\mathbf{r}}\right) g_n(r)$$

- Take the power spectrum and normalize it:

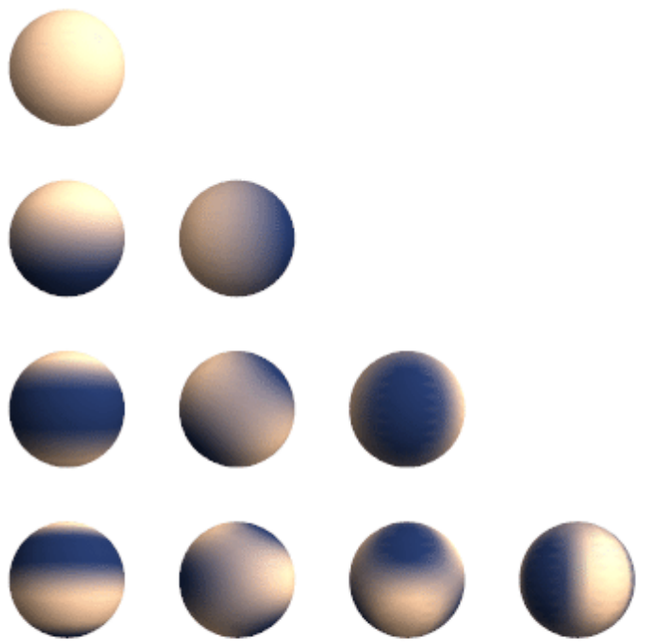
$$\tilde{p}_{nn'l} = \sum_{m=-l}^{l} c^{i*}_{nlm} c^{i}_{n'lm} \qquad \mathbf{p}^i = \tilde{\mathbf{p}}^{\mathbf{i}} / |\tilde{\mathbf{p}}^{\mathbf{i}}|$$



SOAP kernel

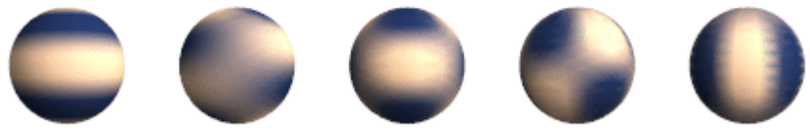
- Start by representing the environment of an atom as its neighbour density

$$\rho(\mathbf{r}) = \sum_{i=1}^{M} \mathcal{N}\left(\mathbf{r}_i, \sigma^2\right) f_{cut}(|\mathbf{r}_i|)$$



- Expand the neighbour density in a basis of spherical harmonics and radial functions:

$$\rho(\mathbf{r}) = \sum_{nlm} c^i_{lmn} Y_{lm}\left(\hat{\mathbf{r}}\right) g_n(r)$$



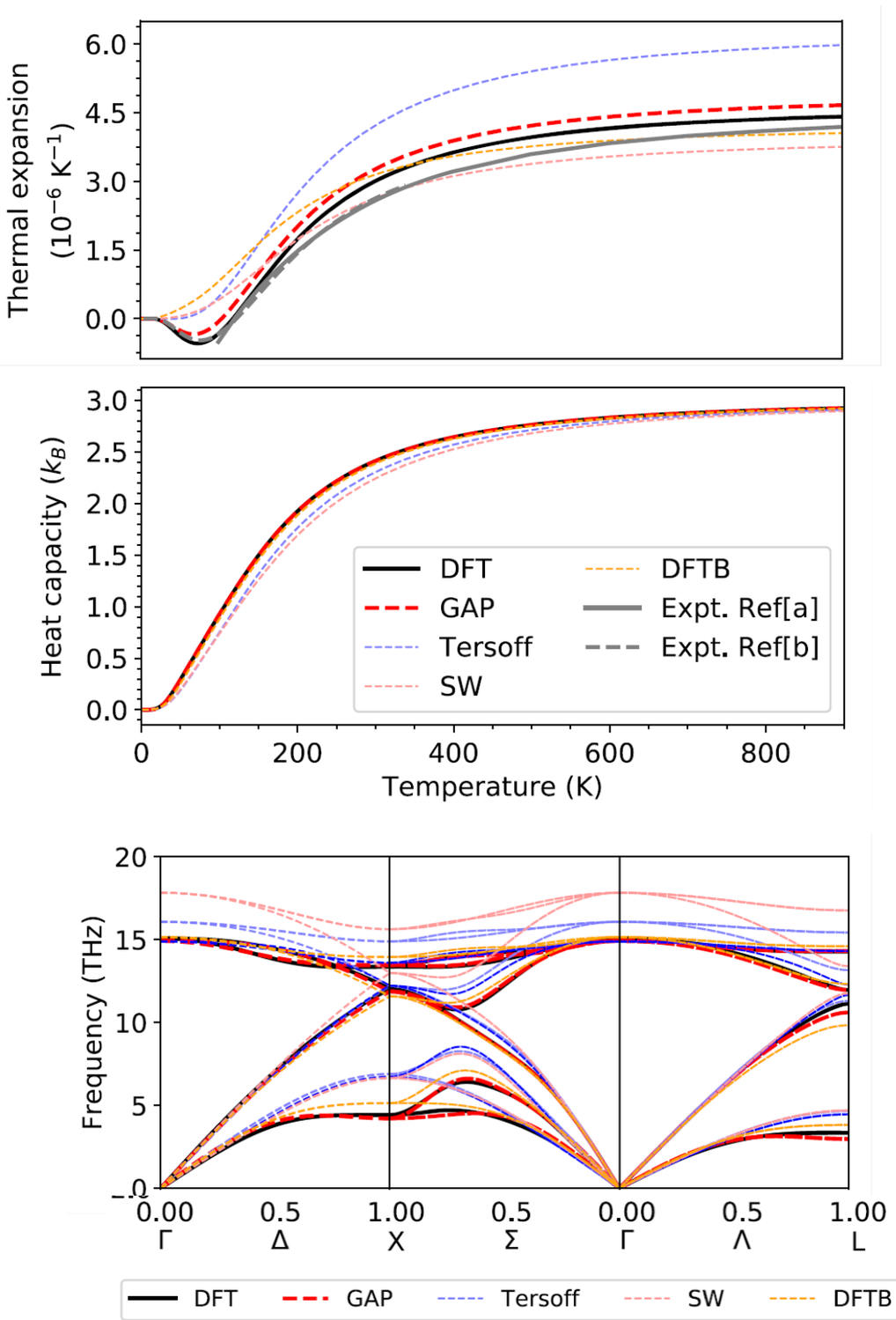
- Take the power spectrum and normalize it:

