Interatomic Potentials Enabled by Machine Learning

Daniel Schwalbe-Koda

Lawrence Livermore National Laboratory
https://dskoda.com, <dskoda@llnl.gov>

Lecture: 4th IKZ-FAIRmat Winter School
Jan. 24, 2023

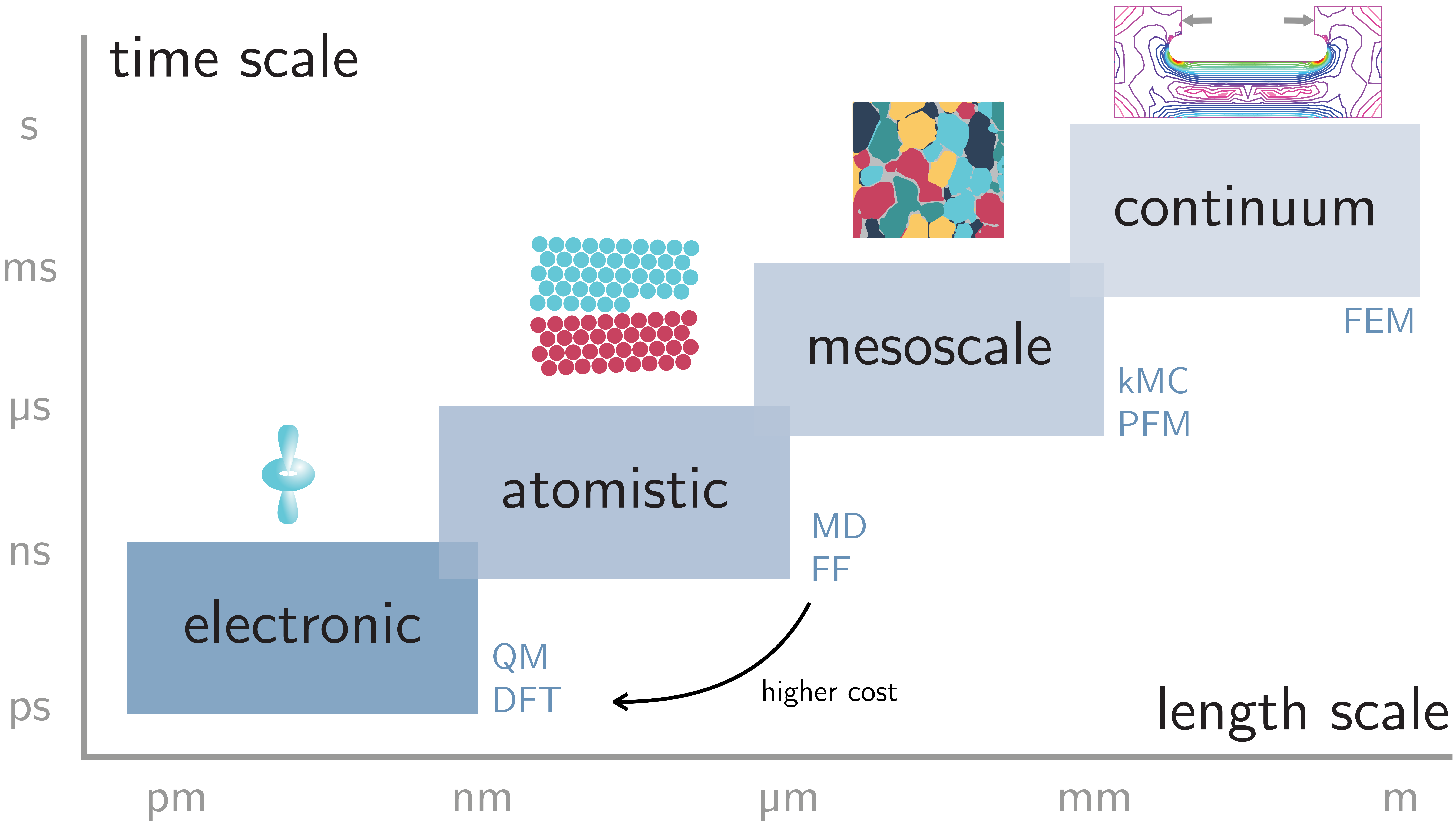
Image created with DALL-E 2

LLNL-PRES-843937
Prepared by LLNL under Contract DE-AC52-07NA27344. LDRD # 22-ERD-055

1. Introduction

Atomistic simulations and machine learning

Materials simulations span multiple length and time scales



time scale

s

ms

µs

ns

ps

electronic

QM
DFT

atomistic

MD
FF

mesoscale

kMC
PFM

continuum

FEM

higher cost

length scale

pm

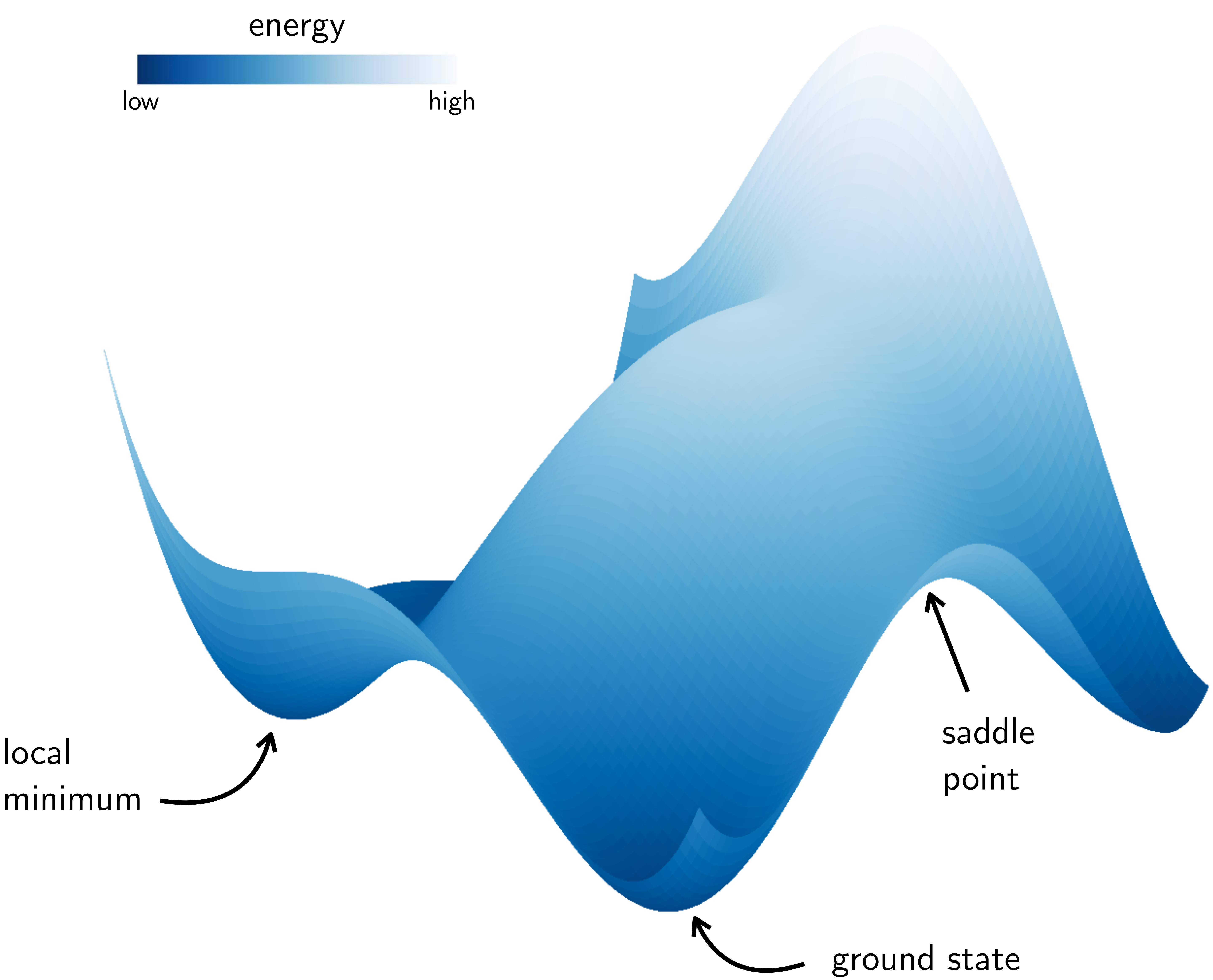
nm

µm

mm

m

In the atomistic world, several properties of interest are obtained from a **potential energy surface** (PES)



From a PES, we can derive: geometries, reaction energies, mechanical properties, dynamical behavior...

A PES is illustrated with respect to system coordinates (positions, distances, angles etc.)

However, in principle, the energy from atomistic systems can be computed from the atomic numbers and coordinates:

$$E = f(Z_i, \mathbf{r}_i)$$

(this is valid even in DFT or QC)

Sometimes, the properties of interest can be obtained from simpler models

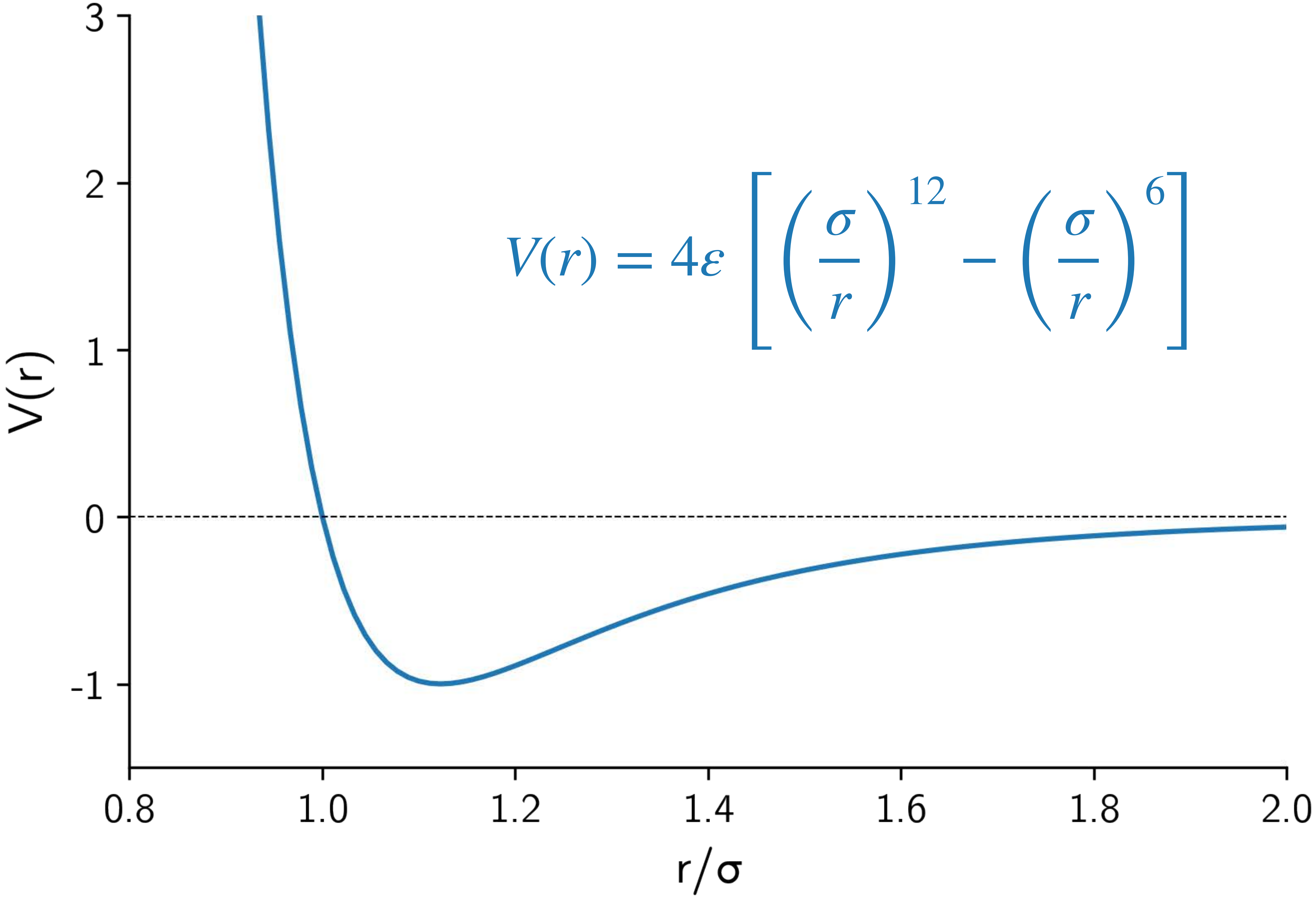
Instead of using quantum mechanics or DFT to model a PES, we can use simpler models to describe interatomic interactions.

This reduces the computational cost: analytical energy functions are **much** faster than quantum mechanics calculations (several orders of magnitude).

For example, a pairwise potential is a simple approximation of the interaction energy between atoms:

$$E_{ij} = V(\mathbf{r}_i - \mathbf{r}_j)$$

Lennard-Jones potential: a simple example of pair potential



$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

For example, potentials for noble gas usually rely on LJ models:

J. Chem. Phys., Vol. 119, No. 15, 15 October 2003

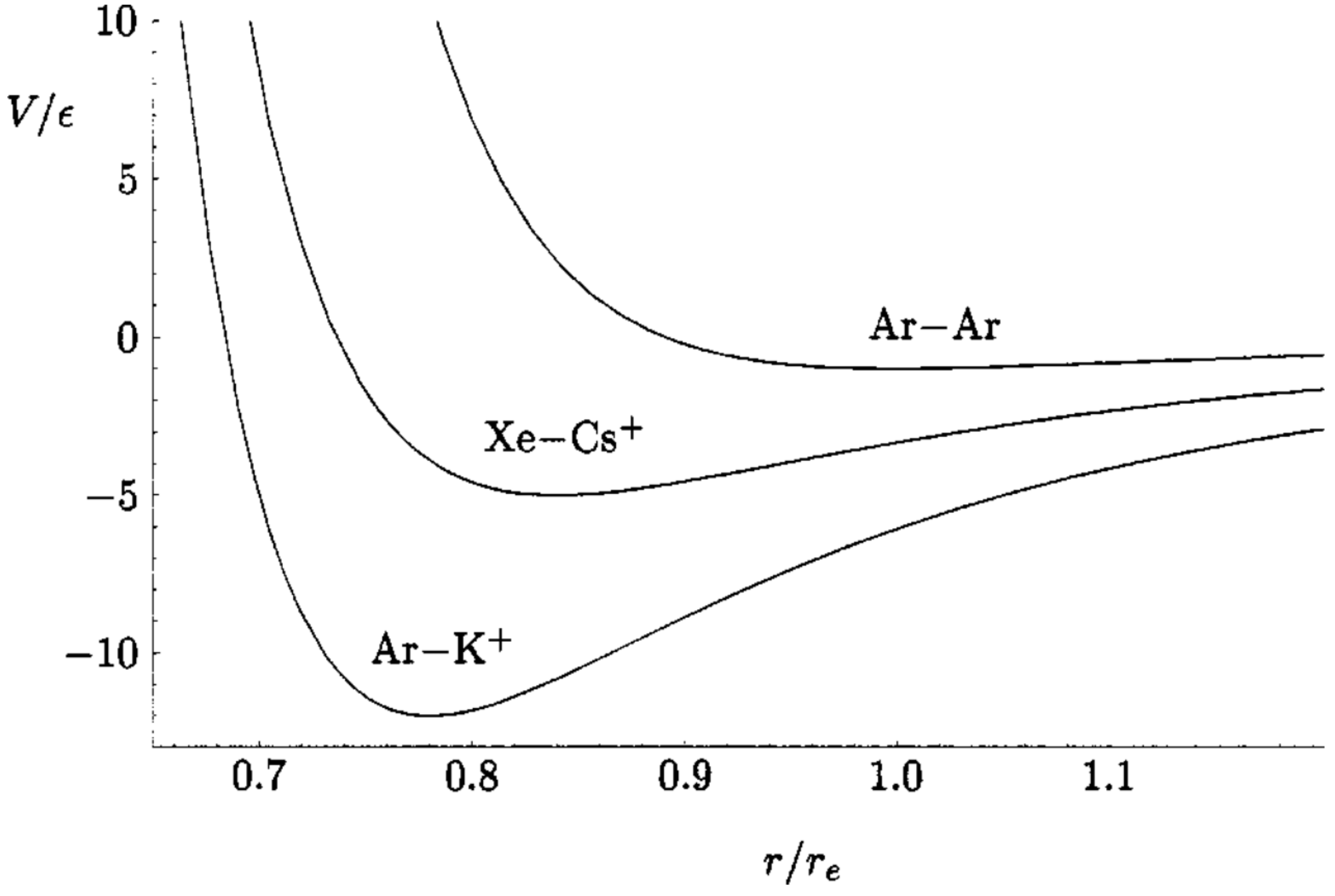
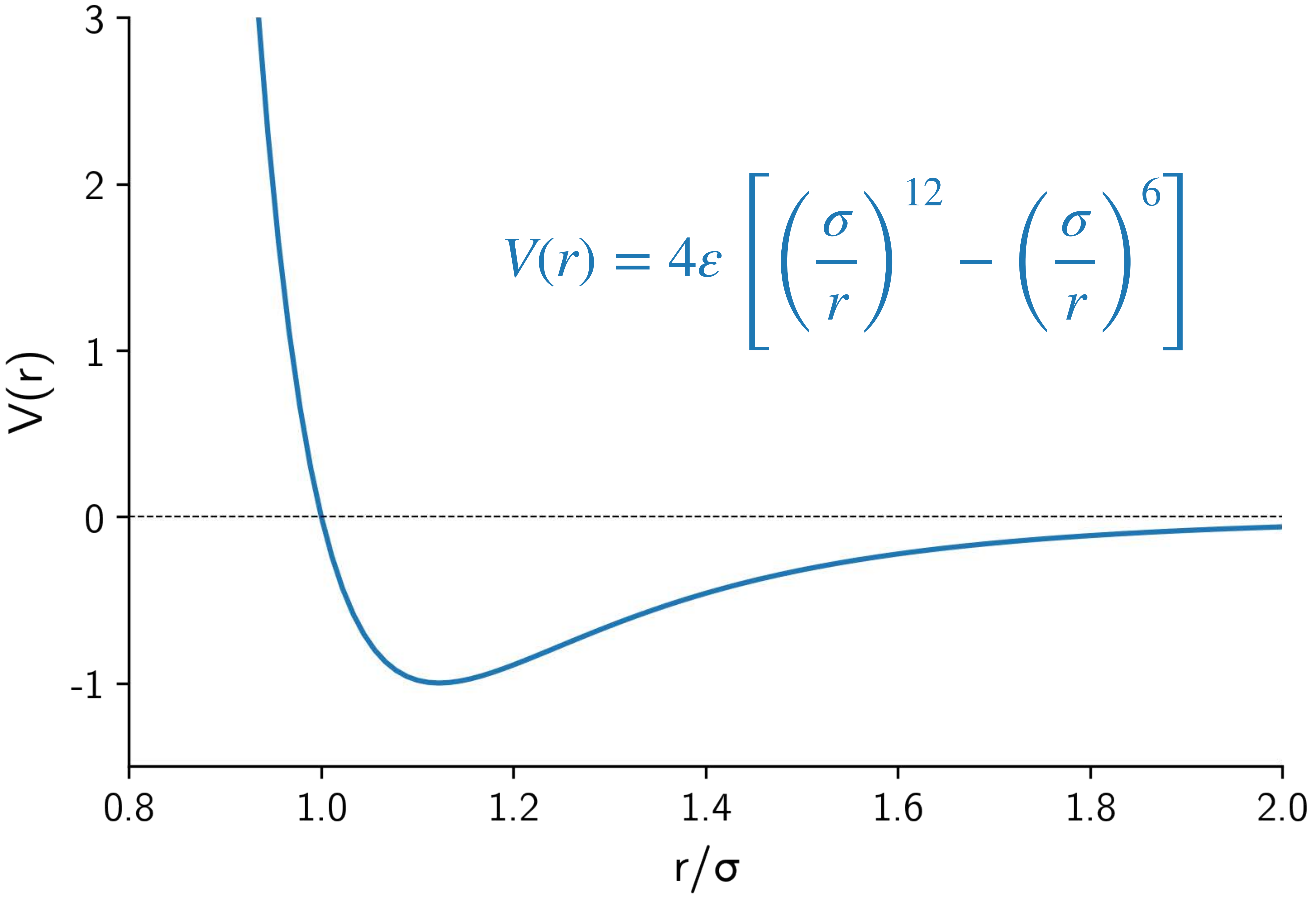


FIG. 1. Comparison of the Lennard-Jones potential used for rare gas interactions (labeled Ar–Ar) with the Mason–Schamp functions employed for Ar–K$^+$ and Xe–Cs$^+$. The appropriate Lennard-Jones pair well depth and pair equilibrium separation are taken as the units of energy and distance, respectively.