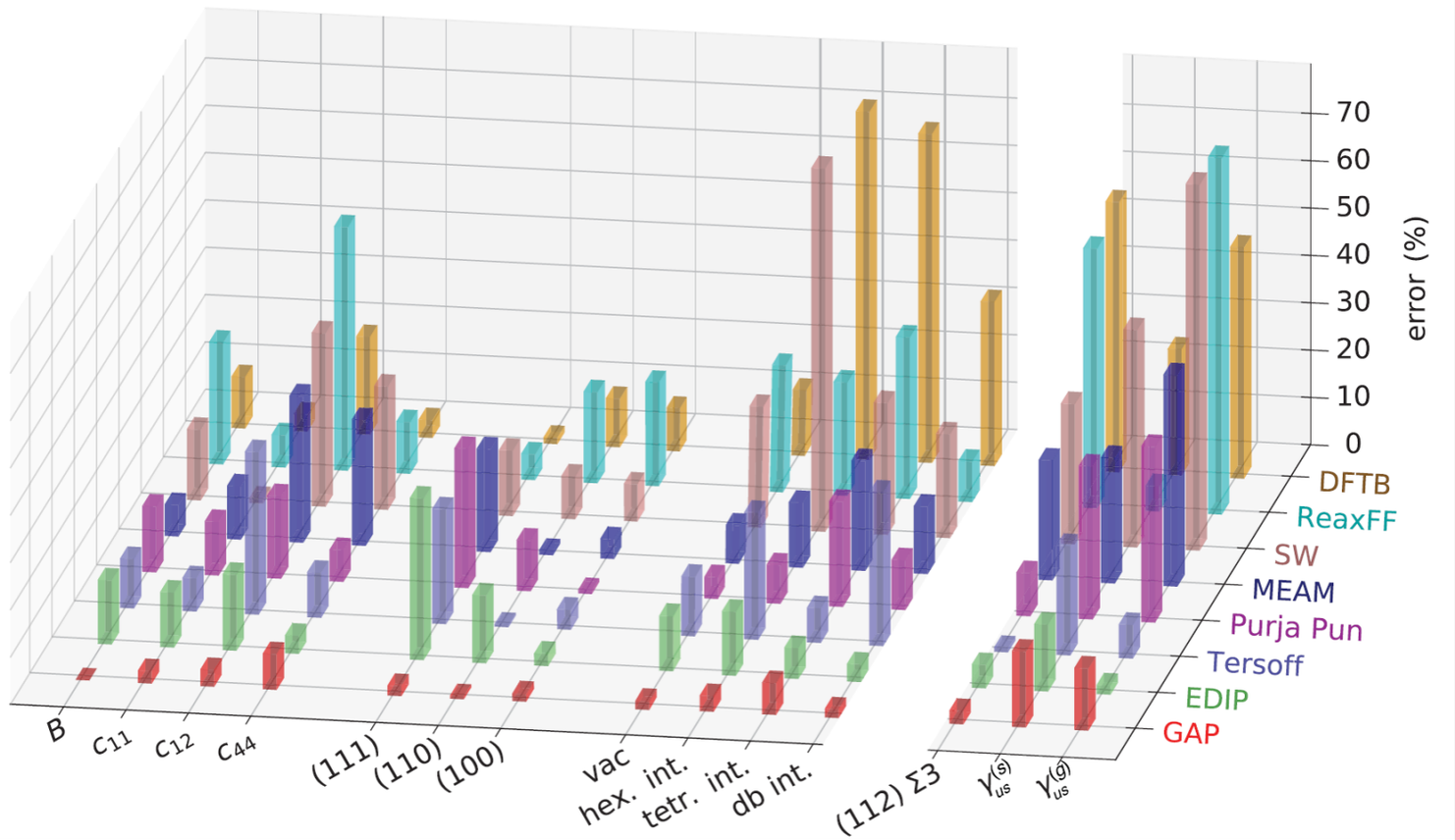
$$\tilde{p}_{nn'l} = \sum_{m=-l}^{l} c^{i*}_{nlm} c^{i}_{n'lm} \qquad \mathbf{p}^i = \tilde{\mathbf{p}}^{\mathbf{i}} / |\tilde{\mathbf{p}}^{\mathbf{i}}|$$

- The SOAP kernel is the scalar product elevated to an integer power:

$$k\left(\mathbf{r_i}, \mathbf{r_j}\right) = A^2 \left|\mathbf{p}^i \cdot \mathbf{p}^j\right|^{\xi}$$

Machine Learning a General-Purpose Interatomic Potential for Silicon



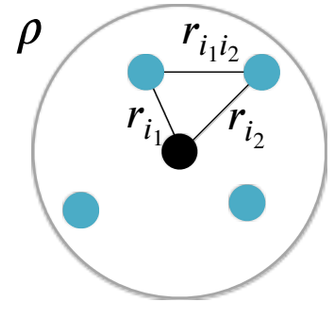


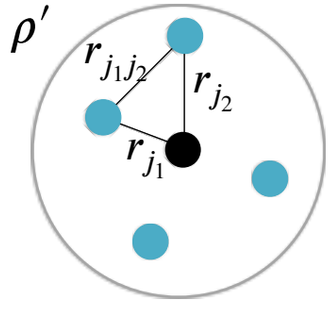
Machine Learning a General-Purpose Interatomic Potential for Silicon
A. P. Bartók, J. Kermode, N. Bernstein, G. Csányi
(PRX 2018)

Explicit *n*-body kernels

$$k_2^s(\rho,\rho') = \sum_{\substack{i\in\rho\\ j\in\rho'}} \mathrm{e}^{-(r_i - r'_j)^2/2\ell^2},$$

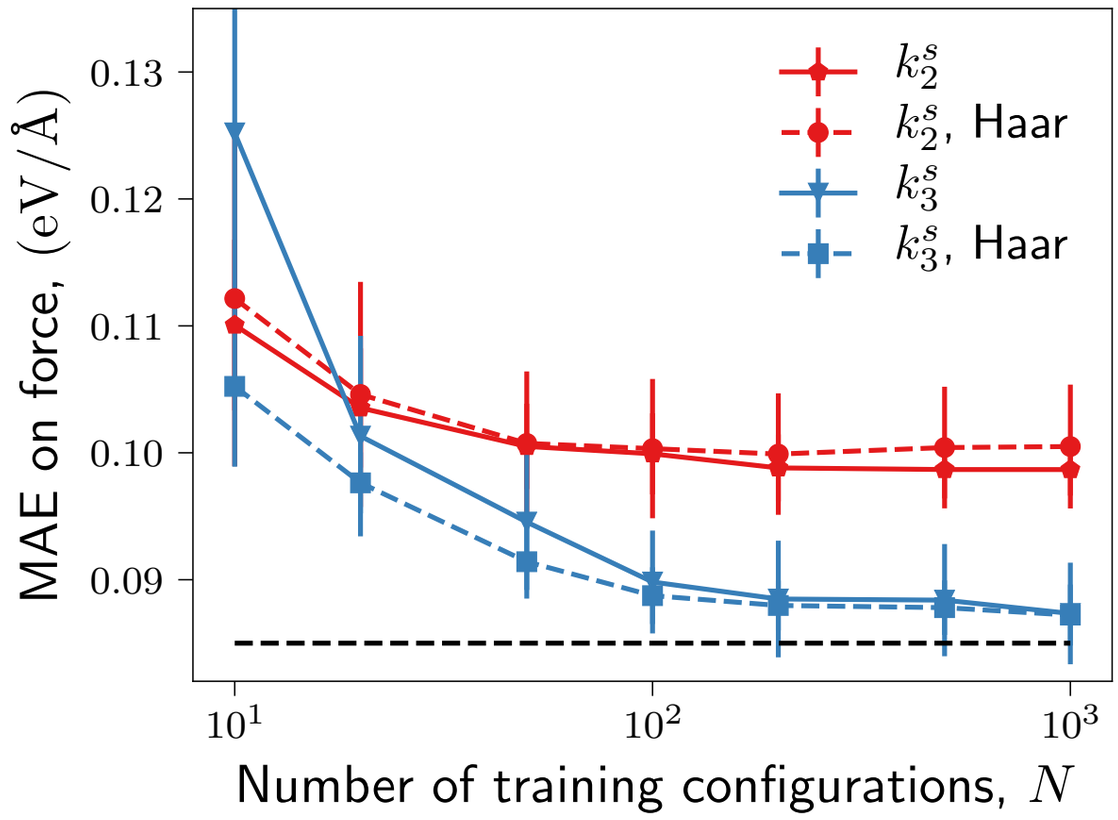
$$k_3^s(\rho,\rho') = \sum_{\substack{i_1>i_2\in\rho\\ j_1>j_2\in\rho'}} \sum_{\mathbf{P}\in\mathcal{P}} \mathrm{e}^{-\|(r_{i_1},r_{i_2},r_{i_1i_2})^{\mathrm{T}} - \mathbf{P}(r'_{j_1},r'_{j_2},r'_{j_1j_2})^{\mathrm{T}}\|^2/2\ell^2}.$$