J. Hernandez-Rojas, D. Wales. *JCP* **119** (15), 7800 (2003)

Lennard-Jones potential: a simple example of pair potential



$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

However, for more complicated systems, there are several options of potentials such as:

Lennard-Jones potential

$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

Morse potential

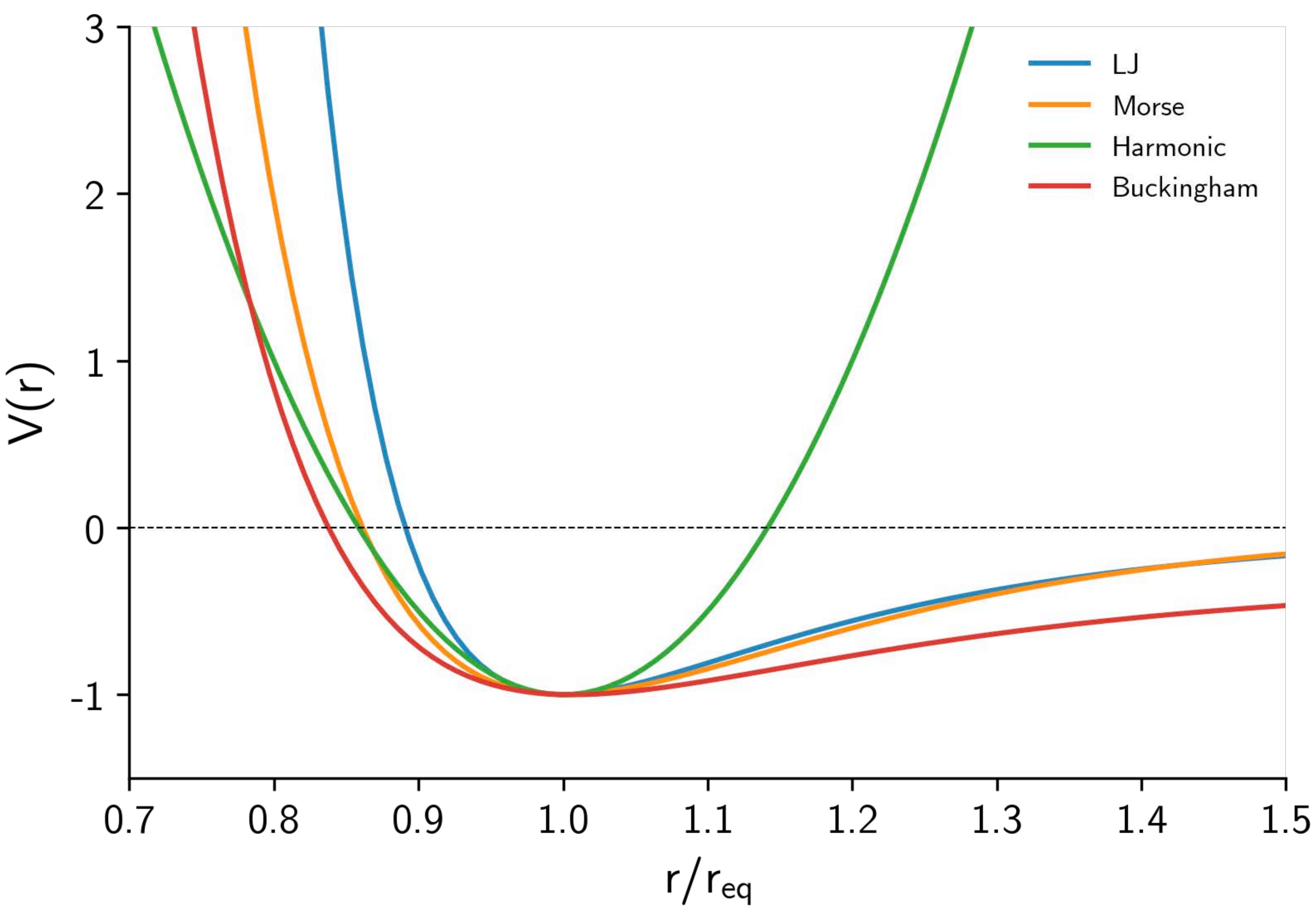
$$V(r) = A\left[1 - e^{-a(r-r_{\mathrm{eq}})}\right]$$

Harmonic potential

$$V(r) = A(r - r_{\mathrm{eq}})^2$$

Buckingham-Coulomb potential

$$V(r) = Ae^{-Br} - \frac{C}{r^6} + \frac{q_1 q_2}{4\pi\varepsilon_0 r}$$



Usually, interatomic potentials involve a combination of these terms

e.g., the CHARMM22 force field:

$$V(r) = \sum_{\text{bonds}} k_b(b - b_0)^2 + \sum_{\text{angles}} k_\theta(\theta - \theta_0)^2 + \sum_{\text{improp.}} k_\omega(\omega - \omega_0)^2 + \sum_{\text{Urey--Bradley}} k_u(u - u_0)^2$$

$$+ \sum_{\text{dihedrals}} k_\phi \left[1 + \cos(n\phi - \delta)\right] + \sum_{\text{nonbond.}} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6} \right] + \sum_{\text{nonbond.}} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

Classical force fields are usually fit to structural, vibrational, and other energy-based models from *ab initio* calculations.

The problem is: how to choose the functional forms and parameters?

From the CHARMM22 paper:

Adjustment of the parameters was performed manually, although in certain cases (e.g., for proline) automated procedures were employed. We have found that automated procedures must be used with great care owing to the extensive nature of parameter space, correlation among the parameters, and their underdetermined nature. An automated least-squares procedure often leads to a combination of “unphysical” parameters that reproduce the input data. More meaningful parameter values, which have a wider range of applicability, were obtained manually with “reasonable” parameter ranges for the optimization in the iterative refinement procedure described above. 😓

M. Karplus et al. *J. Phys. Chem. B* **102** (18), 3586 (1998)

…and there's one more problem: cost vs. accuracy trade-off

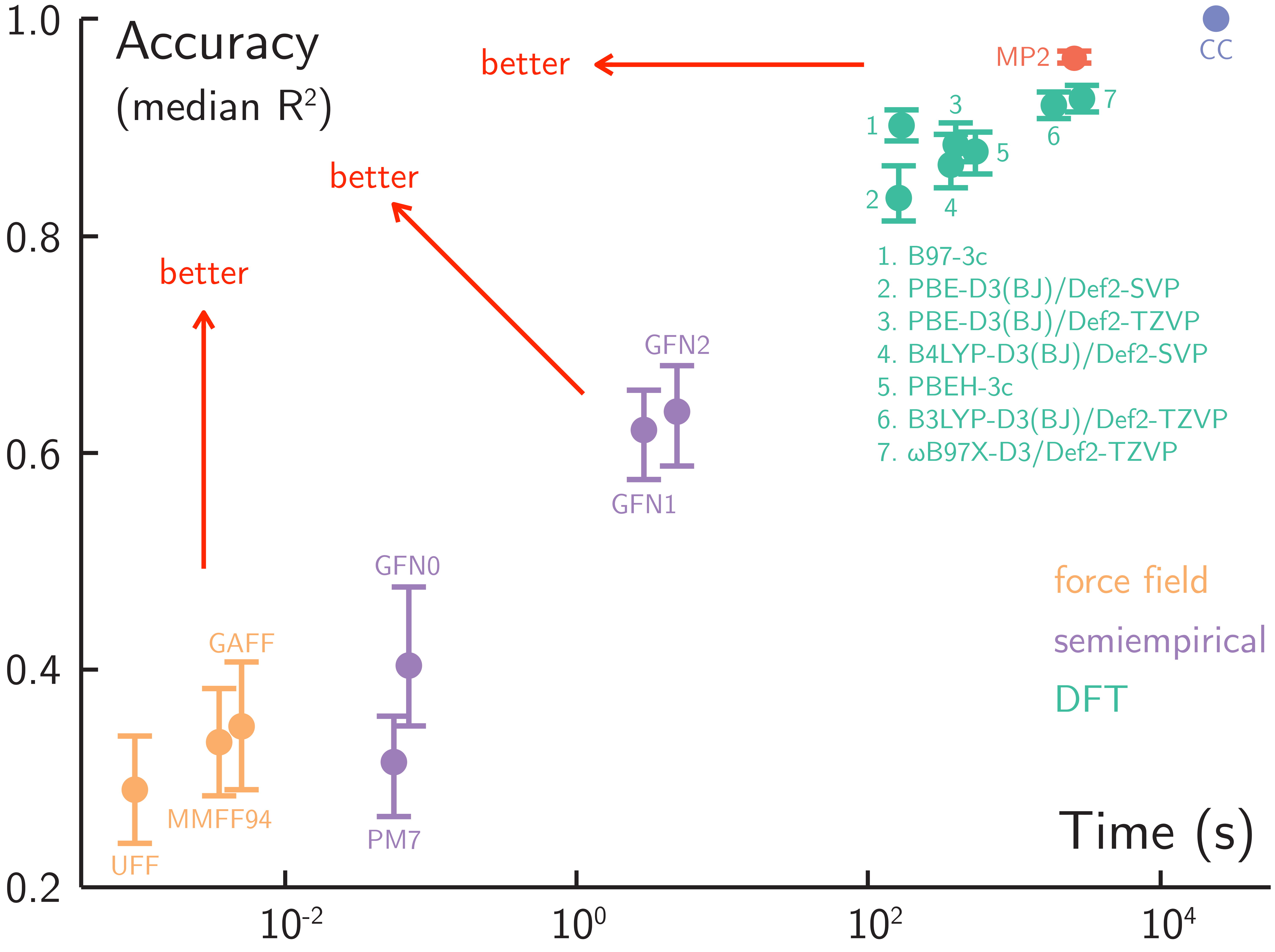


Figure adapted from:

Z. Qiao et al. *J. Chem. Phys.* **153**, 124111 (2020)

(Accuracy computed by Qiao et al. with respect to the Hutchinson conformer benchmark)

Accurate methods (usually QM- or DFT-based) are computationally expensive to compute

Force fields are simple to compute, but their accuracy is low compared to coupled-cluster (or even DFT) methods.

The second problem is: how can we obtain accurate, yet fast potentials?

Enter machine learning (ML)

The use of ML has become a trend to address issues of automation, pattern recognition, and cost-accuracy trade-off

NUMBER of AI PUBLICATIONS by FIELD of STUDY (EXCLUDING OTHER AI), 2010–21

Source: Center for Security and Emerging Technology, 2021 | Chart: 2022 AI Index Report

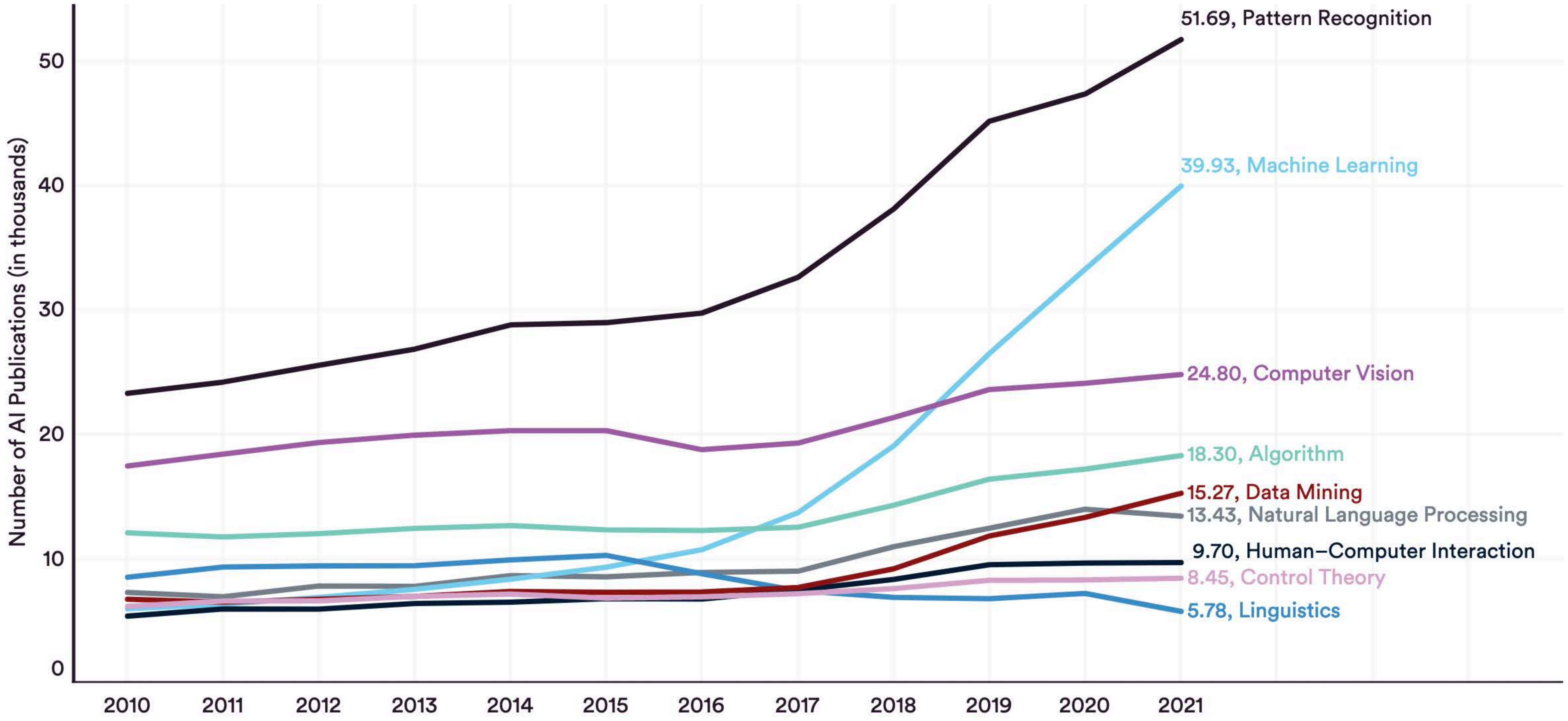
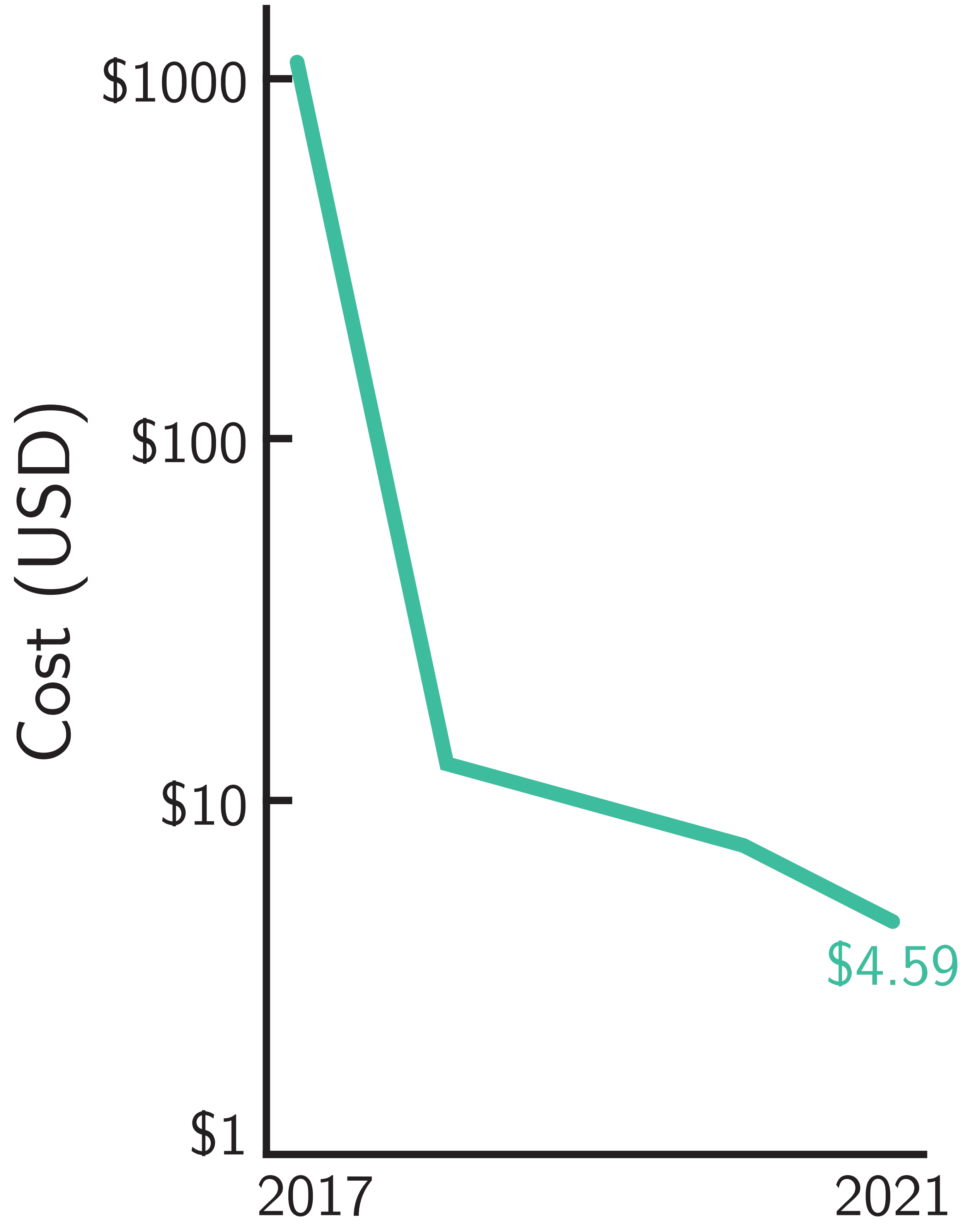


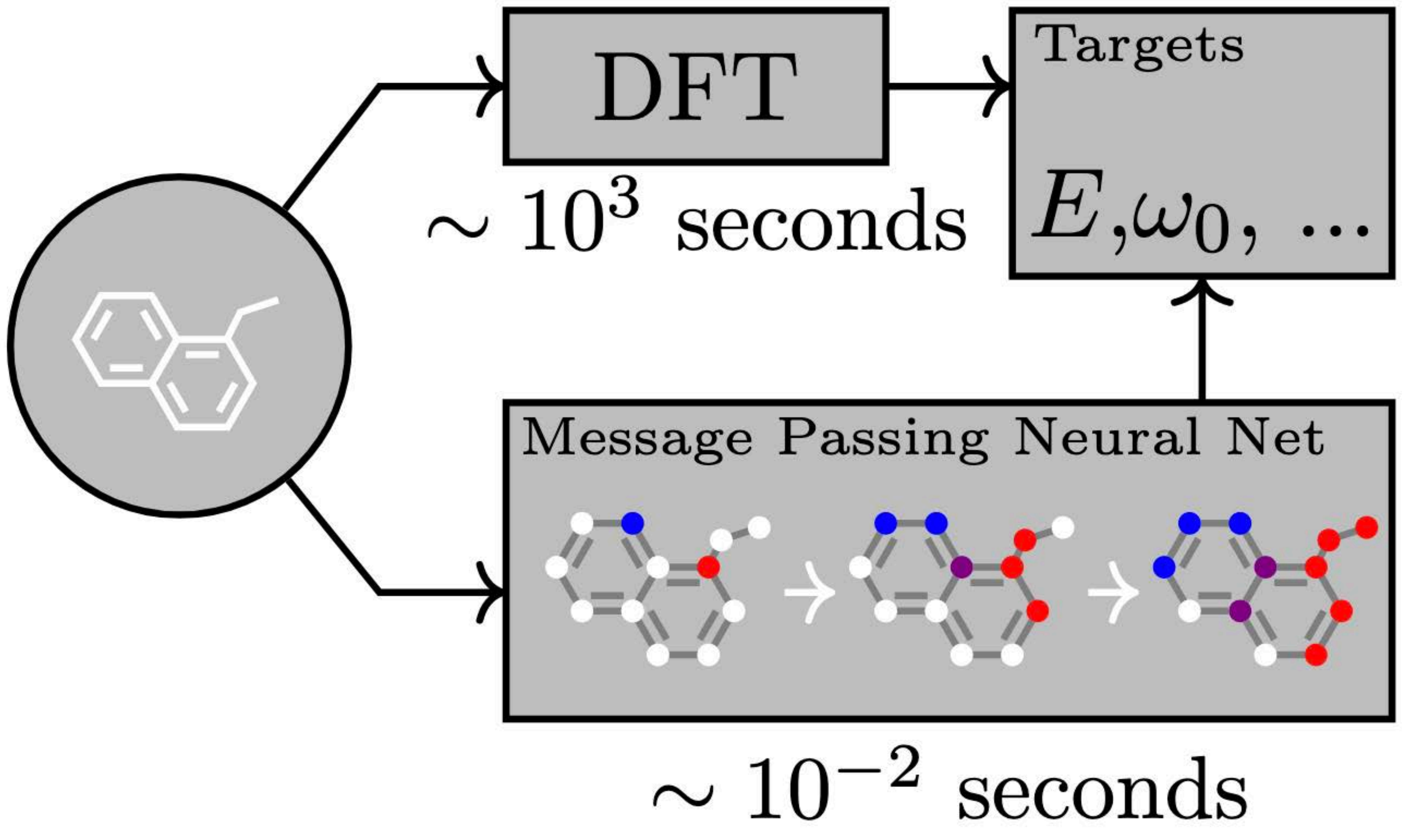
Figure 1.1.3

IMAGENET: training cost (93% acc.)

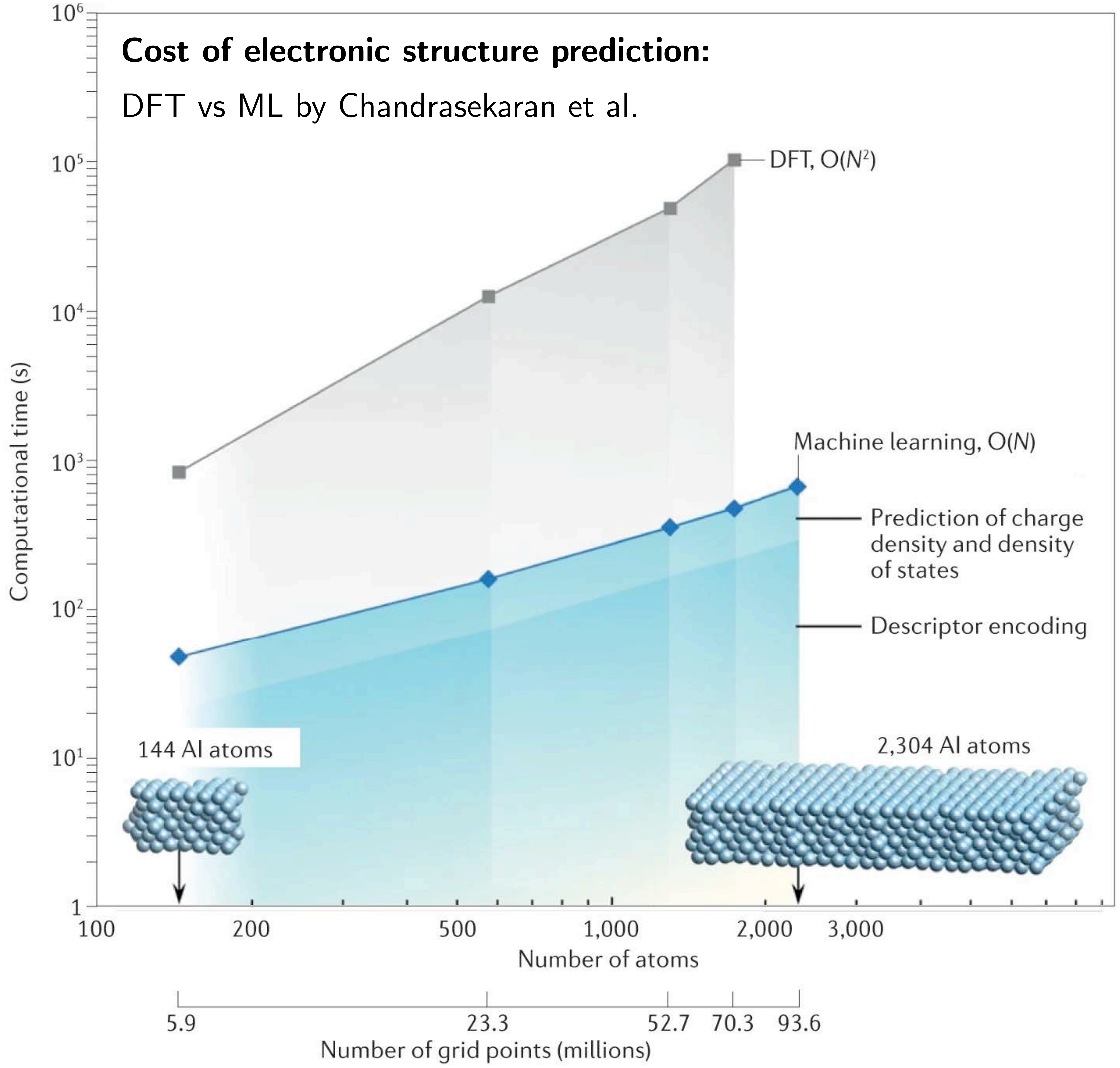


D. Zhang et al. "The AI Index 2022 Annual Report," AI Index Steering Committee, Stanford Institute for Human-Centered AI, Stanford University, March 2022.

ML has been helping reduce the cost of calculations for materials and chemical systems



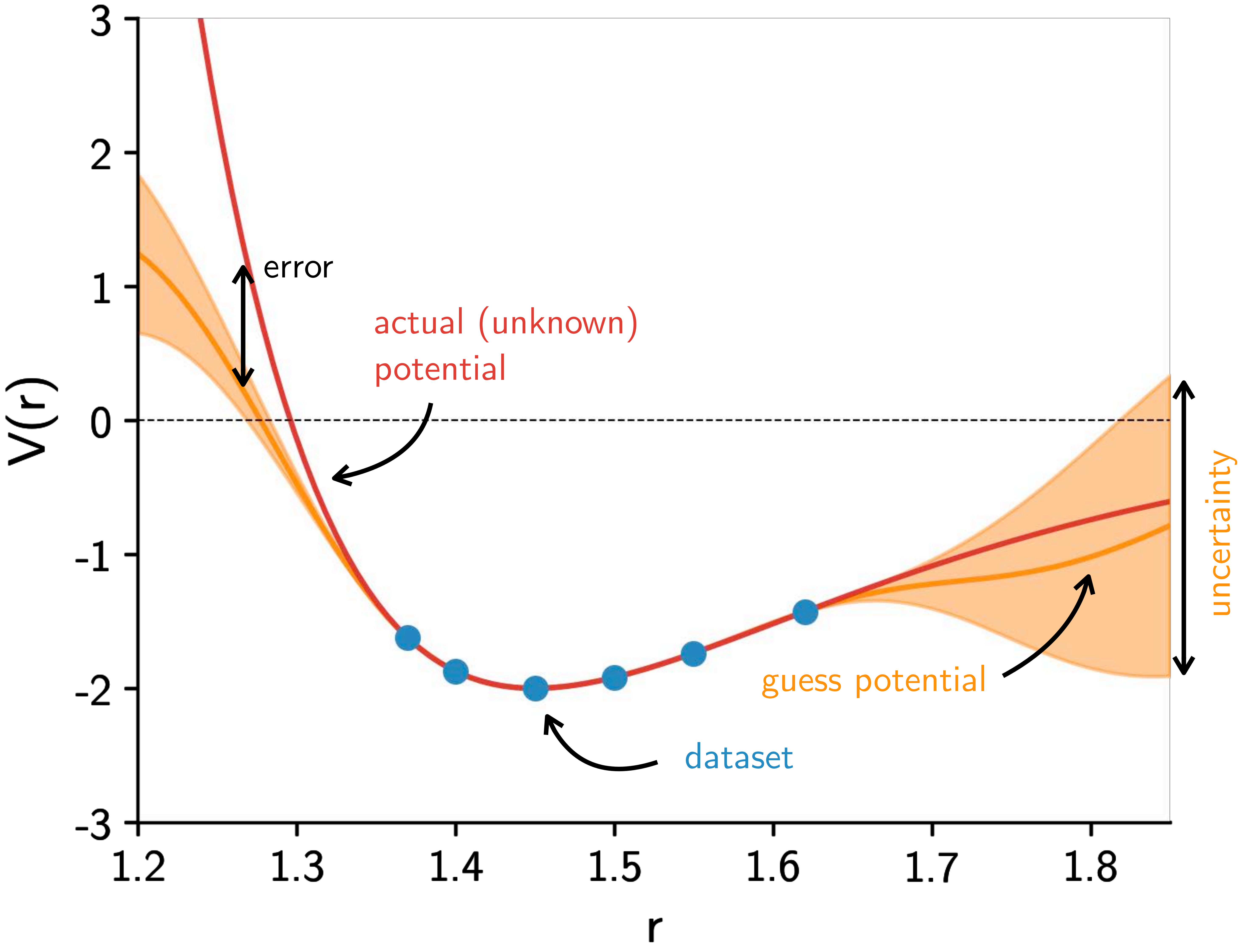
J. Gilmer et al. *arXiv*:1704.01212 (2017)


Cost of electronic structure prediction:

DFT vs ML by Chandrasekaran et al.

A. Chandrasekaran et al. npj Comp. Mater. **5**, 22 (2019)
Adapted from N. Fedik et al. *Nat. Rev. Chem.* **6**, 653 (2022)

Particularly in interatomic potentials, ML helps in fitting to datasets

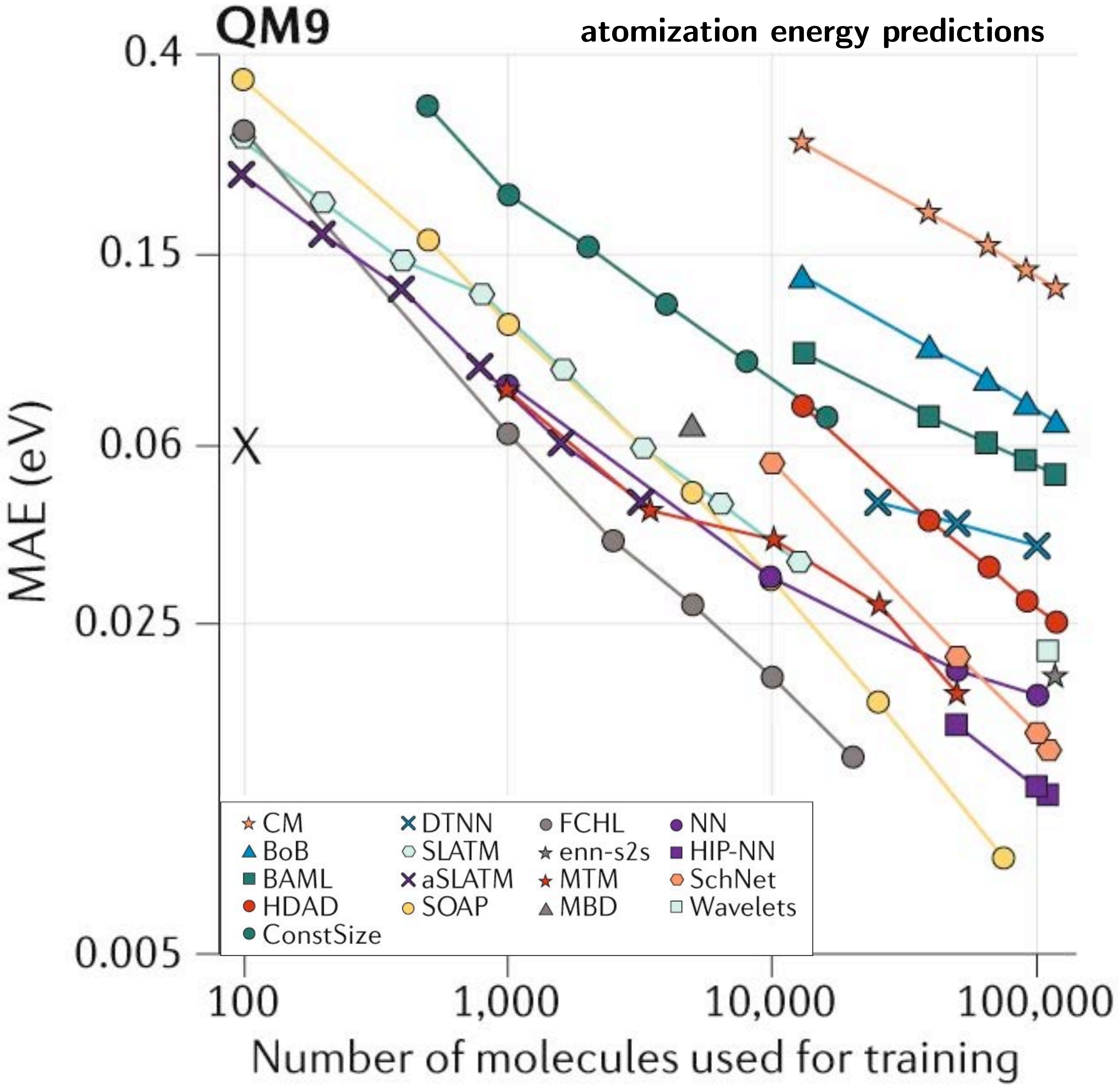


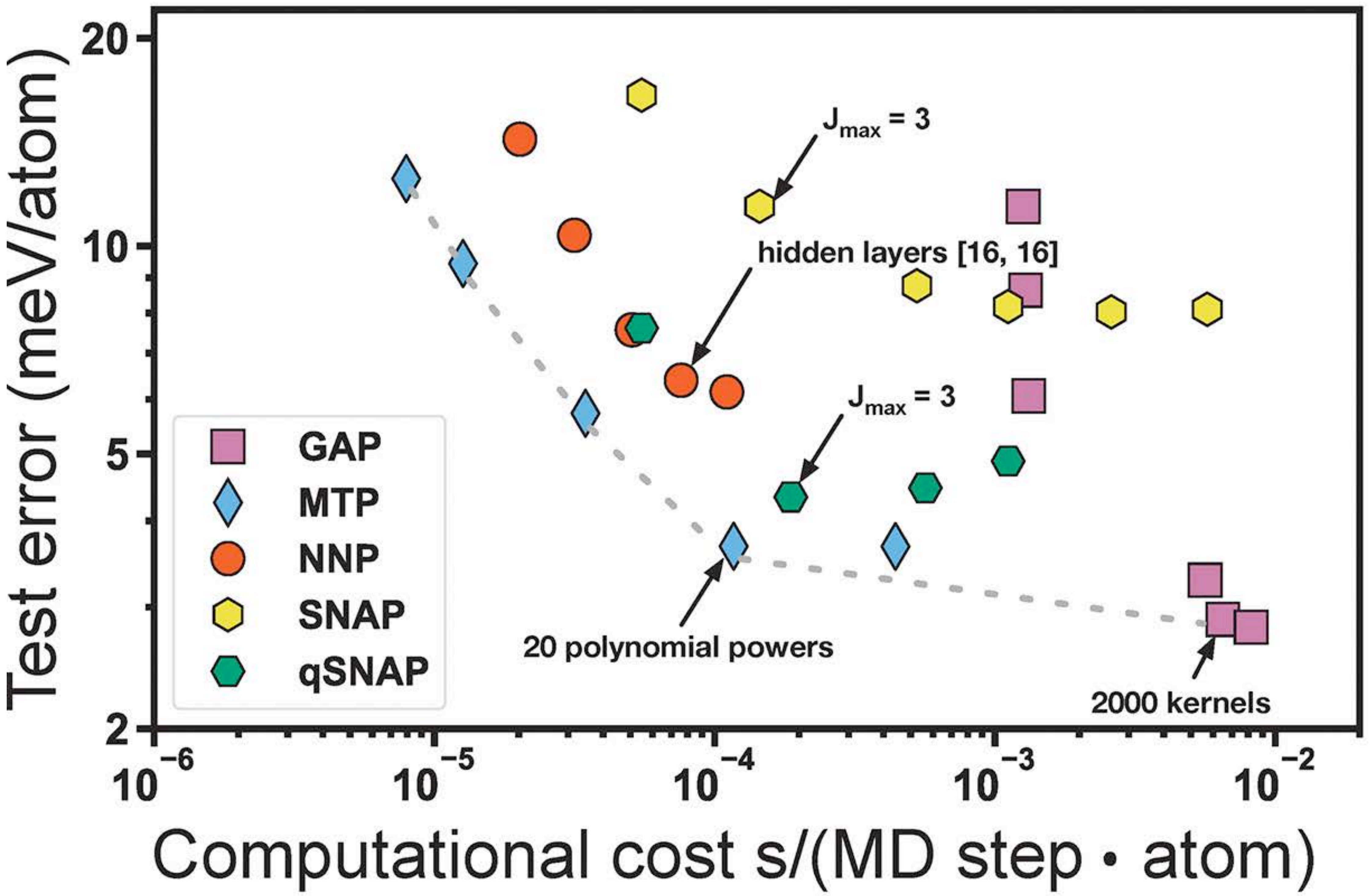
ML methods enable a fitting to guess potentials based on a given dataset of interest.

When implemented, this approach can automate the process of finding functional forms that fit to the data and bypasses functional forms that can be less accurate.

But which ML methods should we use to implement force fields?

There are many ML methods and implementations possible...

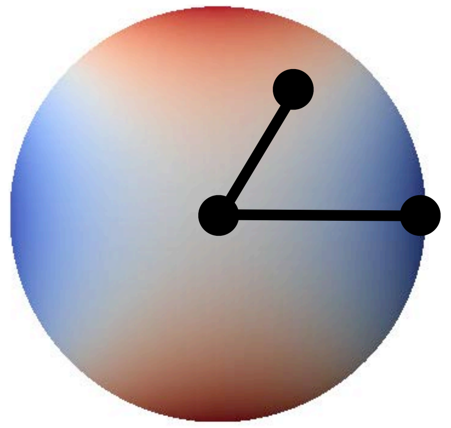


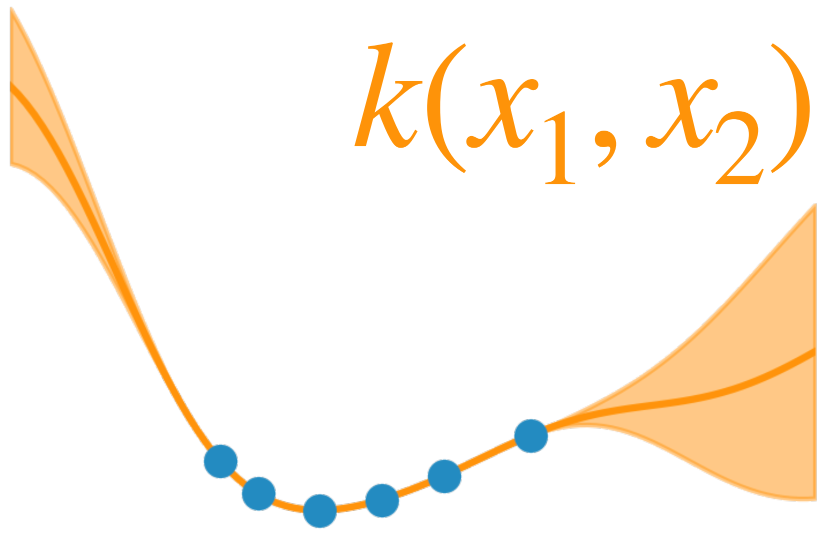


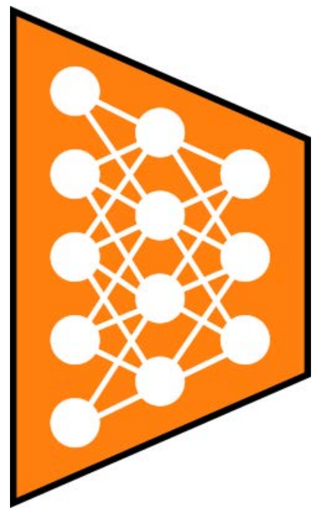
O. A. von Lilienfeld et al. *Nat. Rev. Chem.* **4**, 347 (2020)

Y. Zuo et al. *J. Phys. Chem. A* **124** (4), 731 (2020)

...but the most popular ones are:



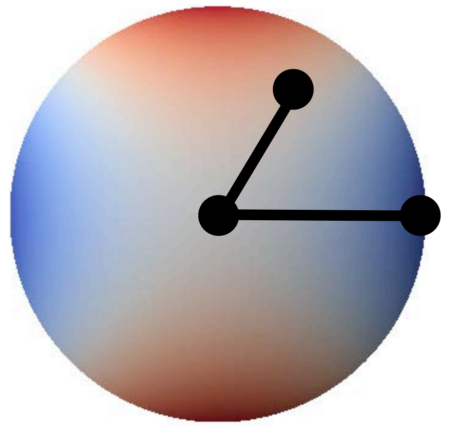




	Linear Methods	**Kernel/Gaussian Process Regression**	**Neural Networks (NNs)**
ℹ️	Polynomial on many-body terms	Computes an explicit similarity between points	“Universal approximator” with non-linear mappings
😁	Simple and fast	Fewer data points (+uncertainty for GPR)	High accuracy
😞	Relies crafting a representation for the inputs	O(N^3) complexity for training for GPR	Large number of trainable parameters

Some examples in the literature:

Linear Methods

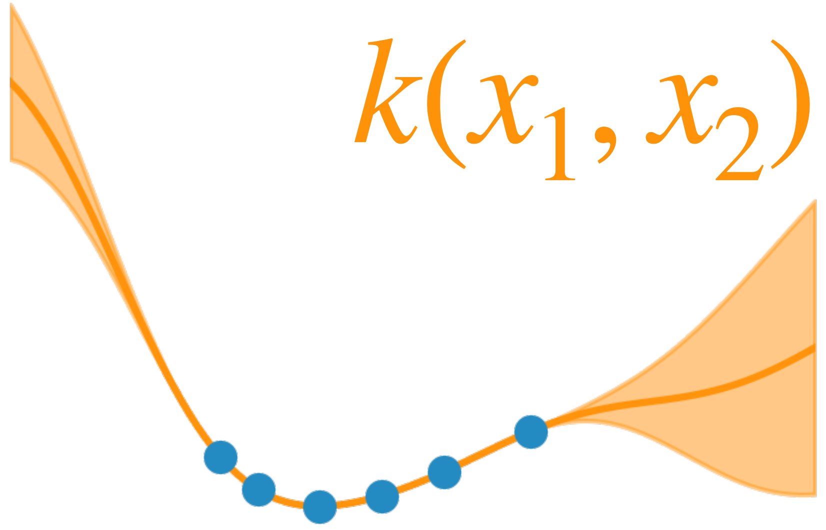


SNAP (Thompson et al.)

MTP (Shapeev)

ACE (Drautz, Kovács et al.)

Kernel/Gaussian Process Regression



GPR:

GAP (Bartok et al.)

MLOTF (Li et al.)

FLARE (Vandermause et al.)

Other kernels with:

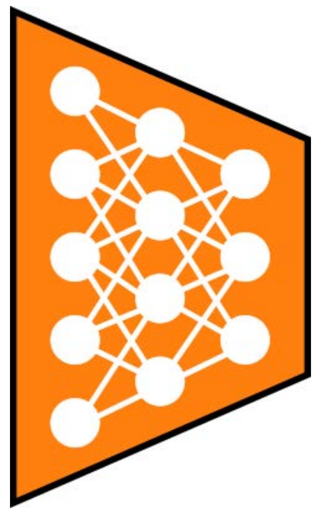
sGDML (Chmiela et al.)

FCHL repres. (Faber et al.)

Coulomb matrices (Rupp et al.)

this lecture:

Neural Networks (NNs)



Behler-Parrinello

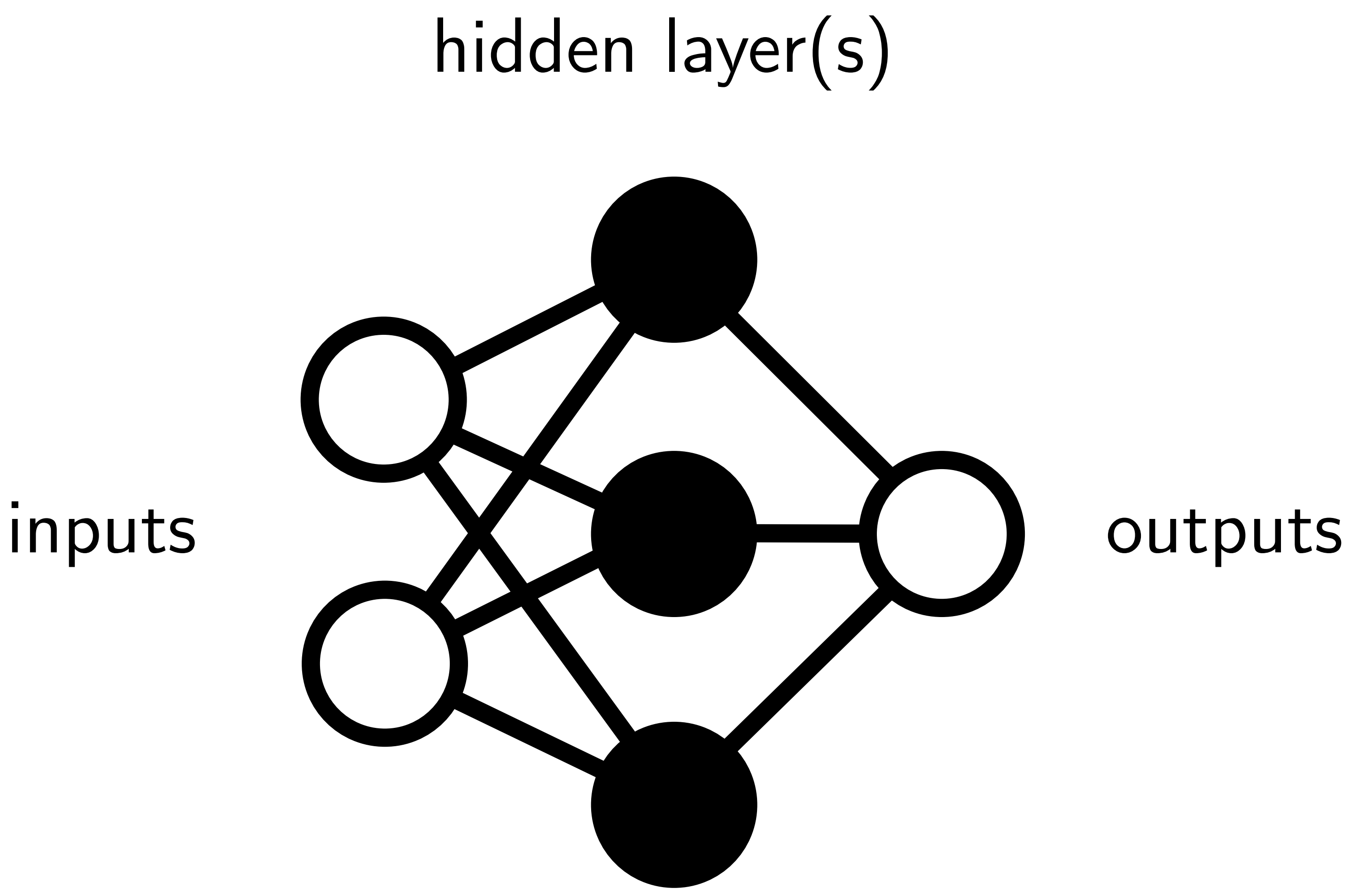
Representation + NN (DeepMD, ANI etc.)

Deep learning-based NNFF (SchNet etc.)

Deep learning + equivariance (NequIP, PaiNN etc.)

Deep learning + many-body expansion (MACE etc.)