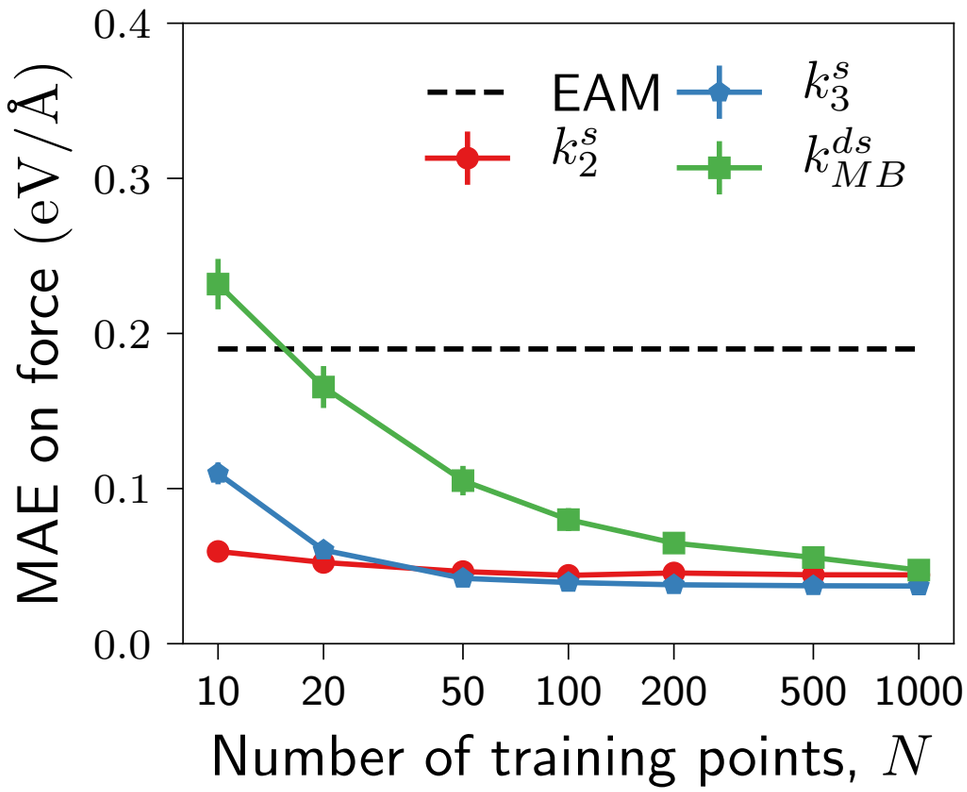
$$k^s_{MB}(\rho,\rho') = \mathrm{e}^{-(k_3^s(\rho,\rho)+k_3^s(\rho',\rho')-2k_3^s(\rho,\rho'))/2\ell^2}$$

- **Simple low-*n* kernels** can be easily constructed directly on invariant degrees of freedom
- Very intuitive and computationally efficient!

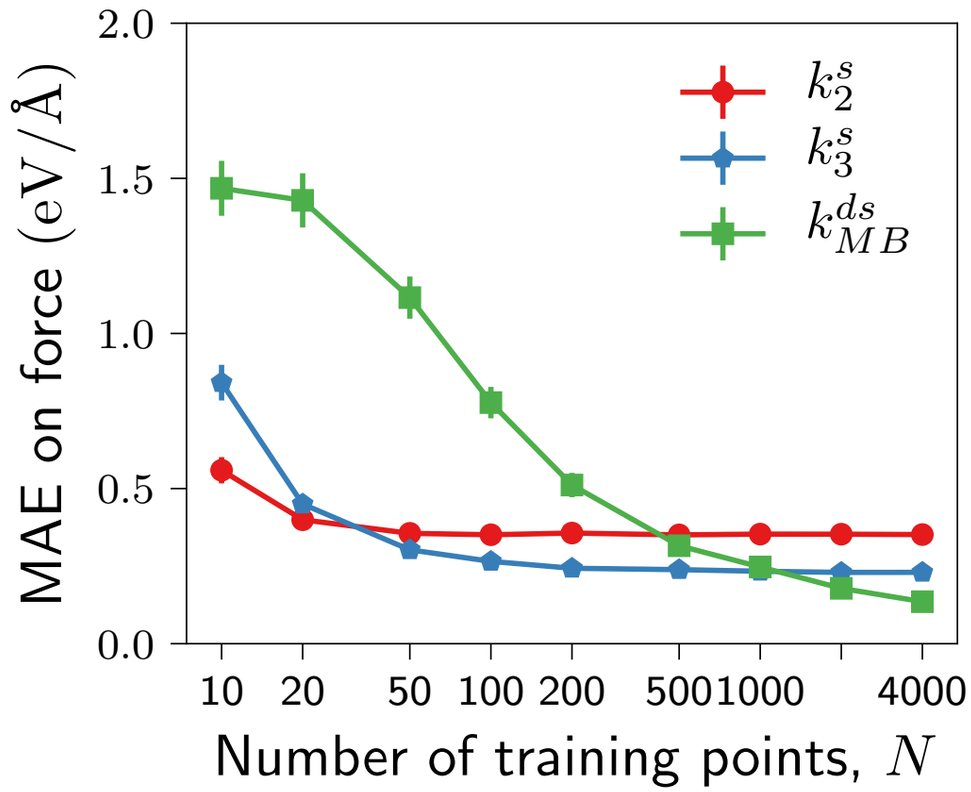


Choosing the interaction order: heuristics

Crystalline Nickel: *learning curves*



Amorphous Silicon: *learning curves*



- GP-FFs are always sensibly more accurate on target forces than traditional FFs
- Learning with a **higher order** kernel is **more accurate** but requires **more data**
- The optimal order depends on the material and it can be chosen as the smallest ***n*** compatible with target accuracy
- Low order models are often optimal

AG, C. Zeni, A. De Vita,
Efficient non-parametric n-body force fields from machine learning (PRB 2018)

Choosing the interaction order: marginal lik.