One-slide neural network refresher



non-linearity

$$f(\mathbf{X}) = \sigma\,(W\mathbf{X} + \mathbf{b})$$

input weight matrix bias

$$\hat{y} = f(\mathbf{X})$$

estimated output

Training of neural networks requires setting a loss function (e.g., for regressors, no regularization):

$$\mathscr{L} = \mathbb{E}_{\mathbf{X}\sim P(X)}\left[\|\hat{y} - y\|^2\right]$$

which updates the weights using the backpropagation algorithm and gradient descent:

loss gradient

$$w_{ij}^{(n+1)} = w_{ij}^{(n)} - \alpha\frac{\partial \mathscr{L}}{\partial w_{ij}}$$

weight at iteration (n + 1)

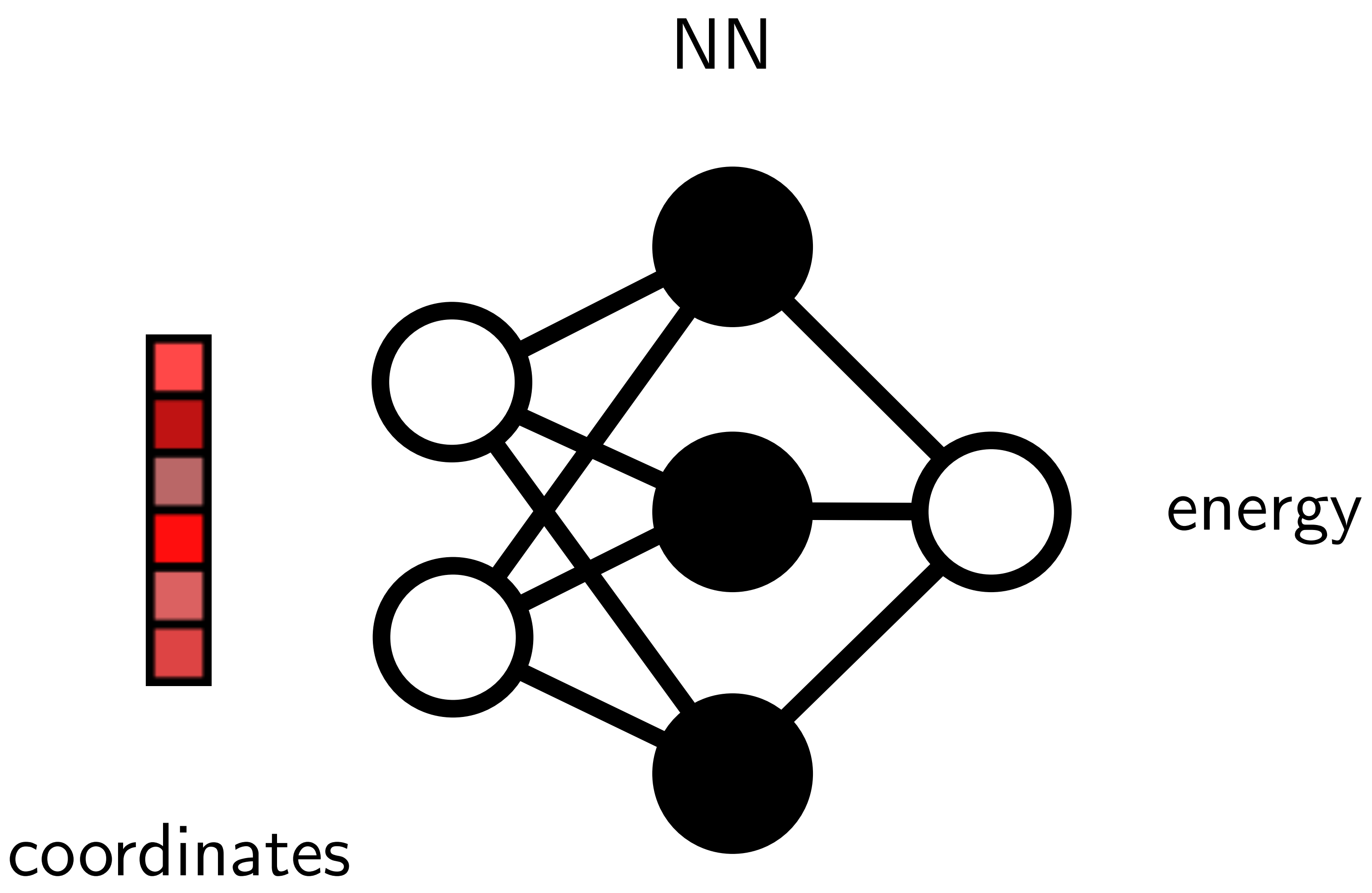
learning rate

Mini Tutorial: NNs are "universal approximators"

2. NN Interatomic Potentials

Fitting PESes with neural networks

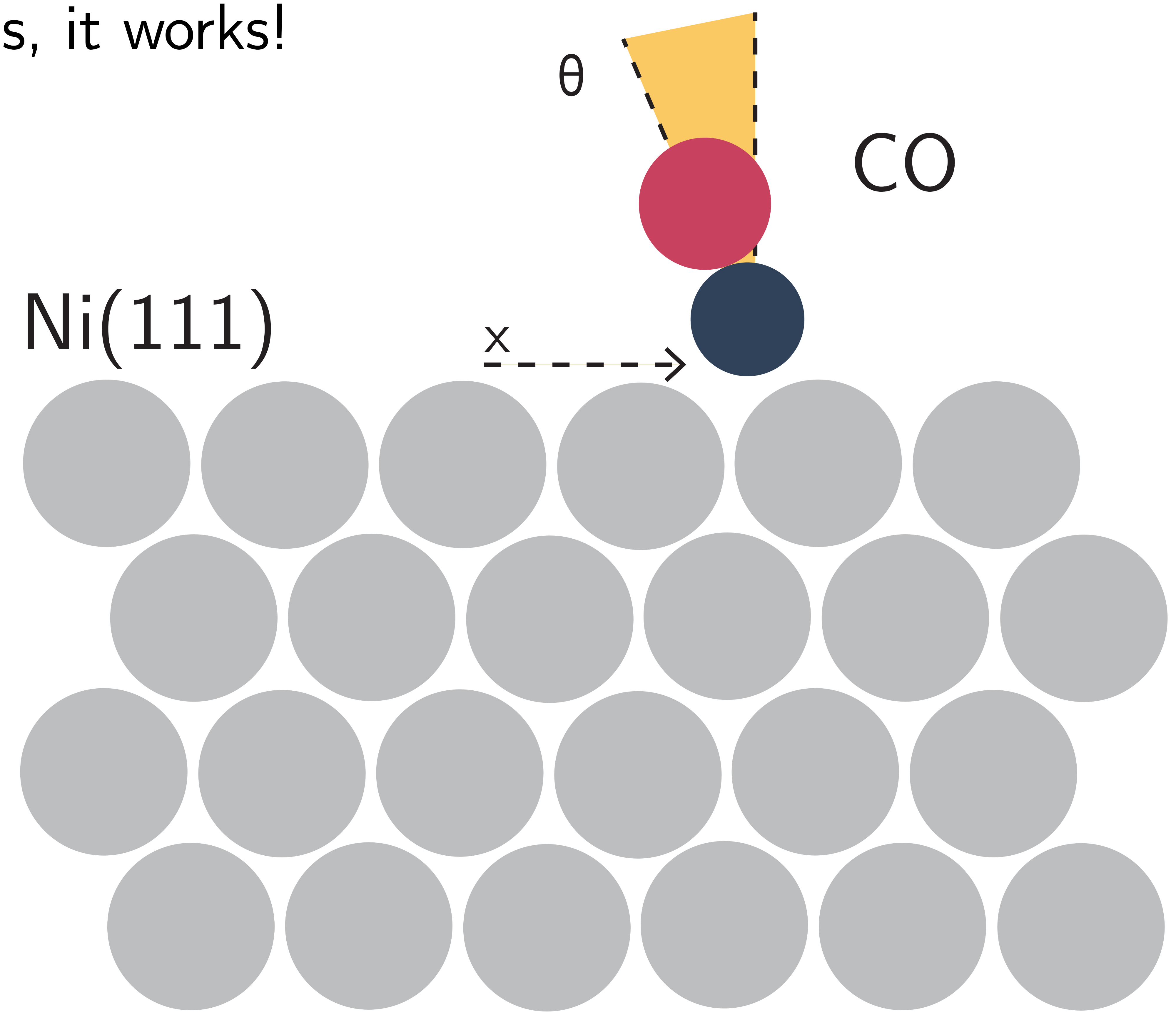
Simplest possible approach in fitting a PES using a neural network



The NN is trained to map input coordinates to energy for a given dataset.

Does this work?

Yes, it works!



T. Blank et al. *J. Chem. Phys.* **103**, 4129 (1995)

What are the advantages/ disadvantages of this approach?

What are the advantages/disadvantages?

✅ Advantages

Fast

No need for extra information

Easy interpretation

❌ Disadvantages

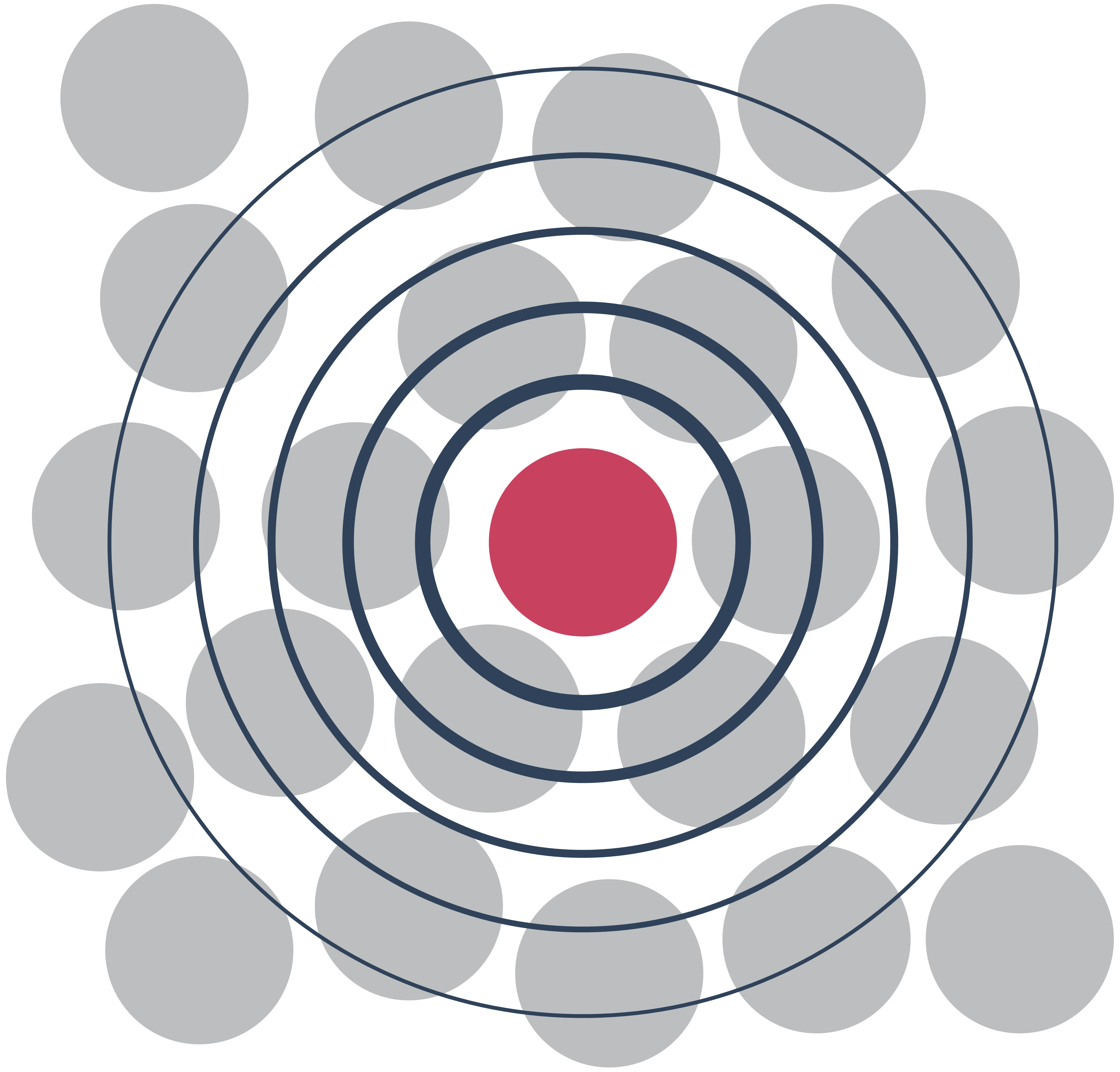
No invariance to translation/rotation

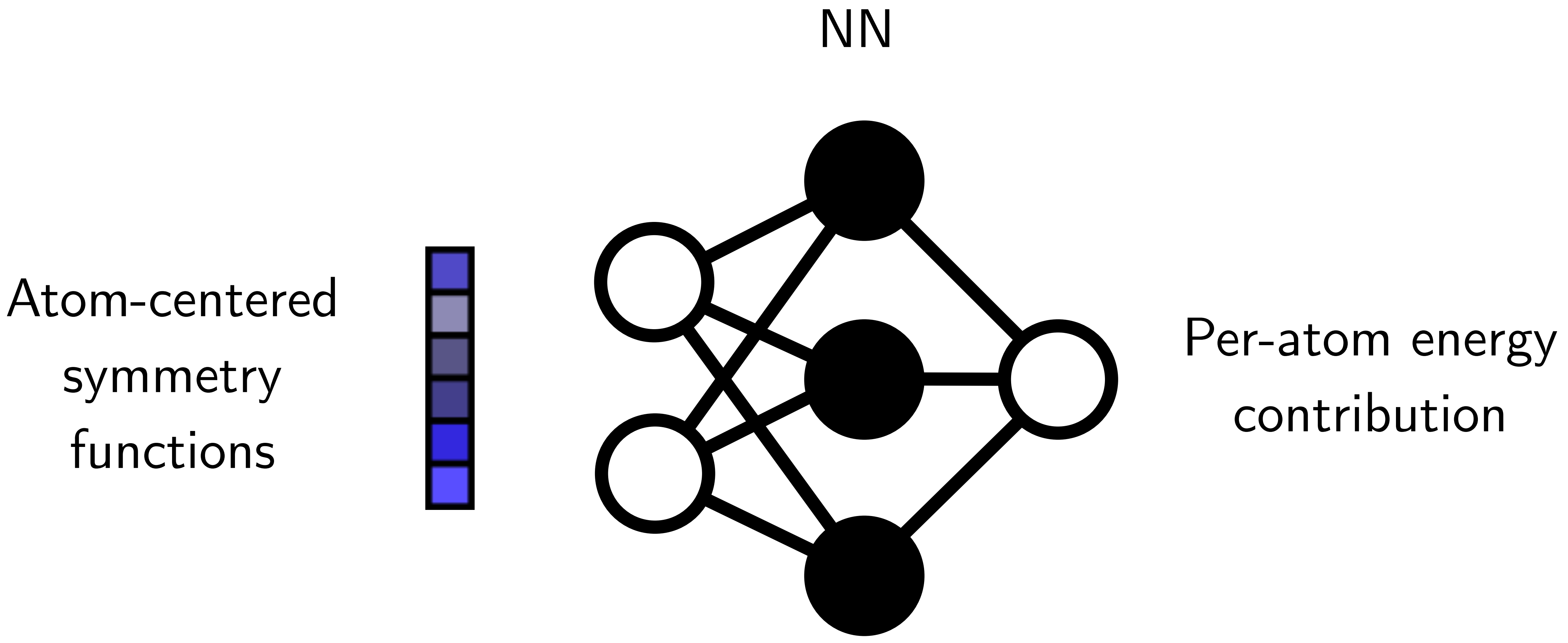
(and/or) No invariance to permutation

No forces

Does not scale to larger systems

Behler-Parrinello scheme




NN
Atom-centered symmetry functions
Per-atom energy contribution

The neighborhood of each atom is encoded into symmetry functions, which embed the rototranslational invariance of the system.

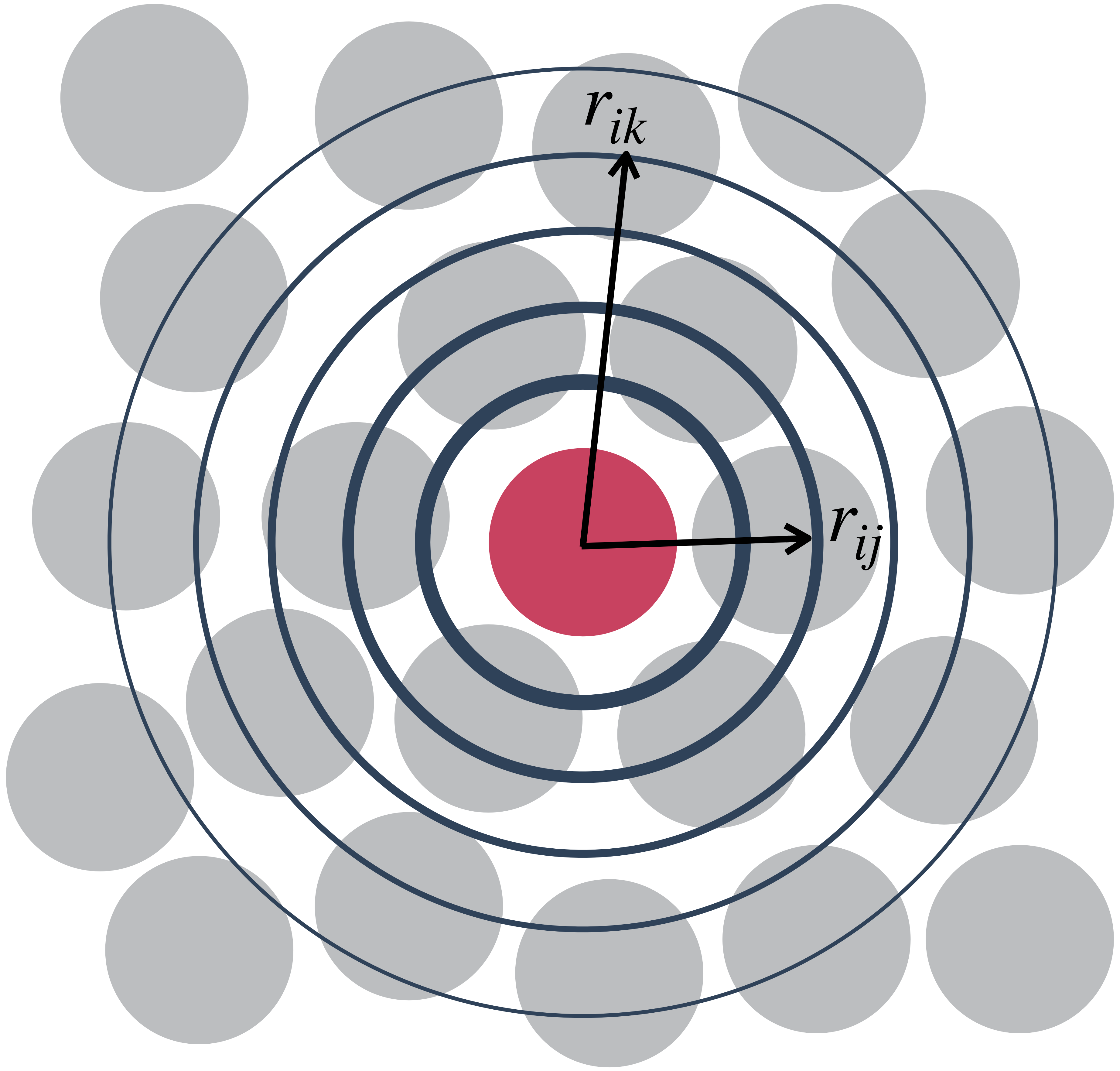
The final energy is computed as:

$$E = \sum_i E_i$$

Think about this: does dividing the energy make sense?

J. Behler and M. Parrinello. *PRL* **98**, 146401 (2007)

But how to define symmetry functions, now?



For each atom, N-body symmetry functions are calculated with different hyperparameters:

$$(\eta, r_s, \zeta)$$

Symmetry functions

cutoff

$$f_c(r_{ij}) = \frac{1}{2}\cos\left(\frac{\pi r_{ij}}{r_c}\right) + \frac{1}{2},\ r_{ij} < r_c$$

radial

$$G_i^1 = \sum_{i \neq j} e^{-\eta(r_{ij}-r_s)^2} f_c(r_{ij})$$

angular

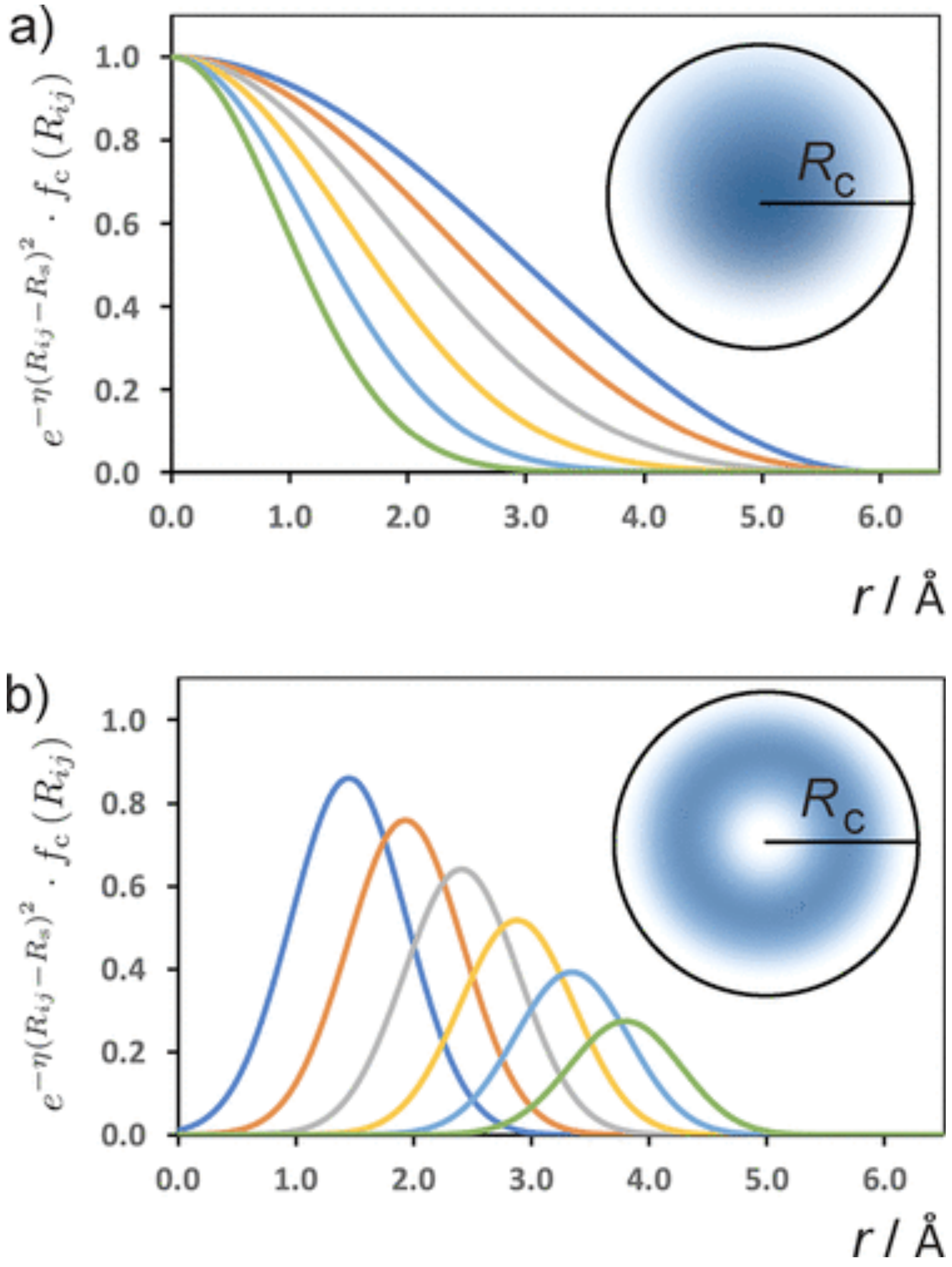
$$G_i^2 = 2^{1-\zeta} \sum_{i \neq j,k} (1 + \lambda \cos\theta_{ijk})^\zeta$$