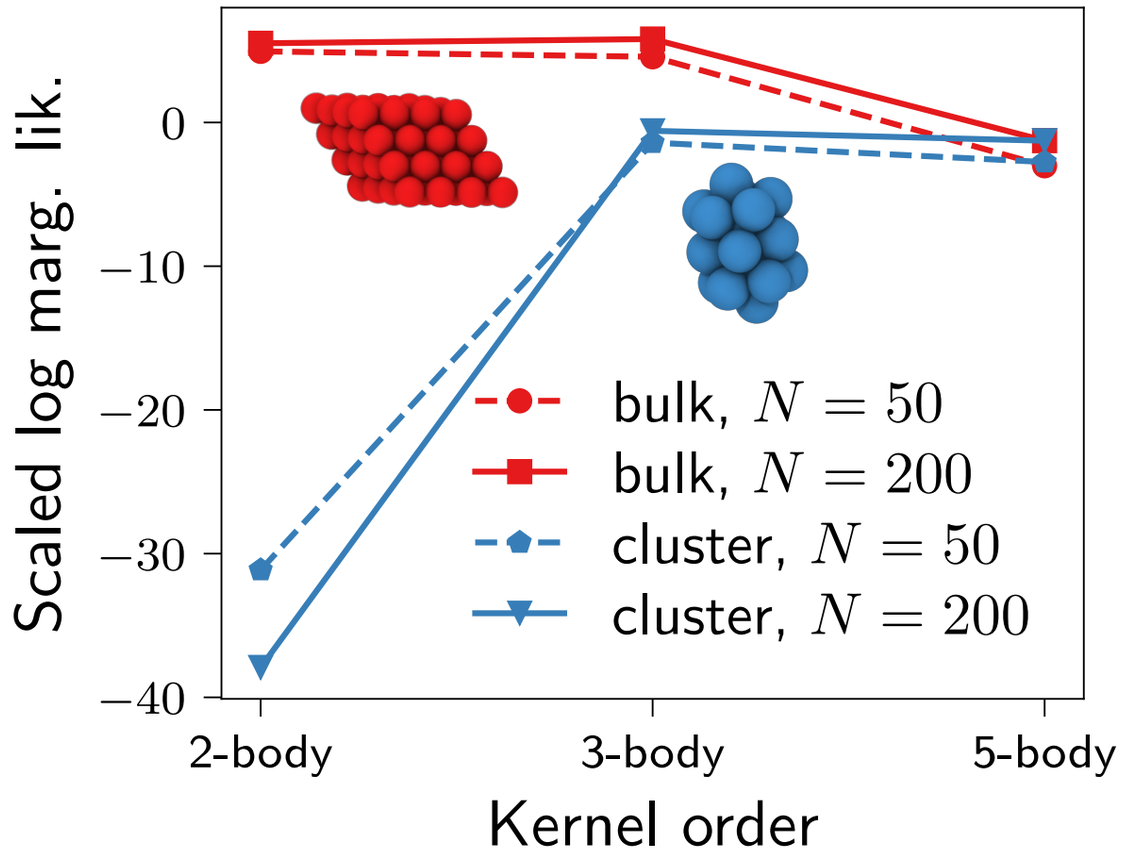
- The optimal interaction order can also be selected as the one achieving the maximal marginal likelihood

Marginal likelihood

$$\ln p(\boldsymbol{\varepsilon} \mid \boldsymbol{\rho}, \mathscr{M}_n) \propto -\frac{1}{2}\boldsymbol{\varepsilon}^{\mathrm{T}}\mathbf{K}_n^{-1}\boldsymbol{\varepsilon} - \frac{1}{2}\ln|\mathbf{K}_n|$$

Fit Complexity

Nickel systems: *marginal likelihood*

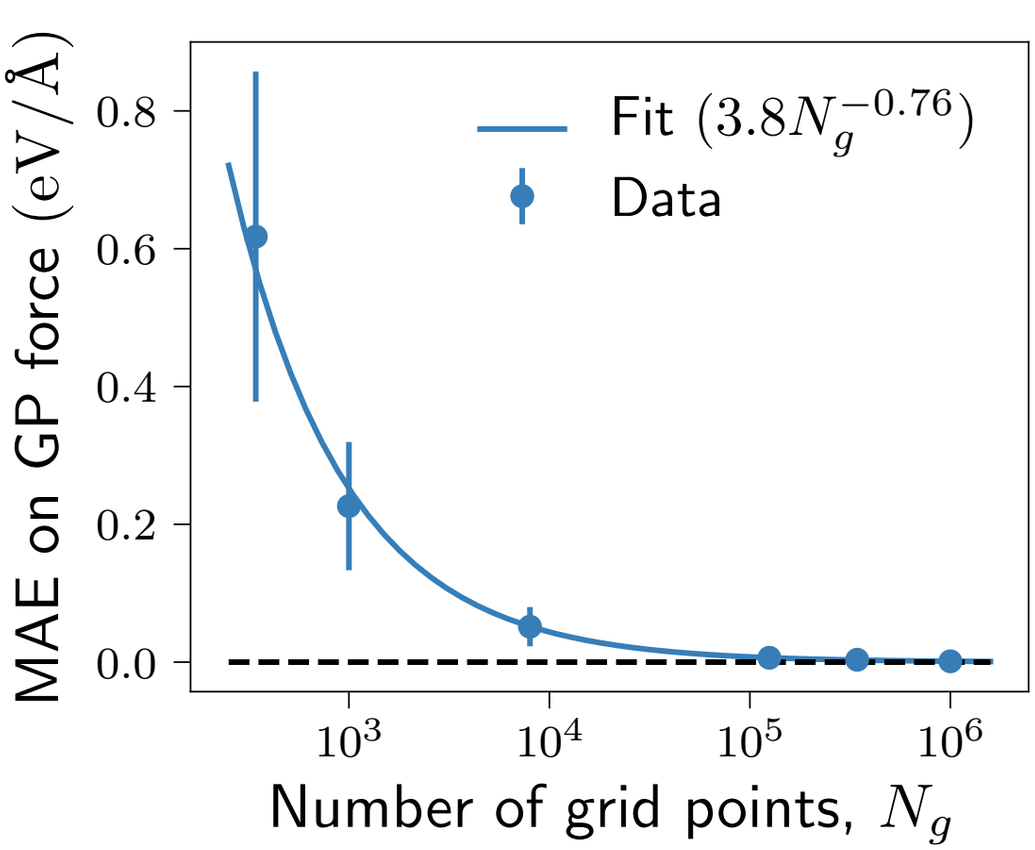


AG, C. Zeni, A. Fekete, A. De Vita,
Bayesian construction of n-body force fields via Gaussian process regression