$$\times\, e^{-\eta(r_{ij}^2+r_{ik}^2+r_{jk}^2)}$$

$$\times\, f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk})$$

J. Behler and M. Parrinello. *PRL* **98**, 146401 (2007)

Visualizing these symmetry functions



J. Behler. *Chem. Rev.* **121**, 10037 (2021)

Symmetry functions

cutoff

$$f_c(r_{ij}) = \frac{1}{2}\cos\left(\frac{\pi r_{ij}}{r_c}\right) + \frac{1}{2},\ r_{ij} < r_c$$

radial

$$G_i^1 = \sum_{i\neq j} e^{-\eta(r_{ij}-r_s)^2} f_c(r_{ij})$$

angular

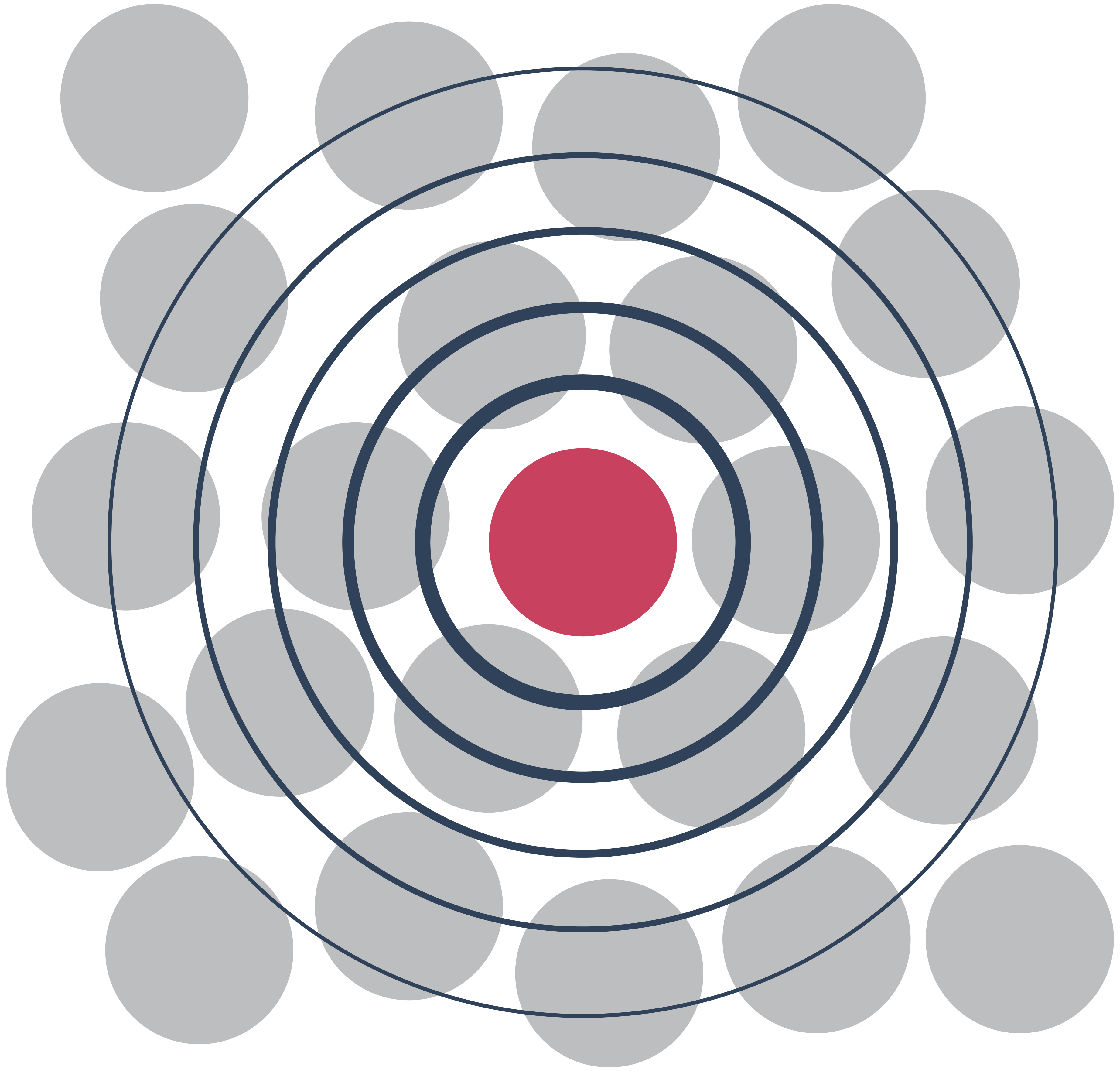
$$G_i^2 = 2^{1-\zeta}\sum_{i\neq j,k}(1+\lambda\cos\theta_{ijk})^\zeta$$

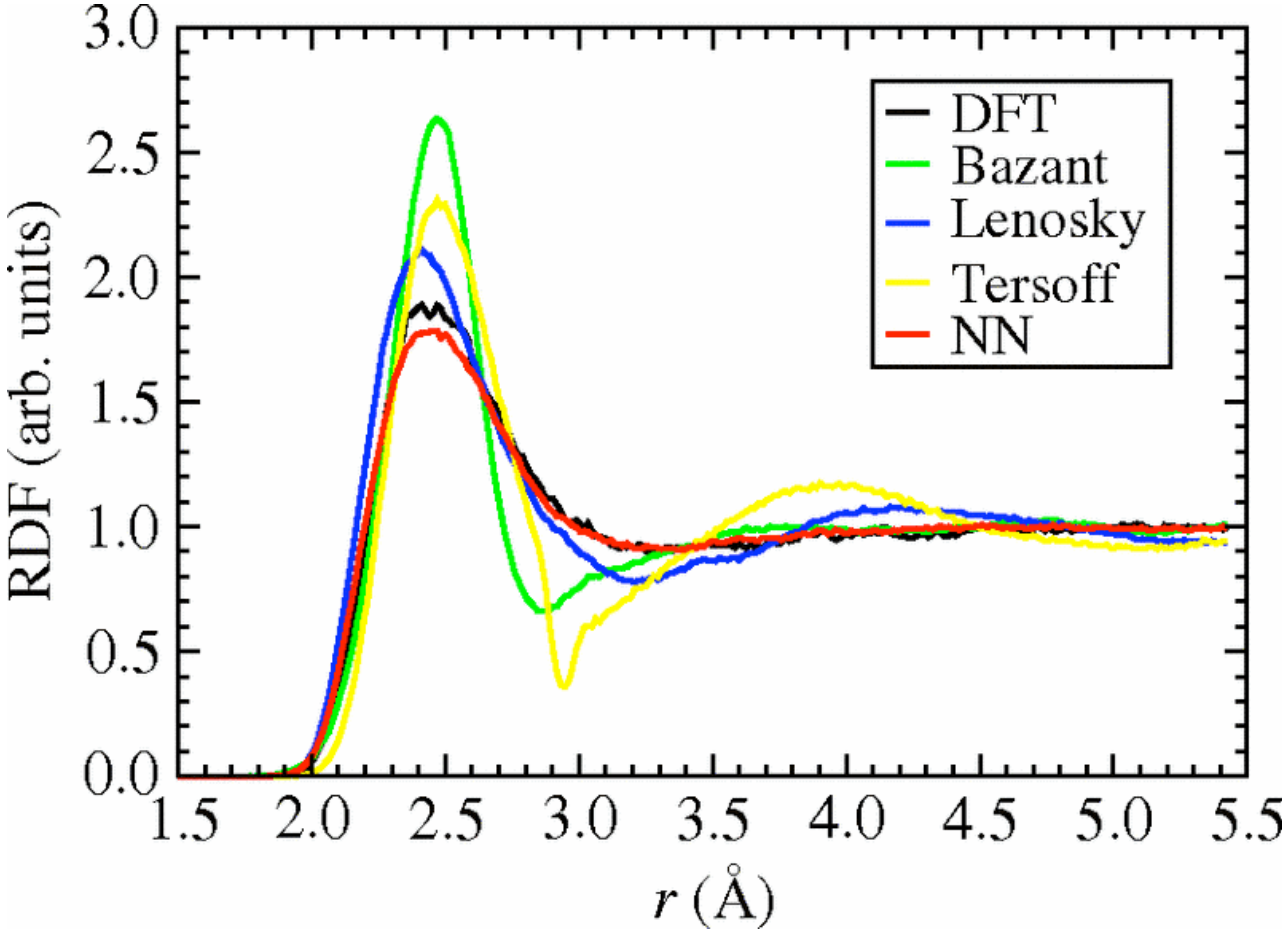
$$\times\, e^{-\eta(r_{ij}^2+r_{ik}^2+r_{jk}^2)}$$

$$\times f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk})$$

J. Behler and M. Parrinello. *PRL* **98**, 146401 (2007)

Results for silicon melt at 3000 K

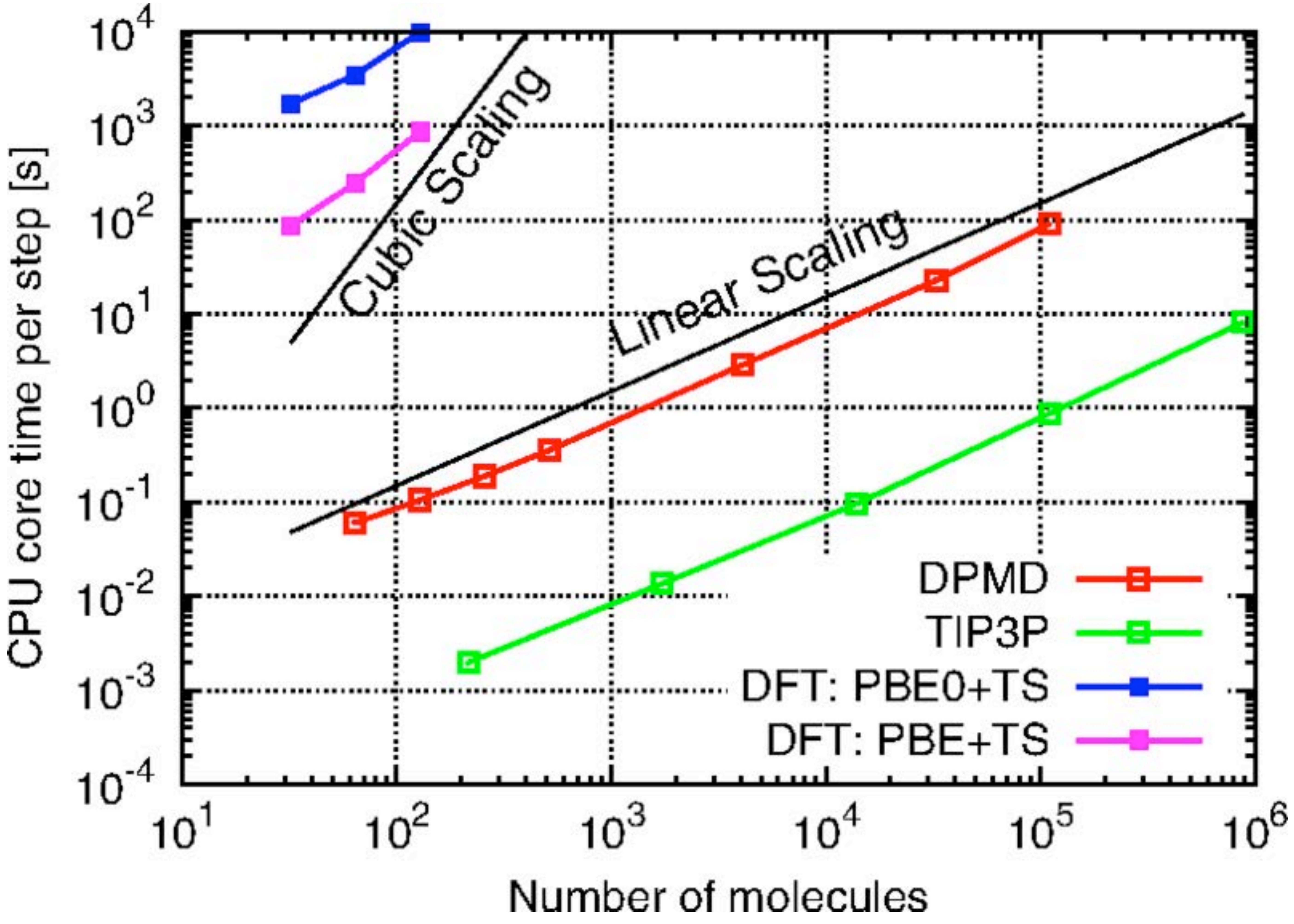




The NN better approximates the structural properties of Si melt than other potentials

J. Behler and M. Parrinello. *PRL* **98**, 146401 (2007)

Another benefit of using NN potentials: much better scaling with system size

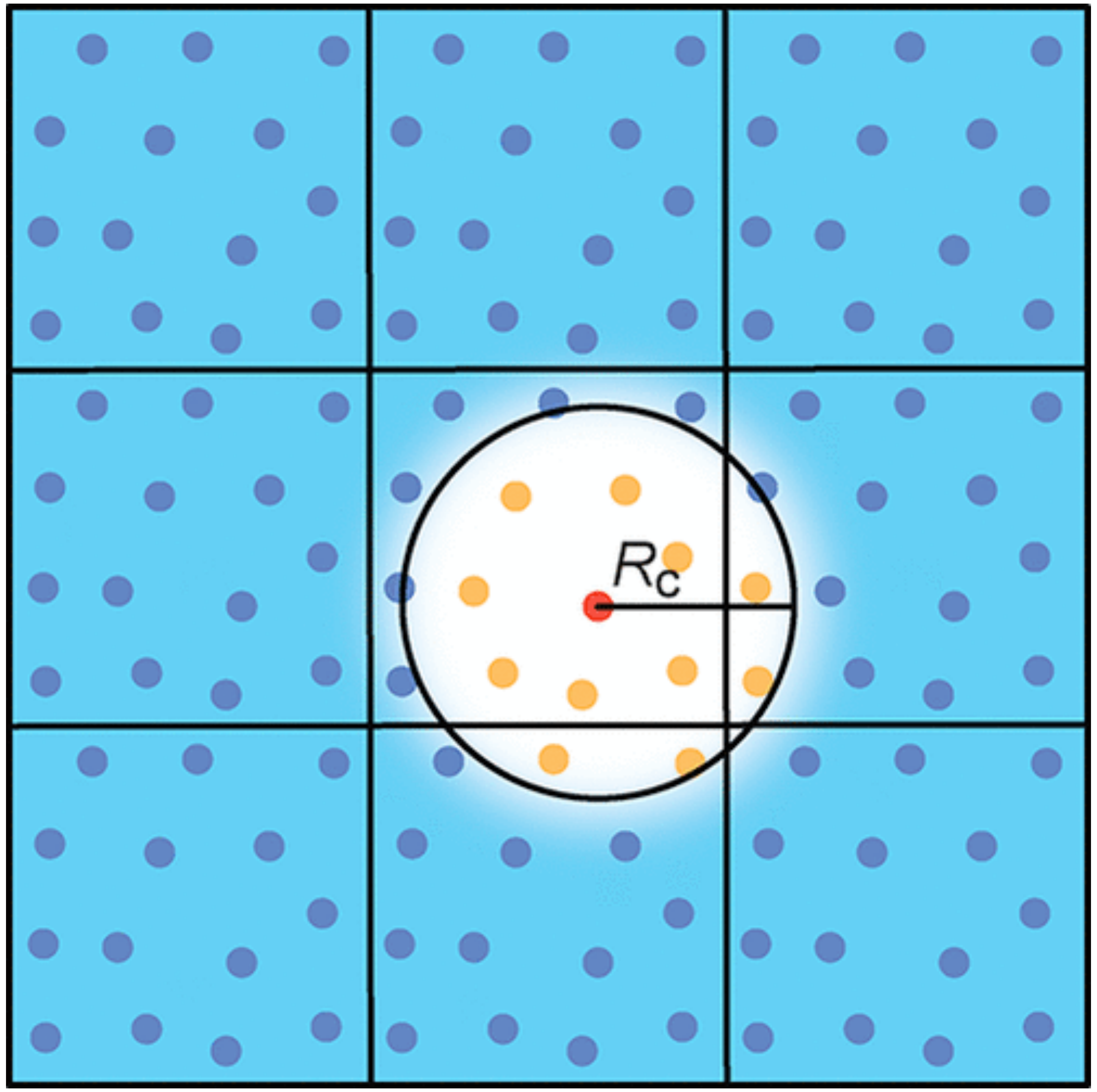


Because DFT scales with the cube of the number of electrons, it is impractical to perform simulations for very large systems (more than a few hundreds of atoms)

On the other hand, evaluating energies in NNIPs often scales linearly with the number of data points, and can be easily parallelized considering local potentials.

L. Zhang et al. *PRL* ***120***, 143001 (2018)

But the locality of the descriptor is also a shortcoming



In the figure above, only atoms within R_c of the central atom are considered.

Because of the locality of the descriptor, the final model cannot fit to interactions such as long-range potentials.

Increasing the cutoff drastically increases the number of atomic environments that have to be sampled.

For multi-element systems, this is even harder.

J. Behler. *Chem. Rev.* **121**, 10037 (2021)

J. Behler and M. Parrinello. *PRL* **98**, 146401 (2007)

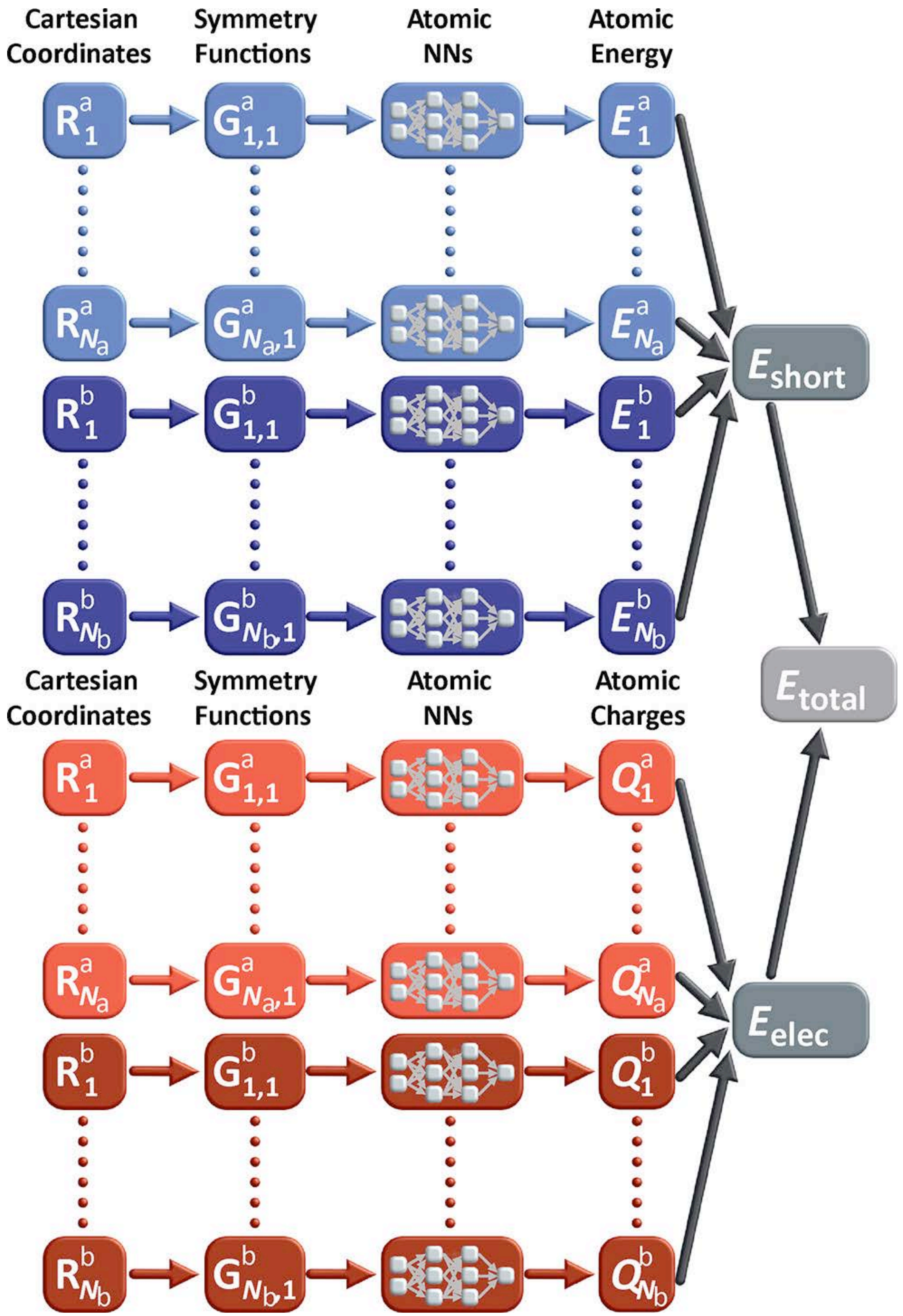
Different approaches have been proposed to deal with long-range interactions

The first option is to include explicit Coulomb terms for fixed charges for each atomic environment:

$$E = \sum_i E_i + E_{\text{elec}}$$

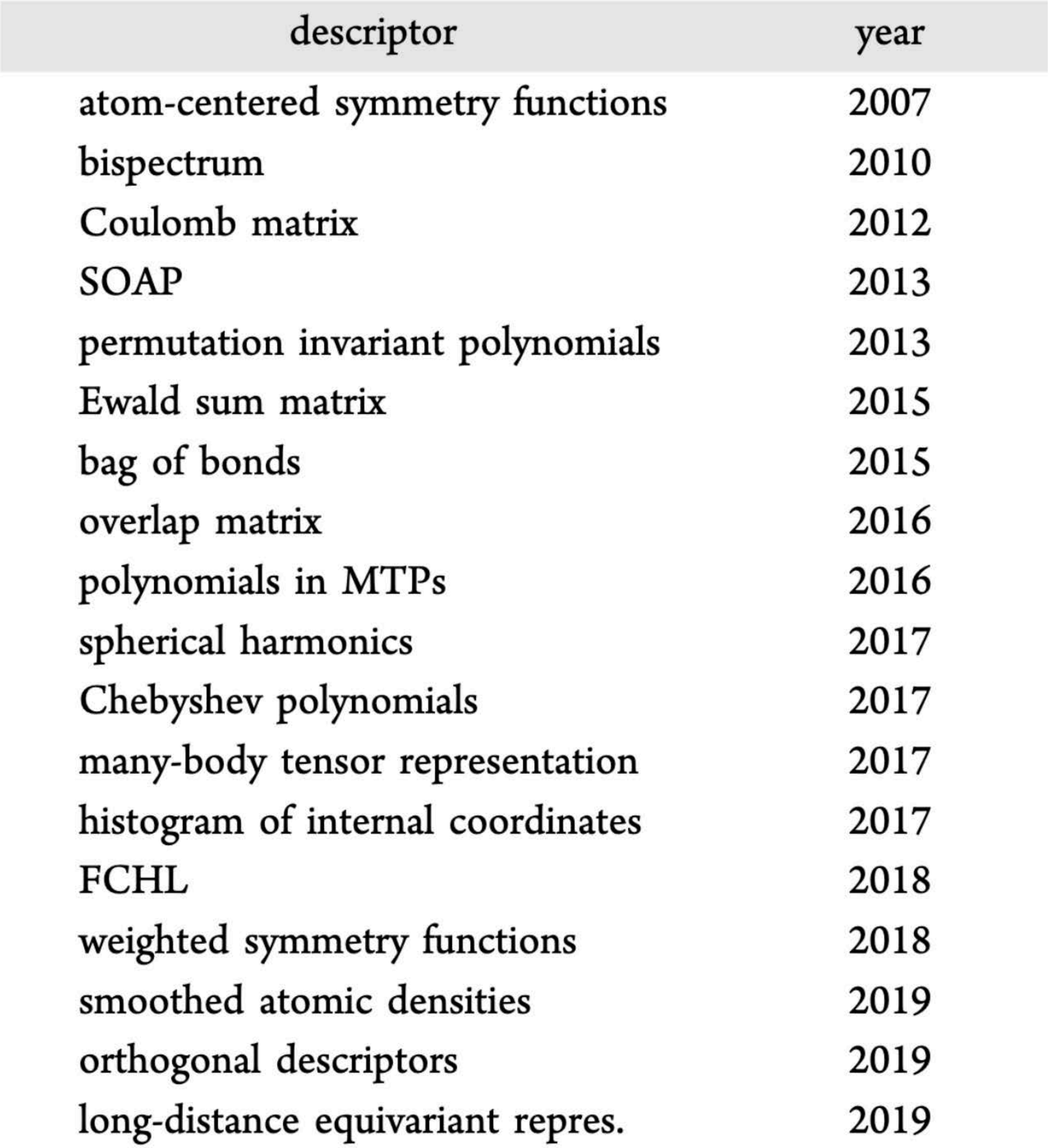
But a better option may be to predict atomic charges using the symmetry functions, then predict a short-range energy and a long-range energy (figure on the right),

$$E = \sum_i E_i + \frac{1}{2} \sum_{i \neq j} \kappa_{ij}(r_{ij}) \frac{Q_i Q_j}{r_{ij}}$$