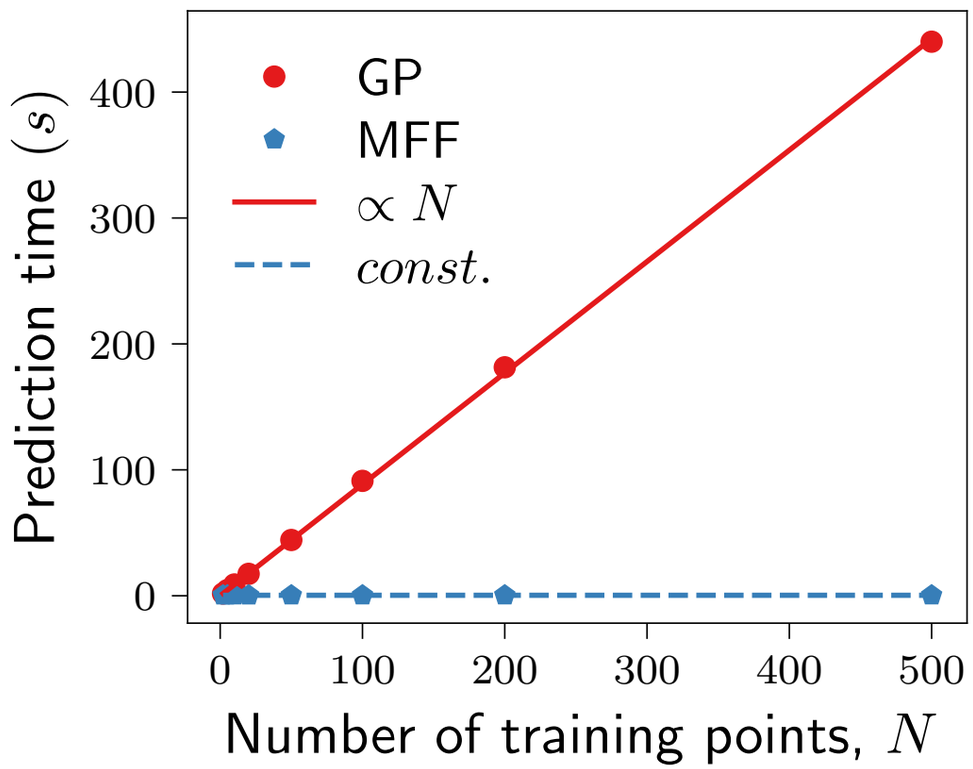
Speeding up low order models

- *n*-body kernels predictions can be decomposed in ***n*-body contributions**
- Hence one can tabulate the GP over the degrees of freedom of *n* particles (e.g. distances and angles) obtaining a very fast **non-parametric** *n*-body force field
- This mapped force field (MFF) is **identical** to the original one but substantially **faster**

Mapping: Convergence of the MFF to the GP force field as a function of the number of grid points



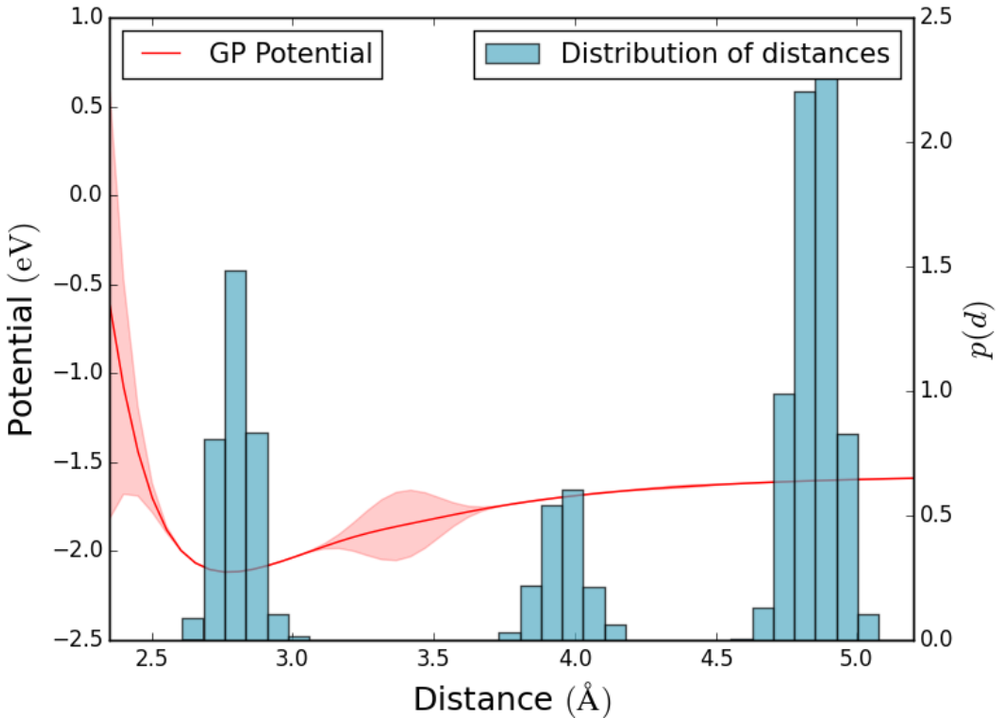
Speedup: Prediction time of GP potential and remapped potential vs *N* (number of training points)



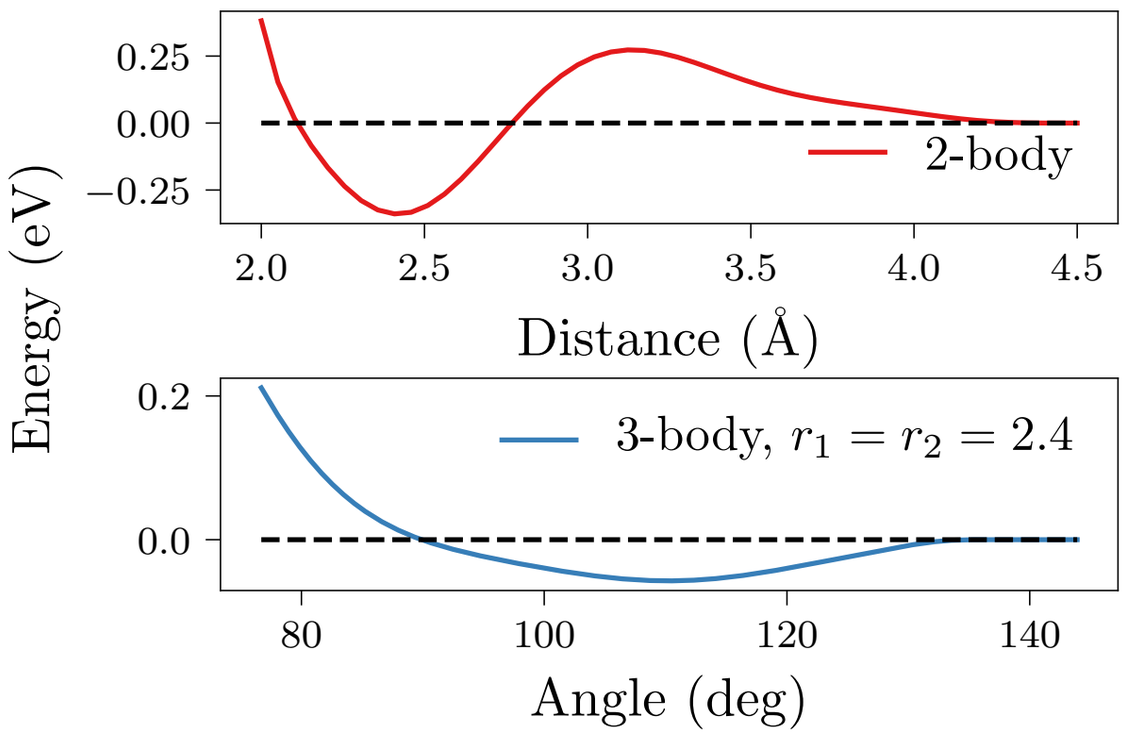
C. Zeni, K. Rossi, AG, A. Fekete, N Gaston, F Baletto, A. De Vita, *Building machine learning force fields for nanoclusters* (JCP 2018)

Examples of MFFs

Copper: *Learnt 2-body MFF*



a-Si: *Learnt 3-body MFF*



- Non-parametric force fields are always more accurate than parametric counterparts
- A confidence level can also be predicted, this can help to avoid extrapolation

Automatic force field construction

1. Obtain an initial database for your system

2. Choose the **order** of the interaction needed (2-body, 3-body, …)

3. Train a Gaussian Process regression with an **n-body kernel**

4. **Tabulate** and save the learned potential onto the effective degrees of freedom $\mathbf{q}$

5. **Interpolate** the tabulated points to yield the n-body potential energy

6. Predict the energy of a new configuration with the **learned *n*-body** as a classical potential

Automatic force field construction

1. Obtain an initial database for your system

2. Choose the **order** of the interaction needed (2-body, 3-body, …)

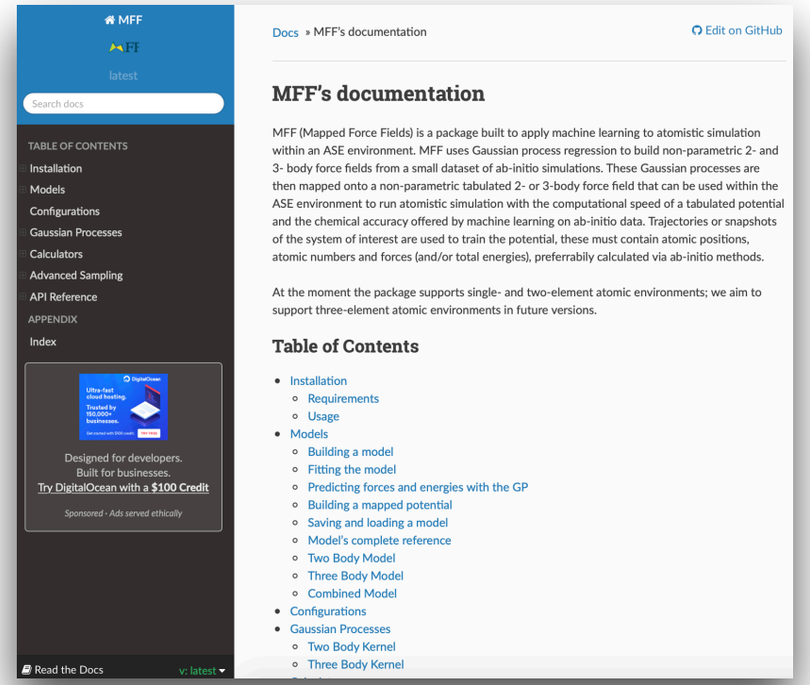
3. Train a Gaussian Process regression with an **n-body kernel**

4. **Tabulate** and save the learned potential onto the effective degrees of freedom $\mathbf{q}$

5. **Interpolate** the tabulated points to yield the n-body potential energy

6. Predict the energy of a new configuration with the **learned *n*-body** as a classical potential

In the MFF package



https://github.com/kcl-tscm/mff/blob/master/tutorials/Tutorial_nanoparticles.ipynb

Automatic force field construction

1. Obtain an initial database for your system

2. Choose the **order** of the interaction needed (2-body, 3-body, …)

3. Train a Gaussian Process regression with an **n-body kernel**

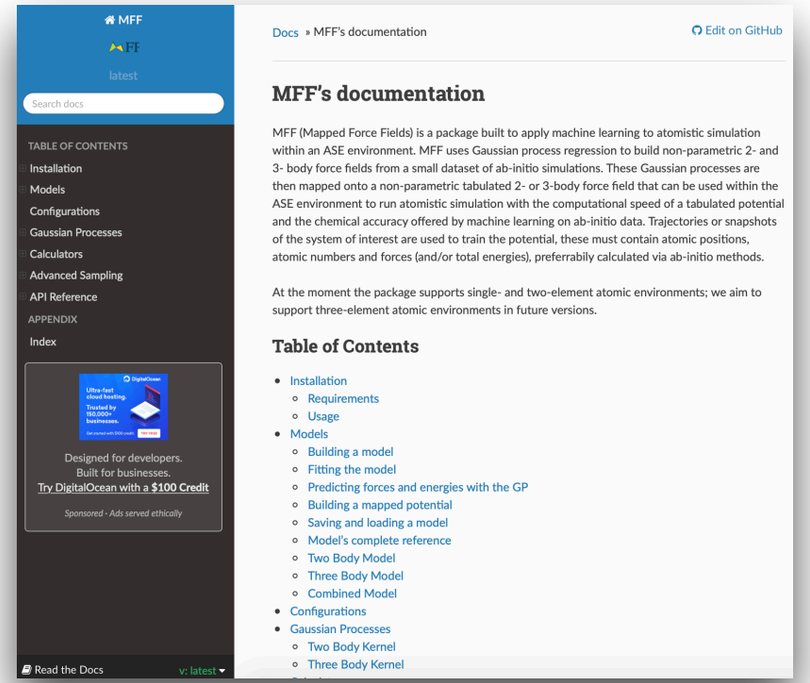
4. **Tabulate** and save the learned potential onto the effective degrees of freedom $\mathbf{q}$

5. **Interpolate** the tabulated points to yield the n-body potential energy

6. Predict the energy of a new configuration with the **learned *n*-body** as a classical potential

(7. If the GP **uncertainty** is too large, run an additional quantum calculation)