J. Behler. *Chem. Rev.* **121**, 10037 (2021)

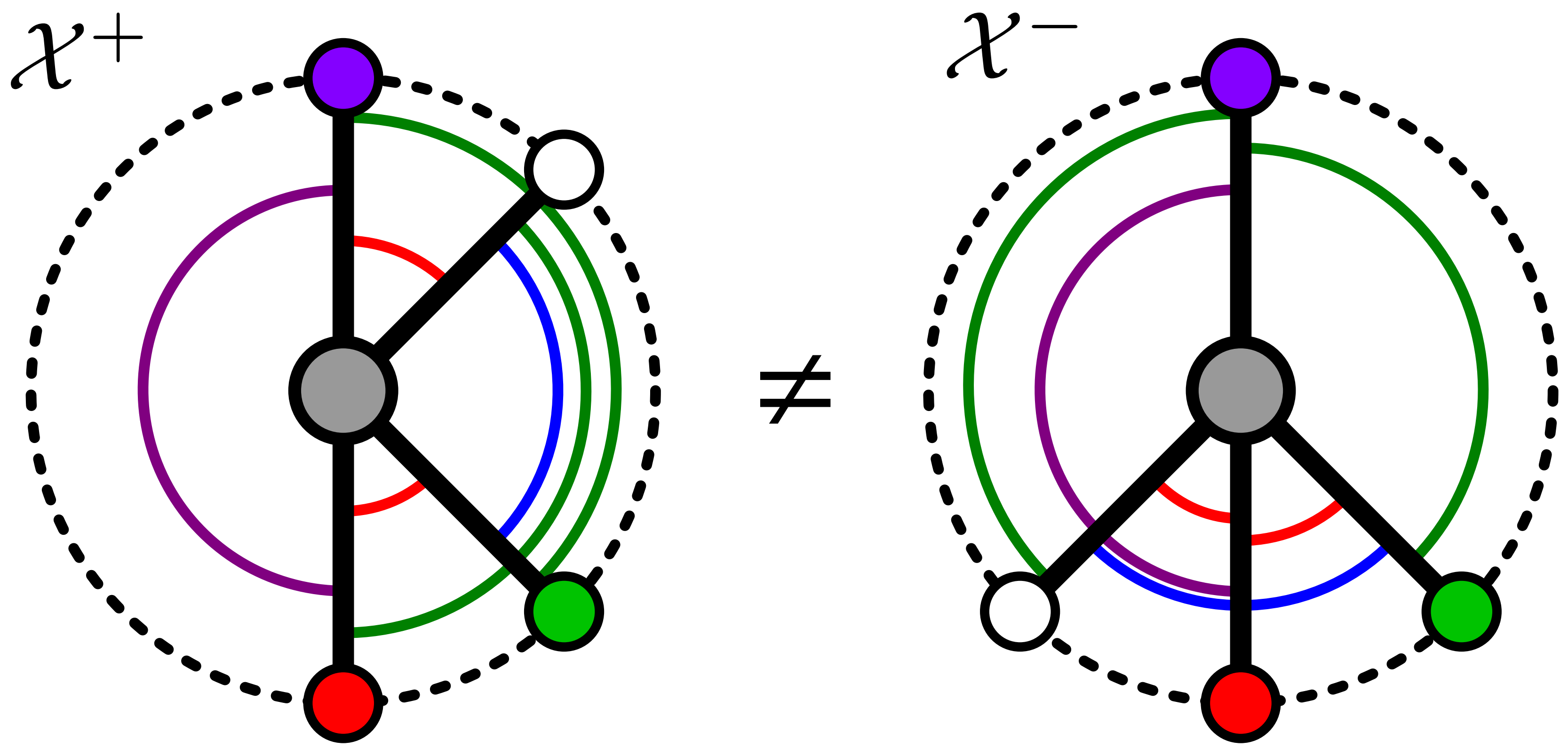
But there's more: we can use several representations



descriptor	year
atom-centered symmetry functions	2007
bispectrum	2010
Coulomb matrix	2012
SOAP	2013
permutation invariant polynomials	2013
Ewald sum matrix	2015
bag of bonds	2015
overlap matrix	2016
polynomials in MTPs	2016
spherical harmonics	2017
Chebyshev polynomials	2017
many-body tensor representation	2017
histogram of internal coordinates	2017
FCHL	2018
weighted symmetry functions	2018
smoothed atomic densities	2019
orthogonal descriptors	2019
long-distance equivariant repres.	2019

J. Behler. *Chem. Rev.* **121**, 10037 (2021)

And there is a lot of discussion on why representations matter so much:

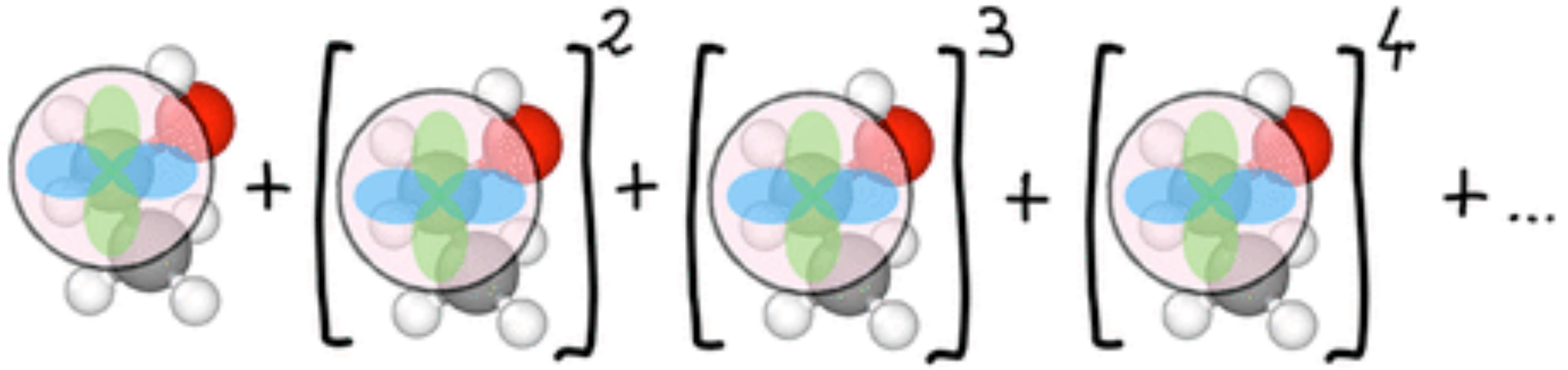


(the environments above, for example, are different, but have the same histogram of triangles)

S. Pozdnyakov et al. *PRL* **125**, 166001 (2020)

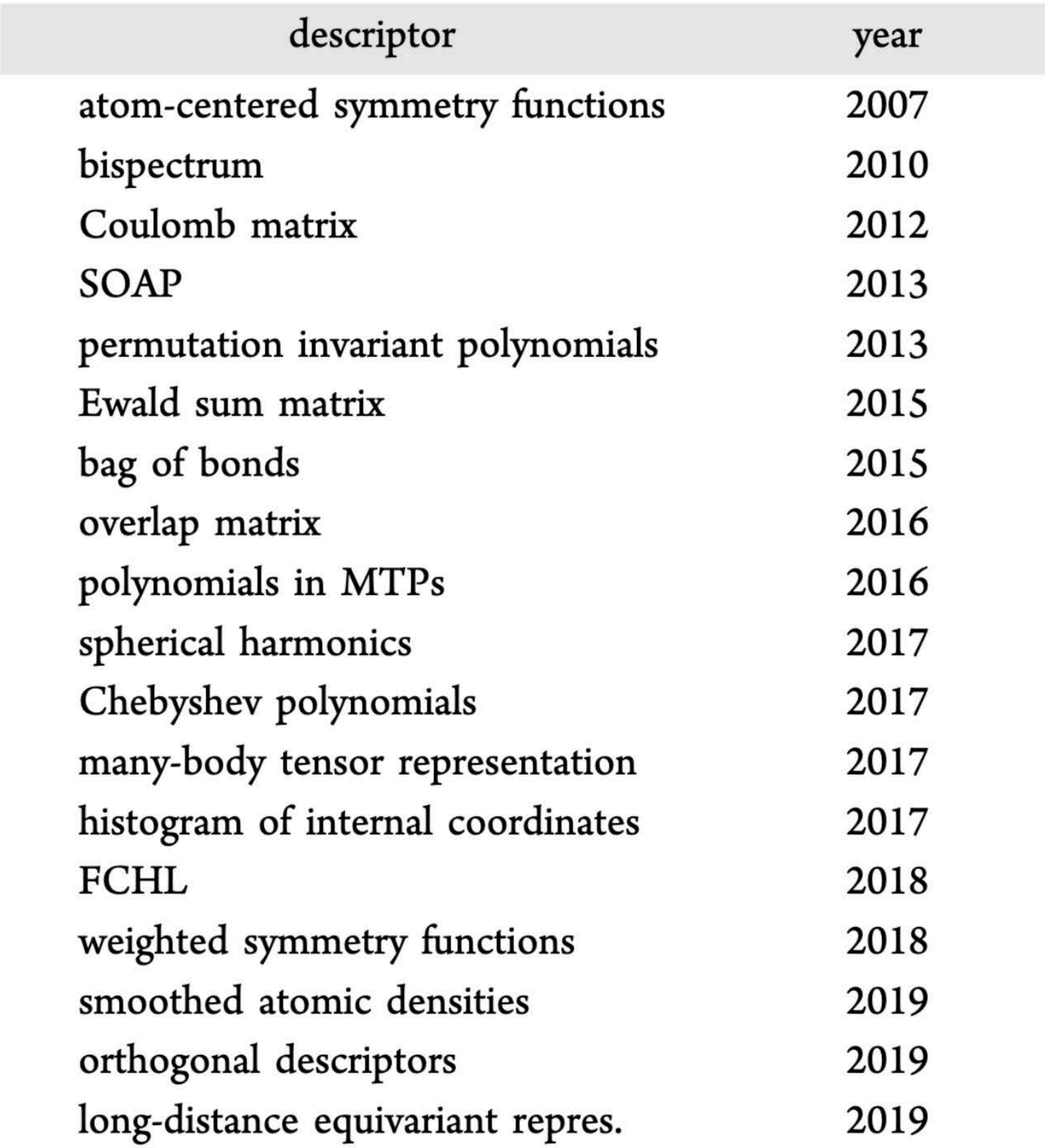
Furthermore, many-body interactions also matter a lot!

$$E_{tot} = E_{1b} + E_{2b} + E_{3b} + E_{4b} + \dots$$



D. Kovács et al. *JCTC* **17** (12), 7696 (2021)

But there's more: we can use several representations



descriptor	year
atom-centered symmetry functions	2007
bispectrum	2010
Coulomb matrix	2012
SOAP	2013
permutation invariant polynomials	2013
Ewald sum matrix	2015
bag of bonds	2015
overlap matrix	2016
polynomials in MTPs	2016
spherical harmonics	2017
Chebyshev polynomials	2017
many-body tensor representation	2017
histogram of internal coordinates	2017
FCHL	2018
weighted symmetry functions	2018
smoothed atomic densities	2019
orthogonal descriptors	2019
long-distance equivariant repres.	2019

J. Behler. *Chem. Rev.* **121**, 10037 (2021)

And there is a lot of discussion on why representations matter so much:

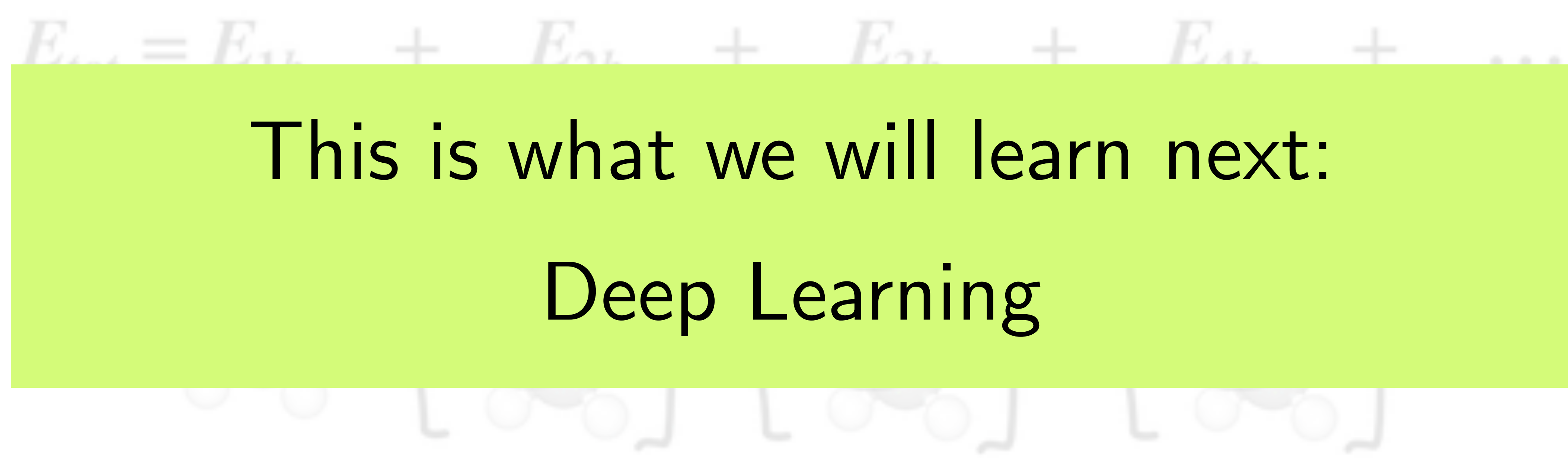
What if we are missing critical factors when proposing new descriptors?

(th the

sam 20)

The question now becomes: can we do better than designing representations?

Furthermore, many-body interactions also matter a lot!

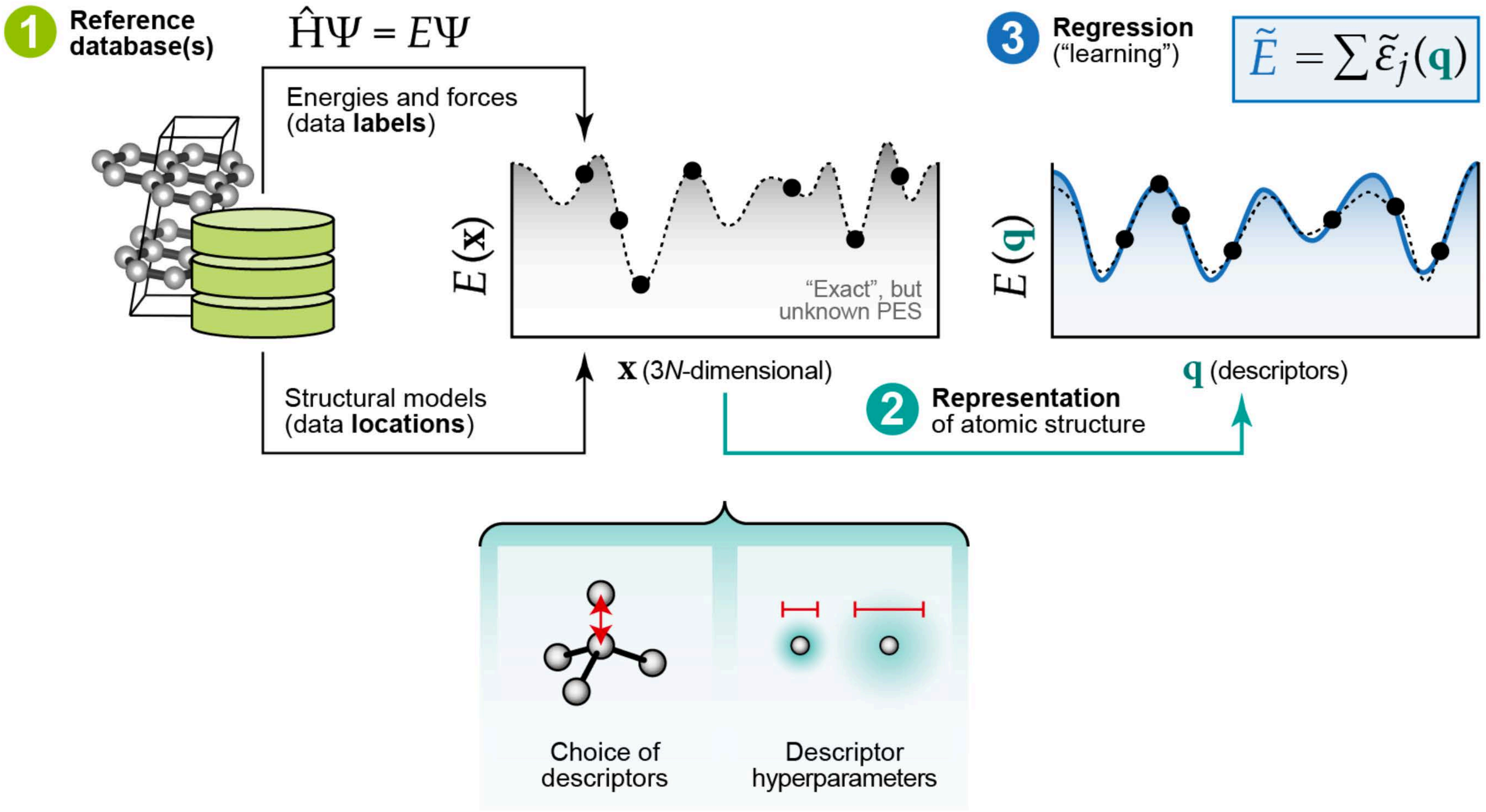


This is what we will learn next:

Deep Learning

D. Kovács et al. *JCTC* **17** (12), 7696 (2021)

To summarize what we have learned so far


1 Reference database(s)
$\hat{H}\Psi = E\Psi$
Energies and forces (data **labels**)
Structural models (data **locations**)
$E(\mathbf{x})$
"Exact", but unknown PES
$\mathbf{x}$ (3N-dimensional)
3 Regression ("learning")
$\tilde{E} = \sum \tilde{\varepsilon}_j(\mathbf{q})$
$E(\mathbf{q})$
$\mathbf{q}$ (descriptors)
2 Representation of atomic structure
Choice of descriptors
Descriptor hyperparameters

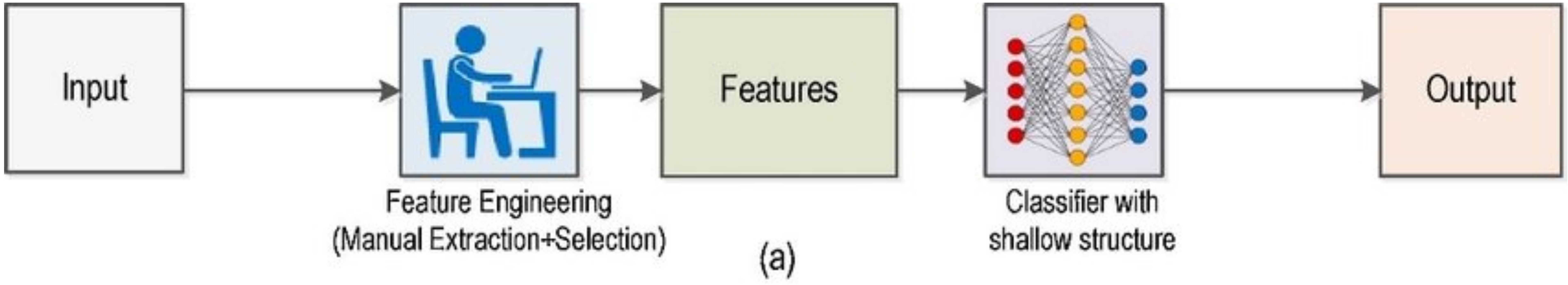
J. Morrow et al. *arXiv*:2211.12484 (2022)

3. Deep Learning Potentials

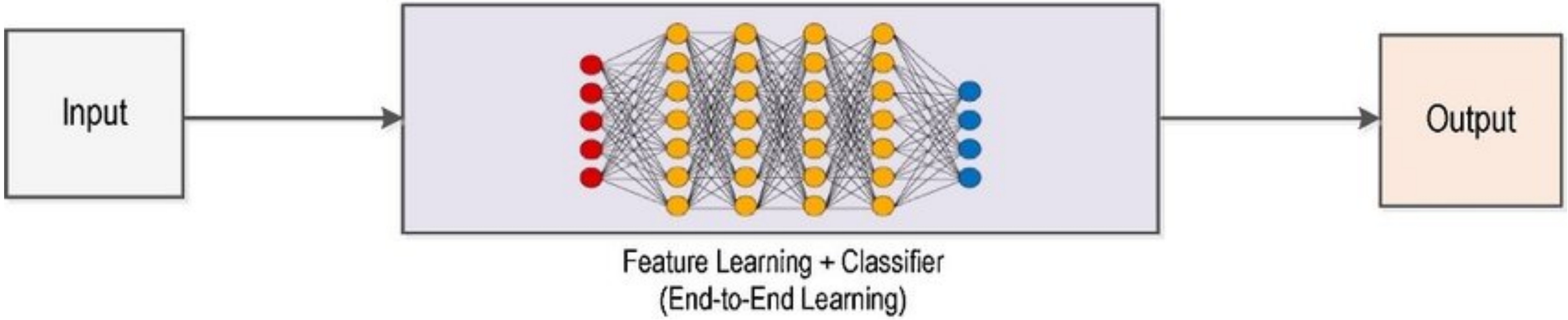
Why use deep learning for interatomic potentials?

A bit of history: why deep learning actually succeeded in other fields?