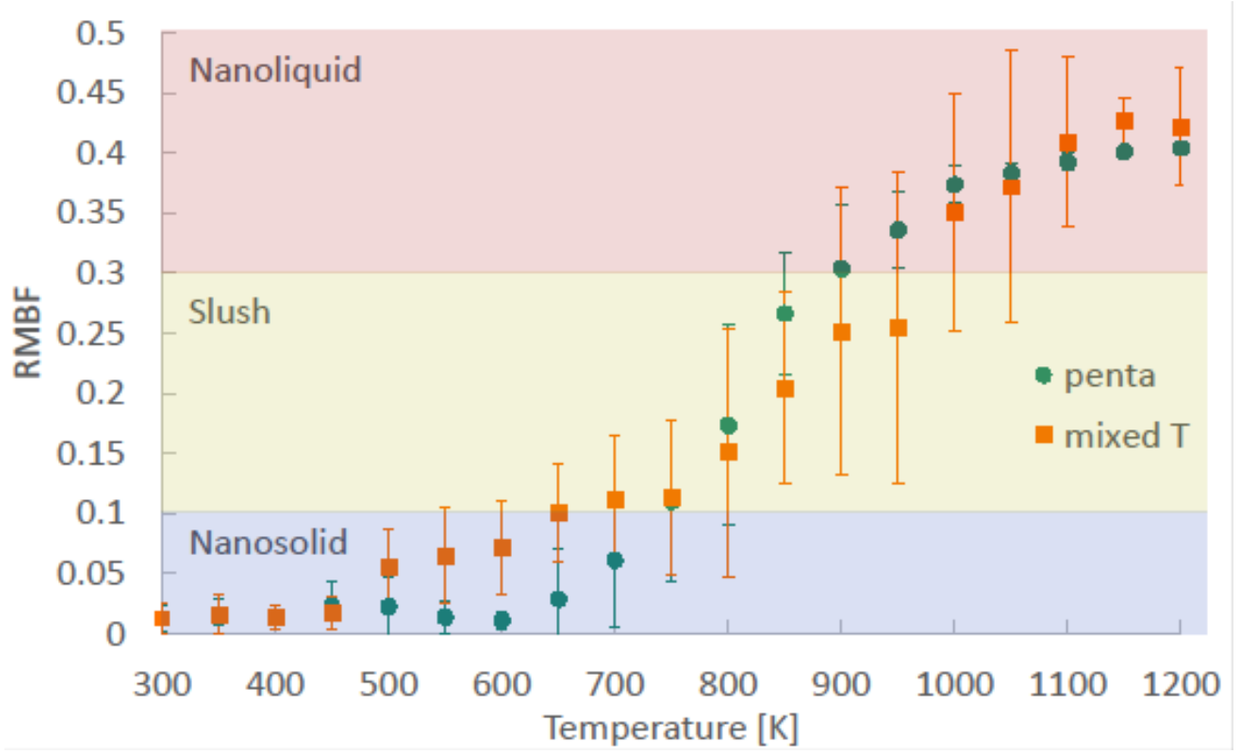
In the MFF package

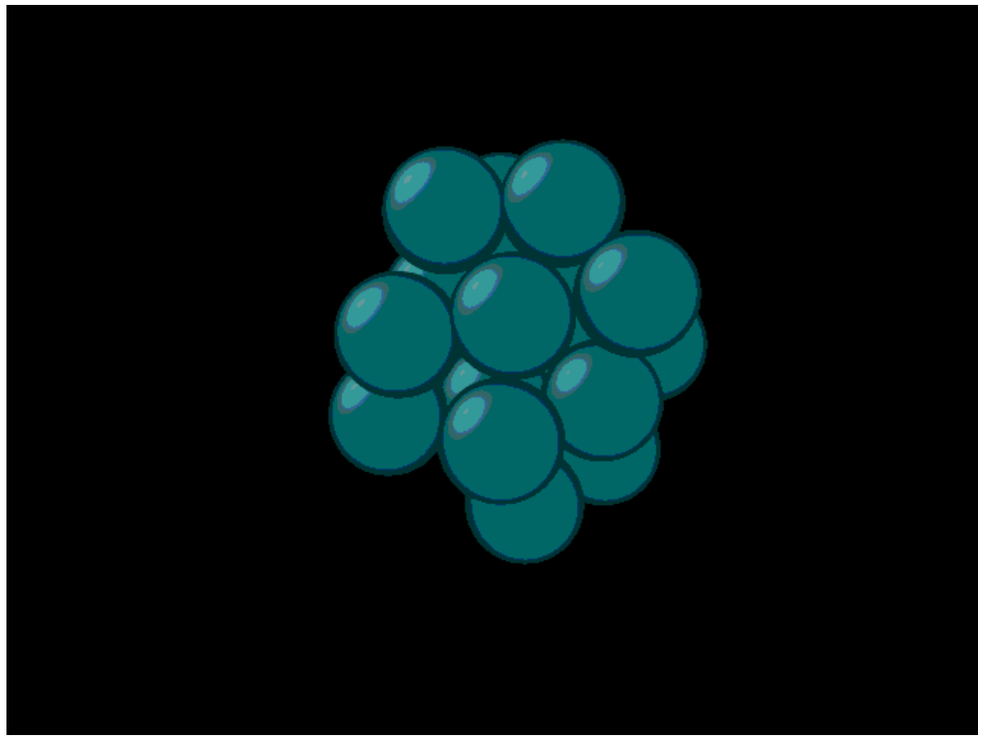


https://github.com/kcl-tscm/mff/blob/master/tutorials/Tutorial_nanoparticles.ipynb

Nanocluster MD

- GP was trained on a set of Ni 19 nanoparticles.
- 3-body MFF containing information of 1000 configurations.
- Melting of Ni19 was observed, the presence of a **slush state** confirmed.
- **61 million** MD steps were simulated in 4 days on a 24 cores.
- With DFT it would have taken 2000 years: **10^6 speed factor**.

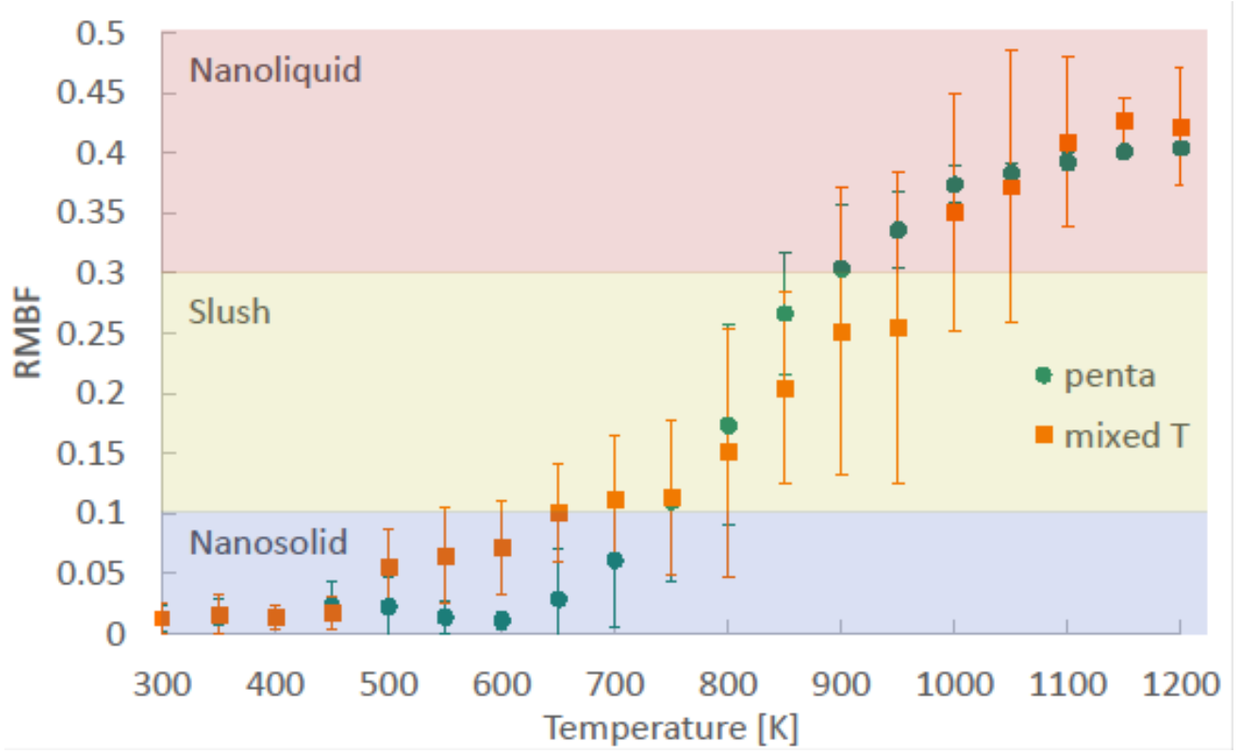


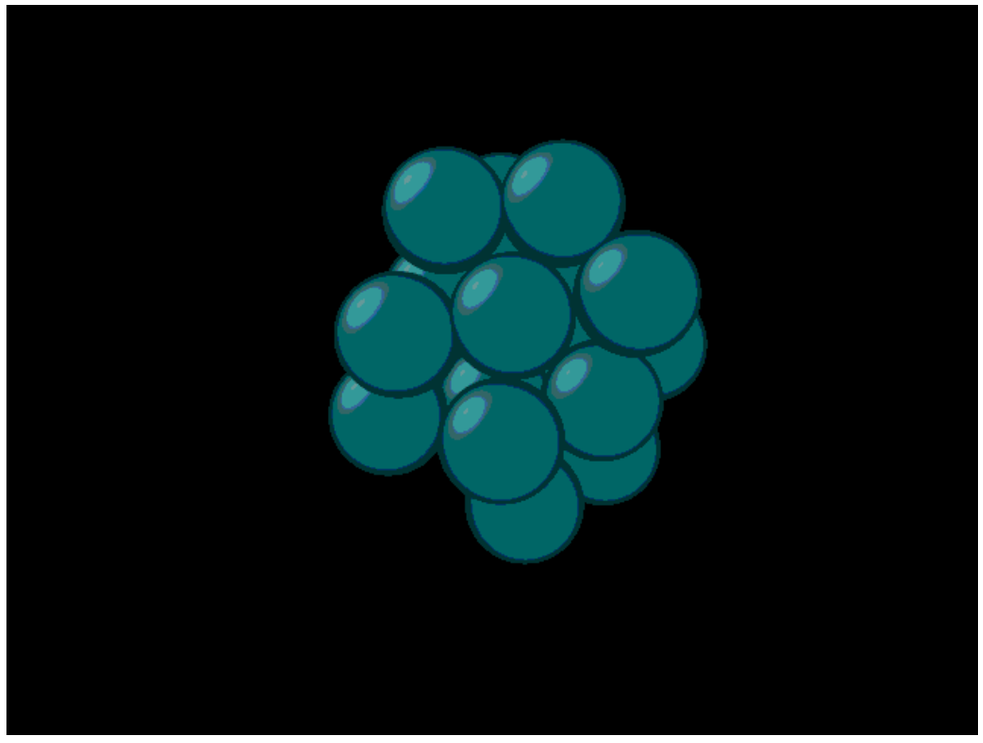


C. Zeni, K. Rossi, AG, A. Fekete, N Gaston, F Baletto, A. De Vita,
Building machine learning force fields for nanoclusters (JCP 2018)

Nanocluster MD

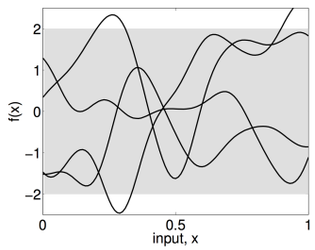
- GP was trained on a set of Ni 19 nanoparticles.
- 3-body MFF containing information of 1000 configurations.
- Melting of Ni19 was observed, the presence of a **slush state** confirmed.
- **61 million** MD steps were simulated in 4 days on a 24 cores.
- With DFT it would have taken 2000 years: **10^6 speed factor**.



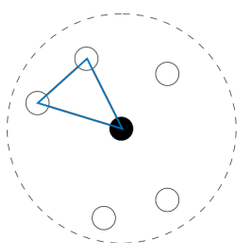


C. Zeni, K. Rossi, AG, A. Fekete, N Gaston, F Baletto, A. De Vita,
Building machine learning force fields for nanoclusters (JCP 2018)

Conclusions



- For robustness and interpretability, we need to be able to **control** the **complexity** of learning algorithms



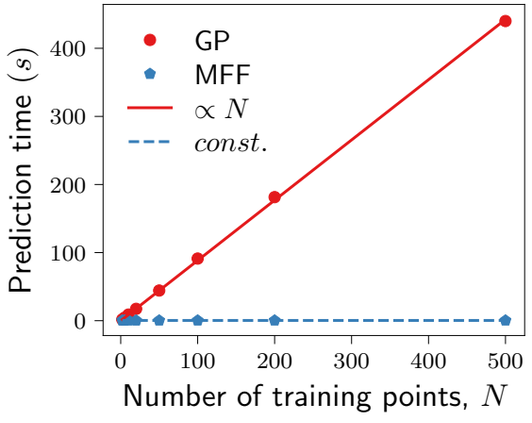
- A way to do so is to include all physical symmetries and appropriately **restrict** the modelled **interaction order**

$$k_n^S(\rho, \rho') = \int_{O(3)} d\mathscr{R} k_n(\rho, \mathscr{R}\rho')$$

- This can be done within GP regression by using fully symmetric ***n*-body kernels**

$$\ln p(\boldsymbol{\varepsilon} \mid \boldsymbol{\rho}, \mathscr{M}_n)$$

- **Low order models** are often found to be sufficiently accurate and should hence be selected for their better extrapolation properties



- Furthermore their predict ions can be **mapped** onto explicit bases, giving rise to fast and accurate MFFs: **non-parametric force fields**

People we need to thank

Prof. Alessandro De Vita
King's College London,
University of Trieste

Dr. Ádám Fekete
King's College London

Dr. Kevin Rossi
EPFL

Questions?