Traditional ML pipeline



Deep learning pipeline

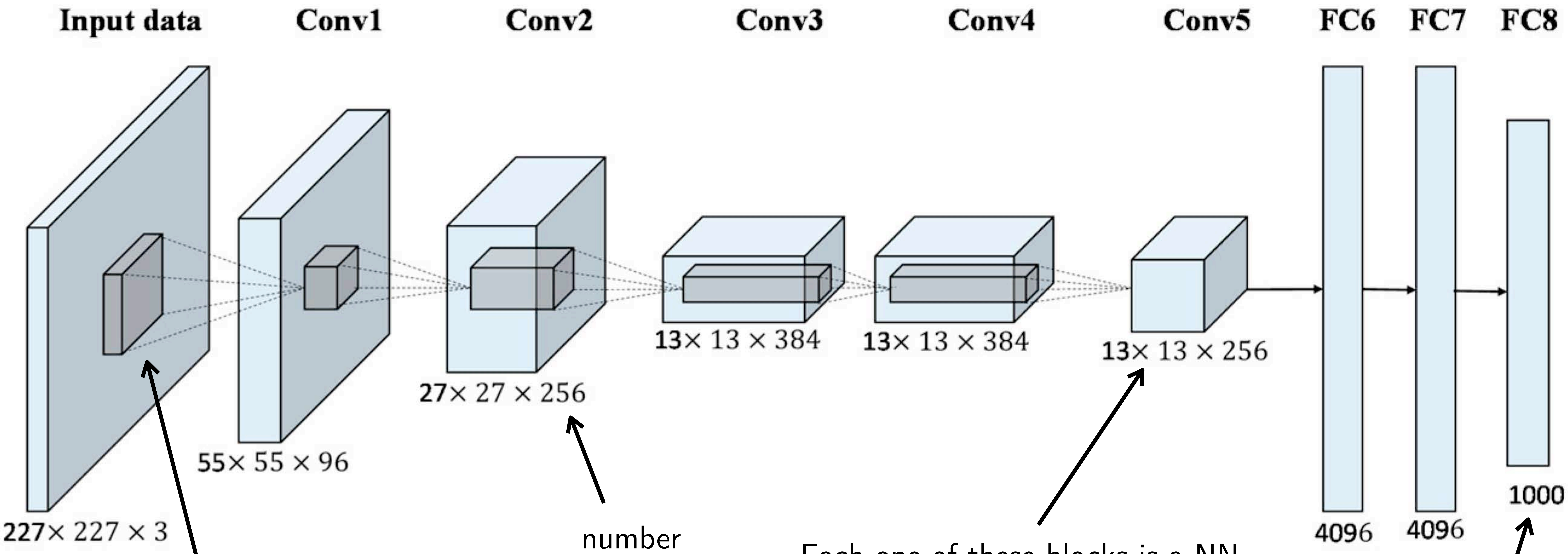


Deep Learning: features identified along with the training process.

Figure from J. Wang et al. *J. Manuf. Syst.* **48**, 144 (2018)

In deep learning, the neural network architectures are more complicated ("deep")

AlexNet architecture



Pixels x pixels x channels
(3 for RGB)

"kernel"

number
of filters

Each one of these blocks is a NN
(tensor + bias + activation)

number
of classes

A. Krizhevsky et al. *NeurIPS* **25** (2012)

Image from X. Han et al. *Remote Sens.* **9** (8), 848 (2017)

If we take each of the filters, we will see the features extracted from the dataset:

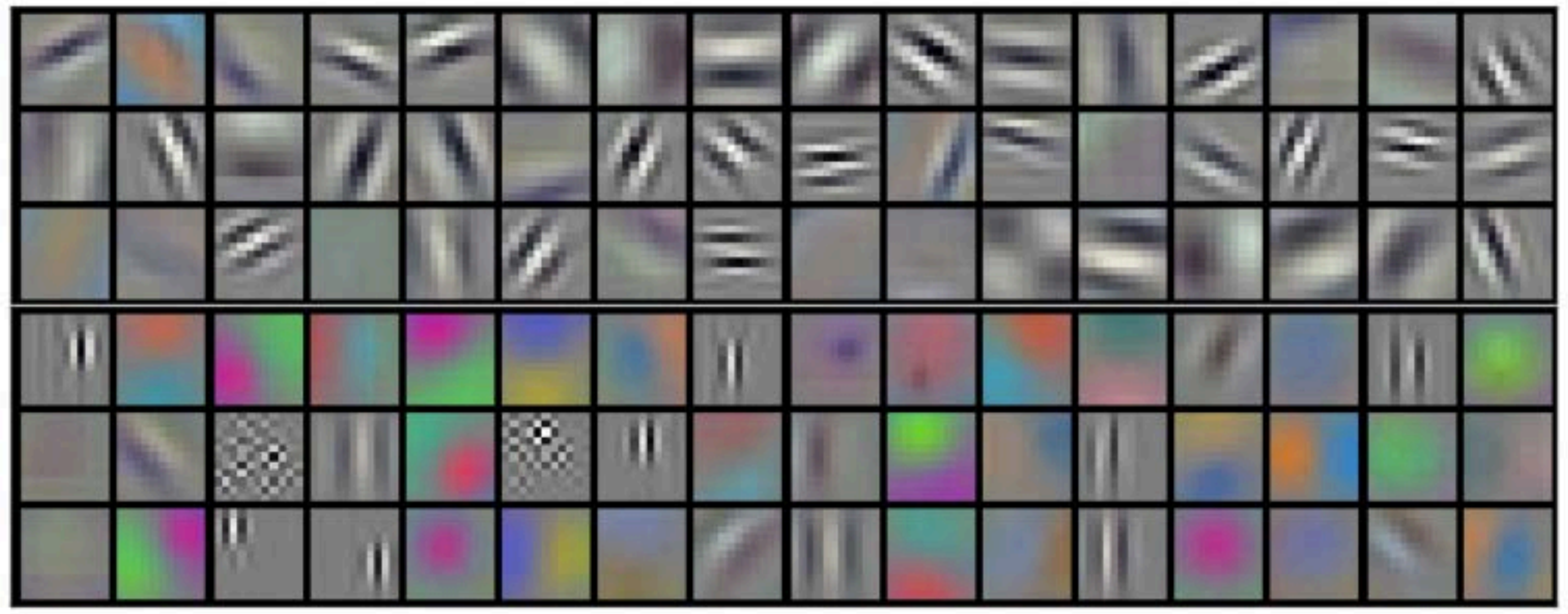
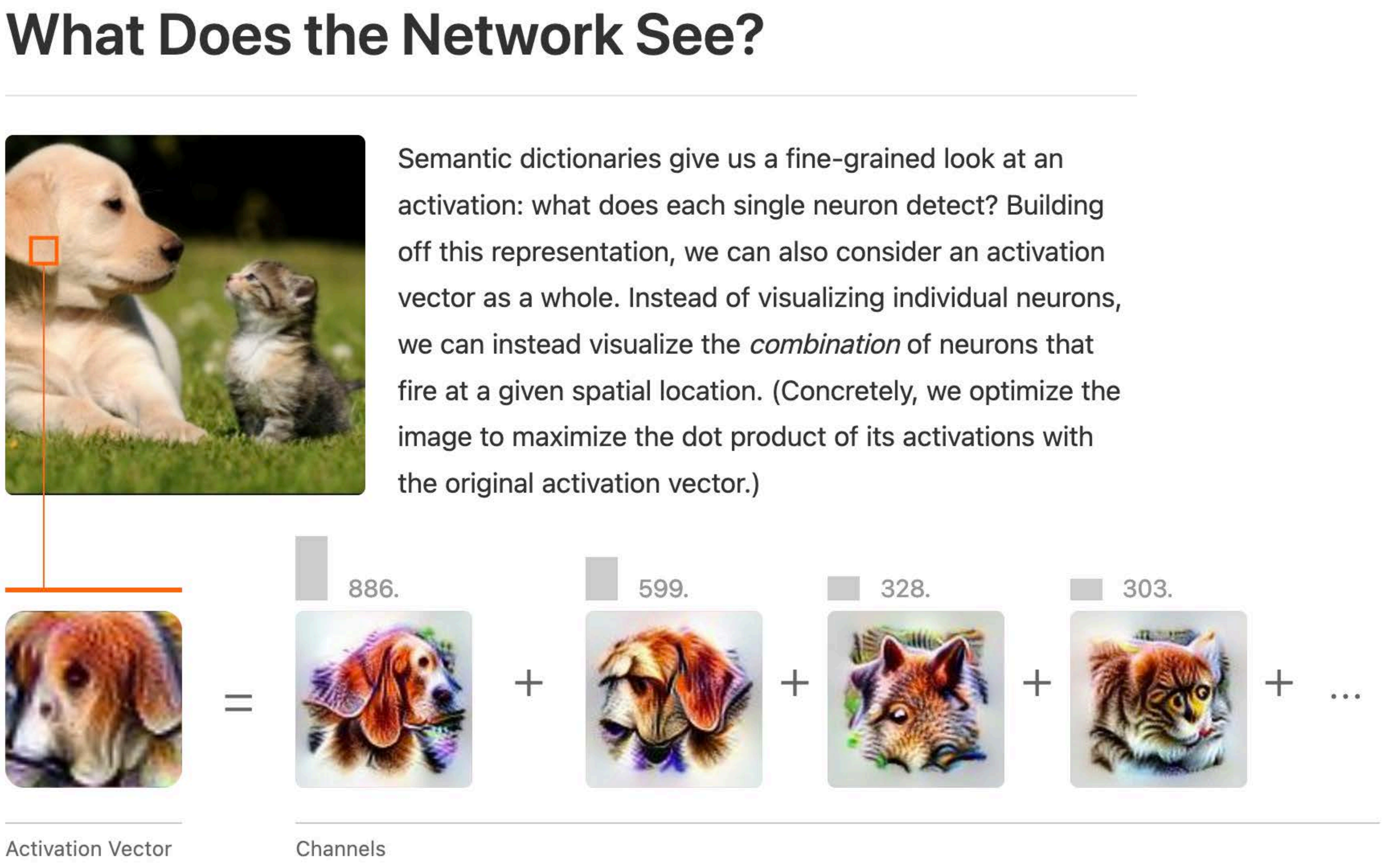


Figure 3: 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.



What Does the Network See?

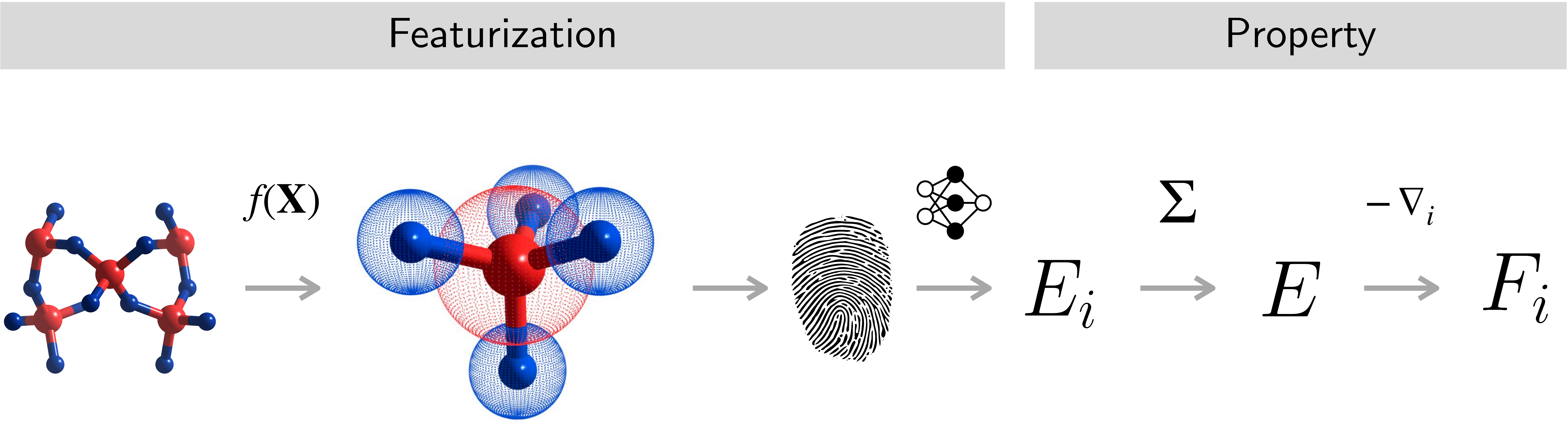
Semantic dictionaries give us a fine-grained look at an activation: what does each single neuron detect? Building off this representation, we can also consider an activation vector as a whole. Instead of visualizing individual neurons, we can instead visualize the *combination* of neurons that fire at a given spatial location. (Concretely, we optimize the image to maximize the dot product of its activations with the original activation vector.)

Mini Tutorial: CNN activations

A. Krizhevsky et al. *NeurIPS* **25** (2012)

C. Olah et al. *Distill* (2018). DOI: 10.23915/distill.00010

Back to NNIPs: crafting features vs learning representations

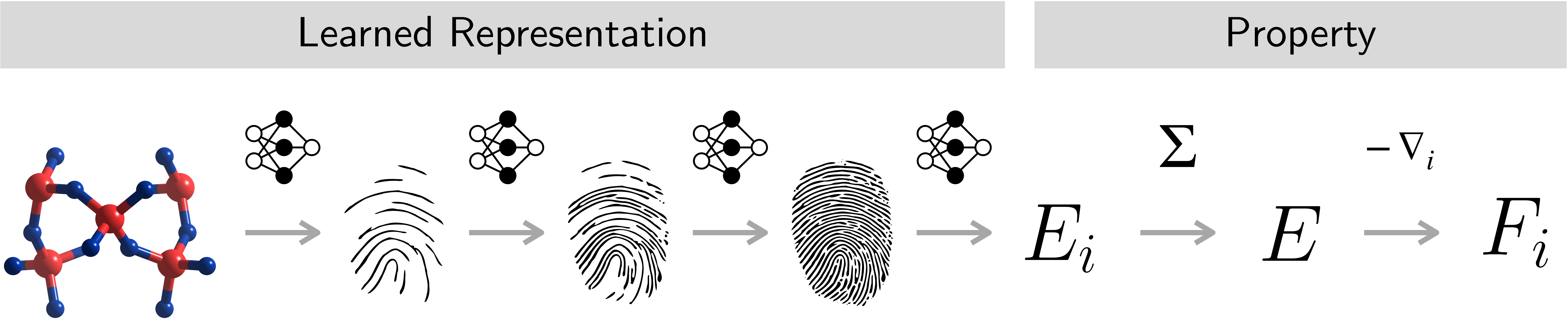


Featurization

Property

$\xrightarrow{f(\mathbf{X})}$

$E_i \xrightarrow{\Sigma} E \xrightarrow{-\nabla_i} F_i$

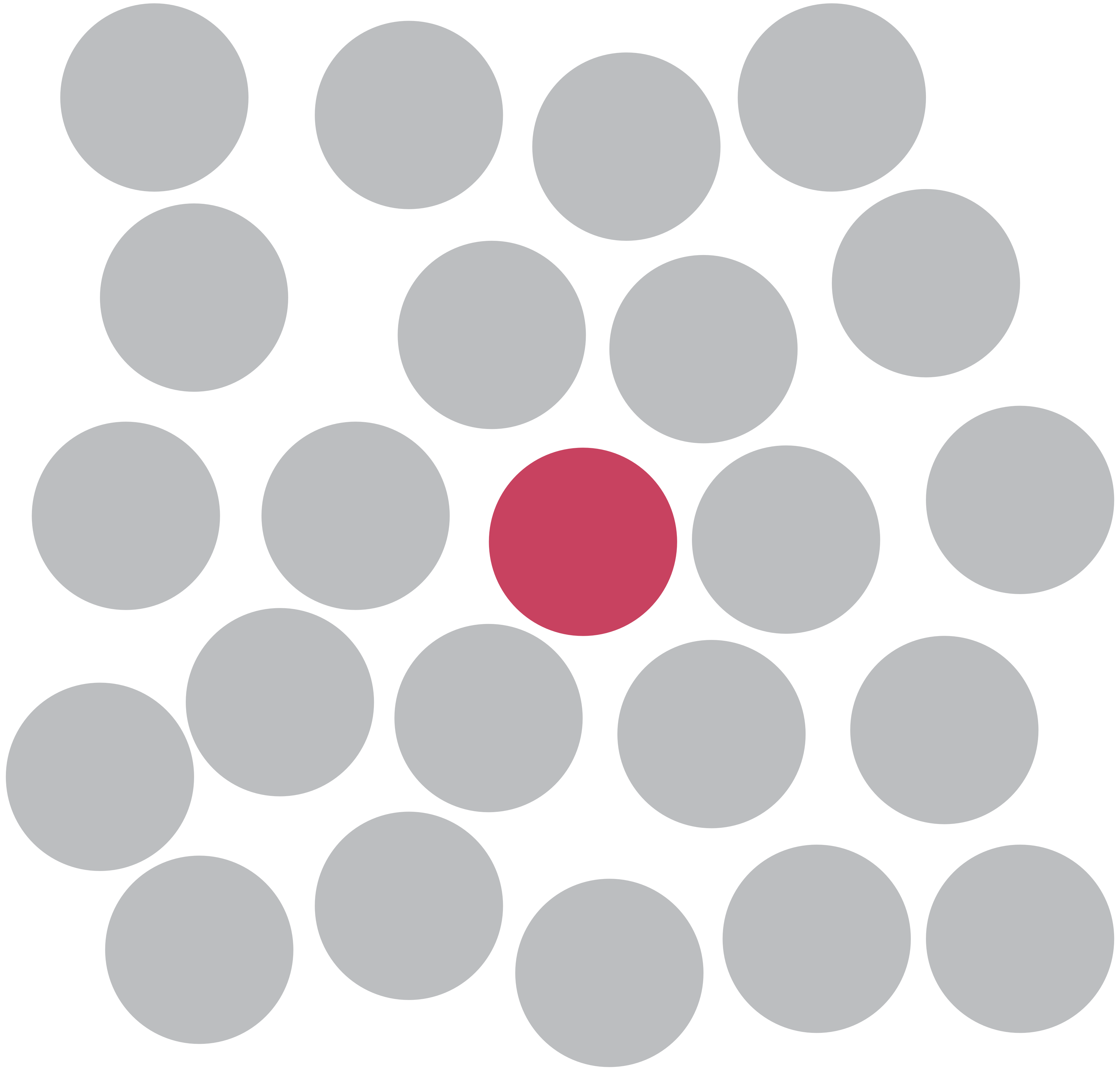


Learned Representation

Property

$E_i \xrightarrow{\Sigma} E \xrightarrow{-\nabla_i} F_i$

But, again, how to represent our data?



In images, a mapping between a (N x N x 3) image and (C) classes is well-defined:

$$f : \mathbb{R}^{N \times N \times 3} \to \mathbb{R}^{C}$$

(assuming the prediction of class logits)

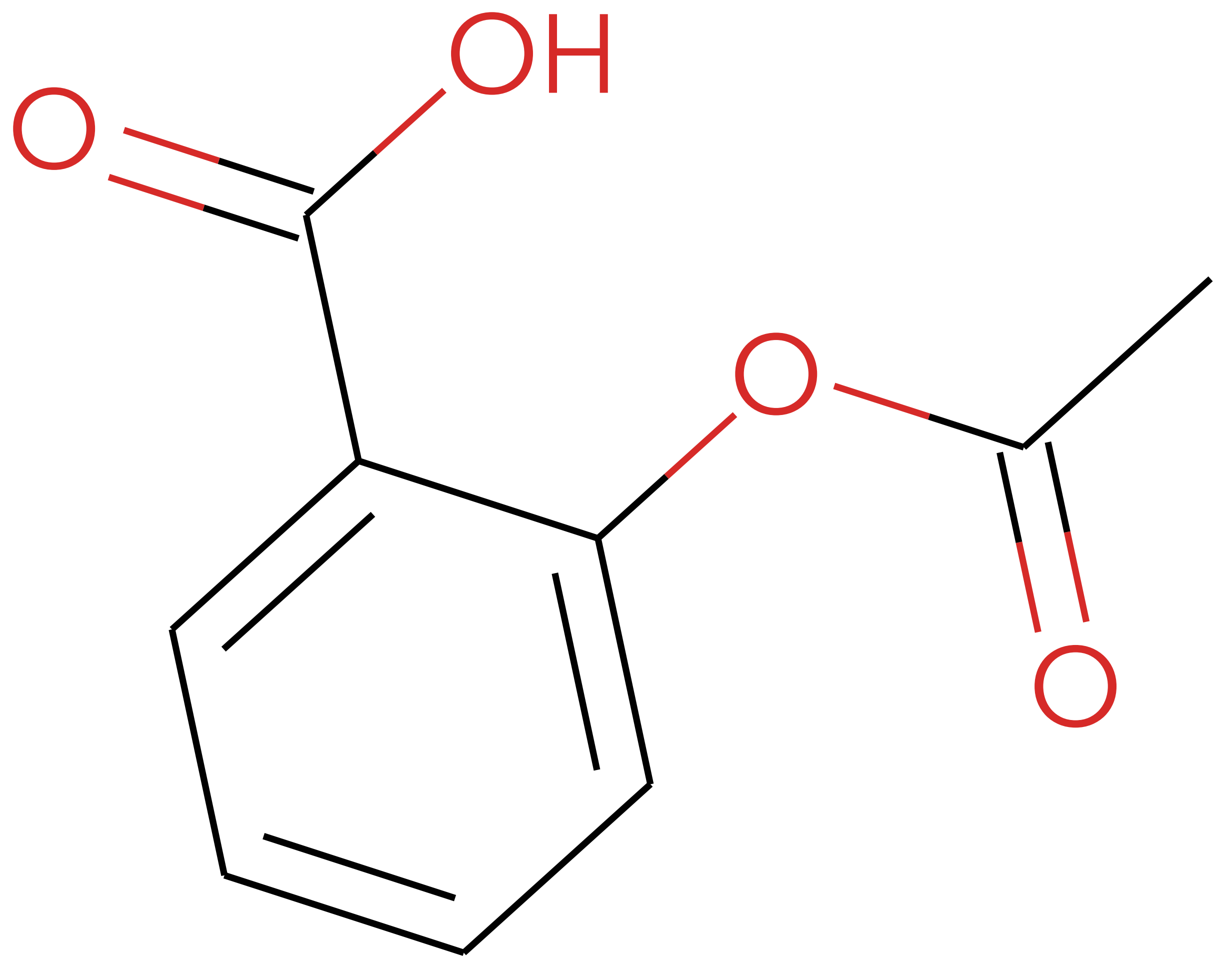
In materials or molecules, our data is not an image nor a sequence. Instead, our mapping is for a set of atoms in the 3D space:

$$V : \mathbb{Z} \times \mathbb{R}^{N \times 3} \to \mathbb{R}$$

with the 3D space not well represented by an image.

Why images are not good enough?

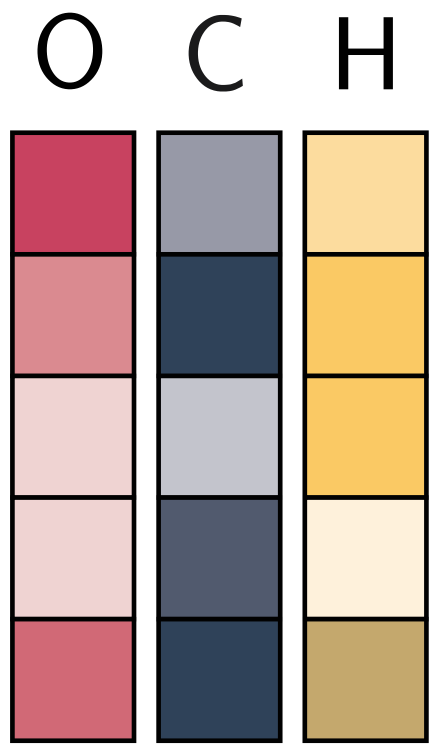
A more natural way to represent the data is a graph



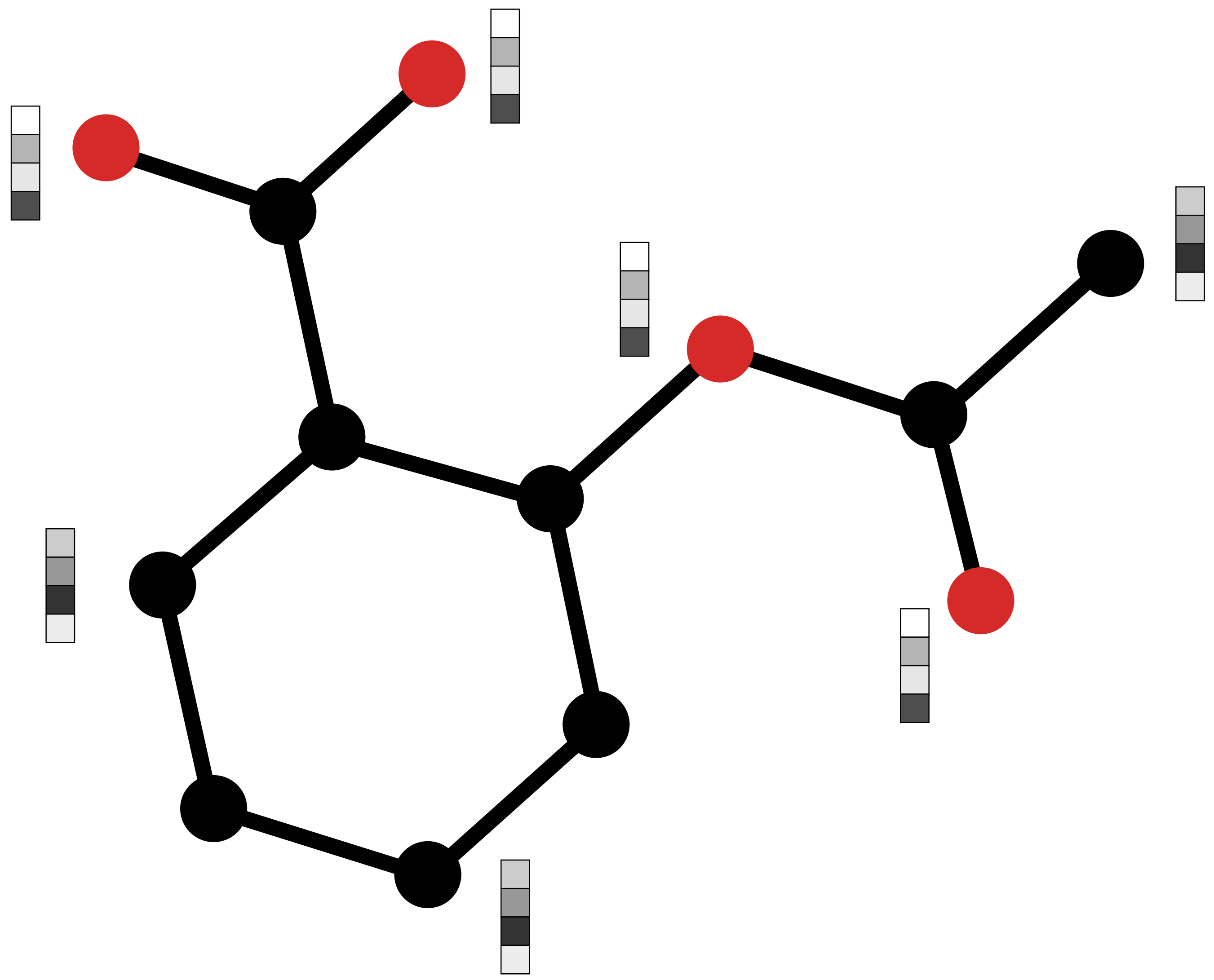
By defining a molecule or material system as a graph,

$$G = (V, E)$$

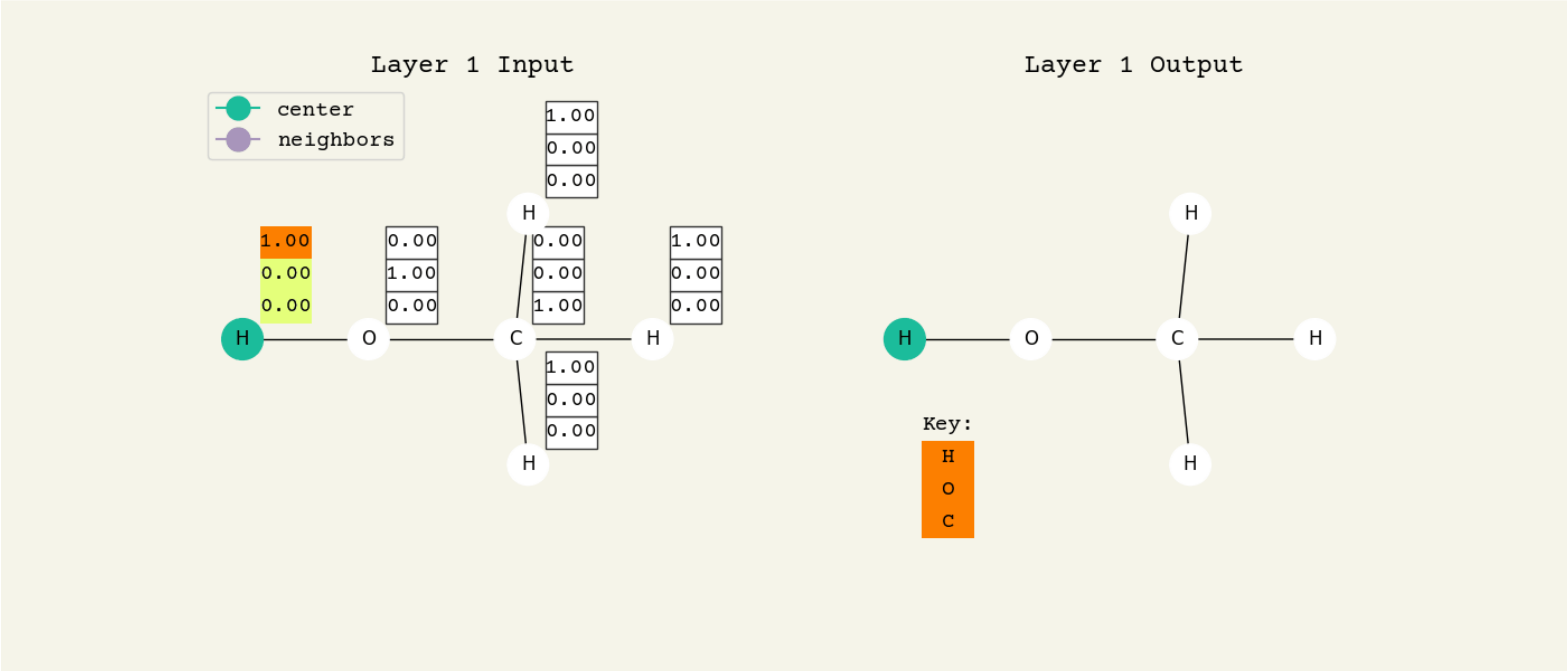
We can initialize some features to each element (or even use random ones):



Now we just need to learn a representation for each atomic environment



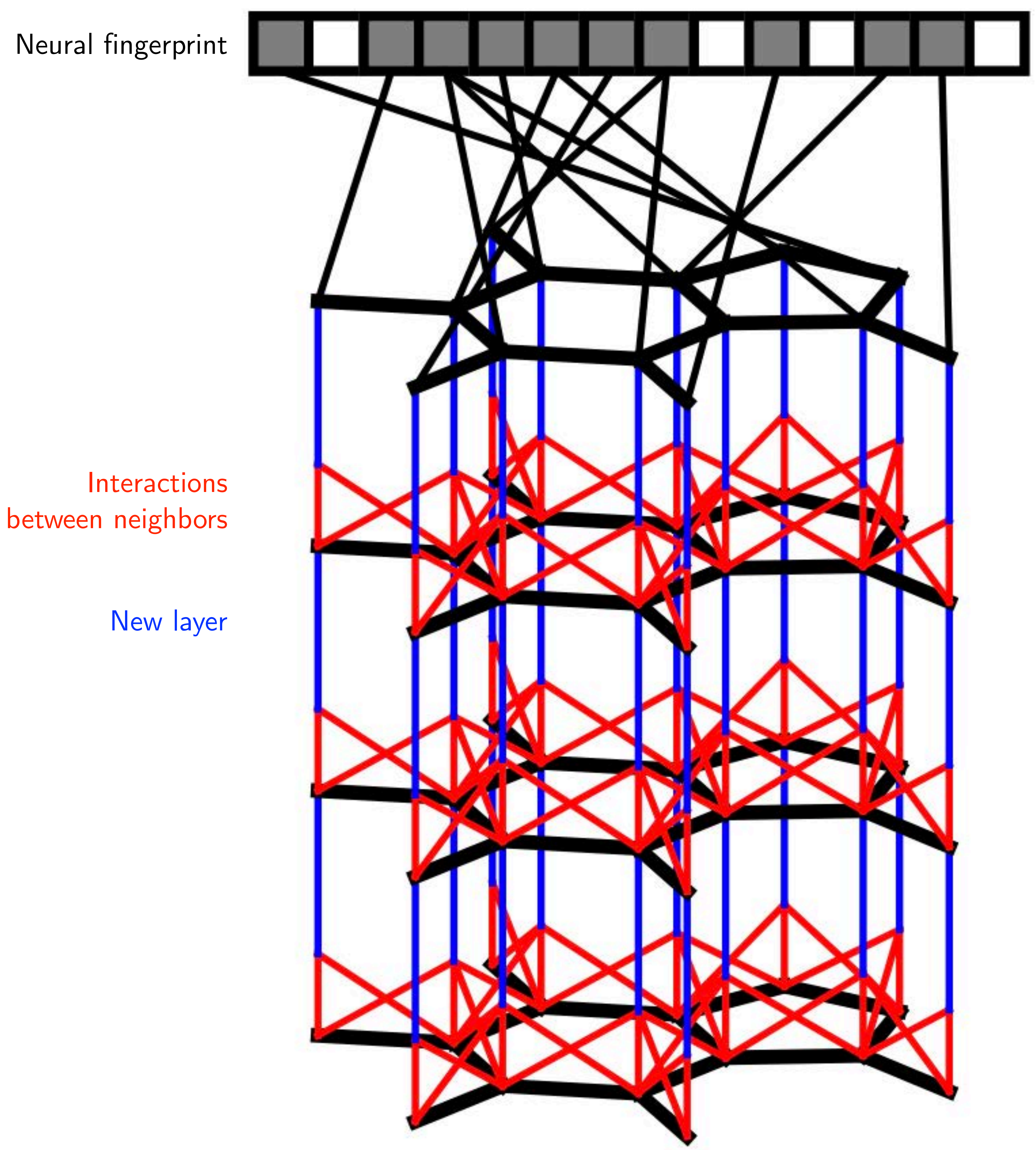
How does a graph convolution look like in a molecule?



A. White. *Living Journal of Computational Molecular Science*
https://dmol.pub (2022)

How does a graph convolution look like in a molecule?

Graph convolution on molecule



D. Duvenaud et al. *NeurIPS* **28** (2015)

In some molecular systems, the graph is simply the connectivity graph of a molecule (covalent bonds).

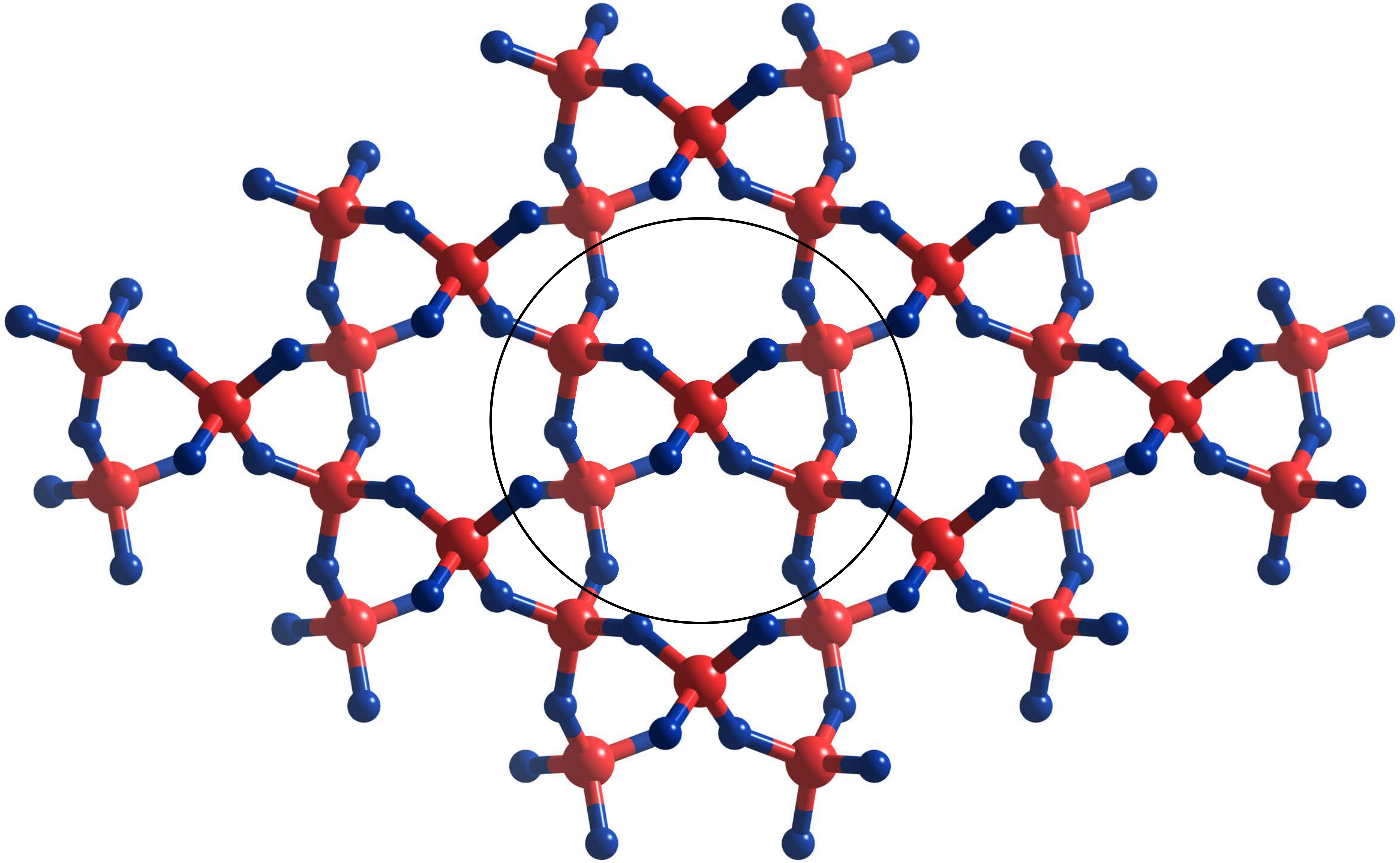
Each atom interacts with its neighbors, and a "filter" is trained by understanding the interactions between neighbors.

This enables us to predict properties from the molecular graph.

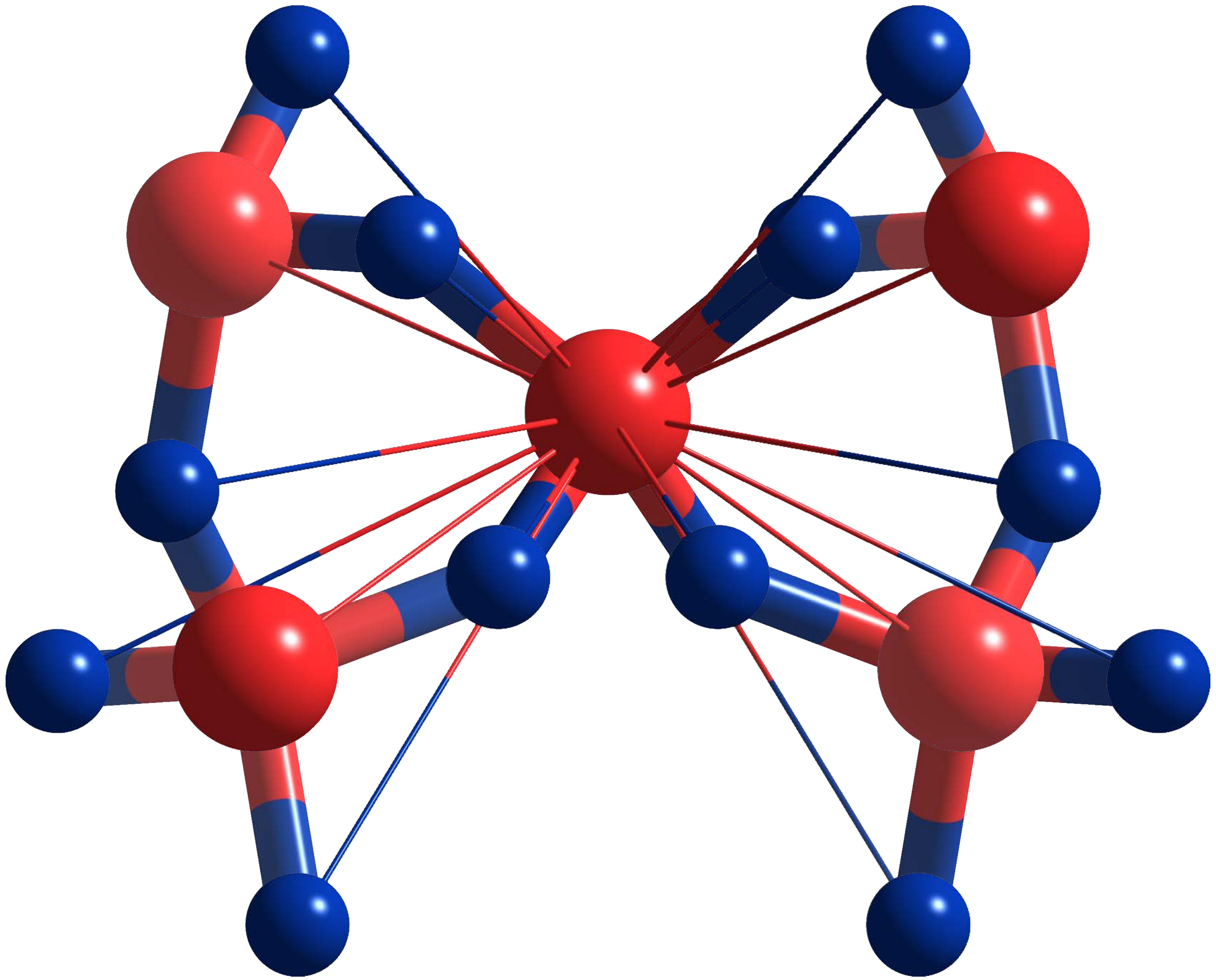
Some problems with a graph-only approach:

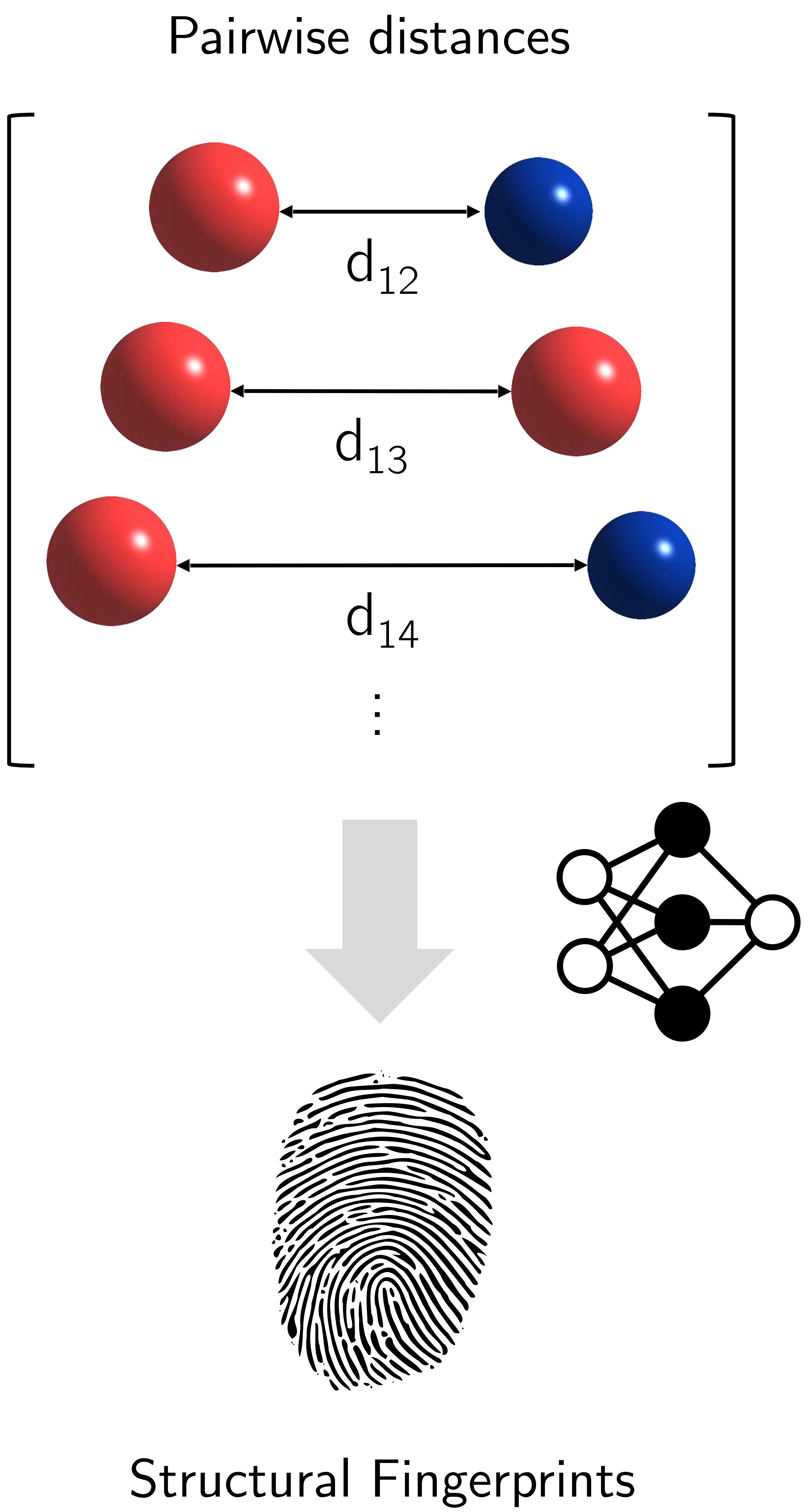
1. Connectivity graphs do not tell us anything about conformers (or PESes)
2. What about materials?

Let's examine how the concept of "graph convolution" work with a solid material



Like before, we analyze each atomic environment and create a fingerprint for them





Pairwise distances

d_{12}

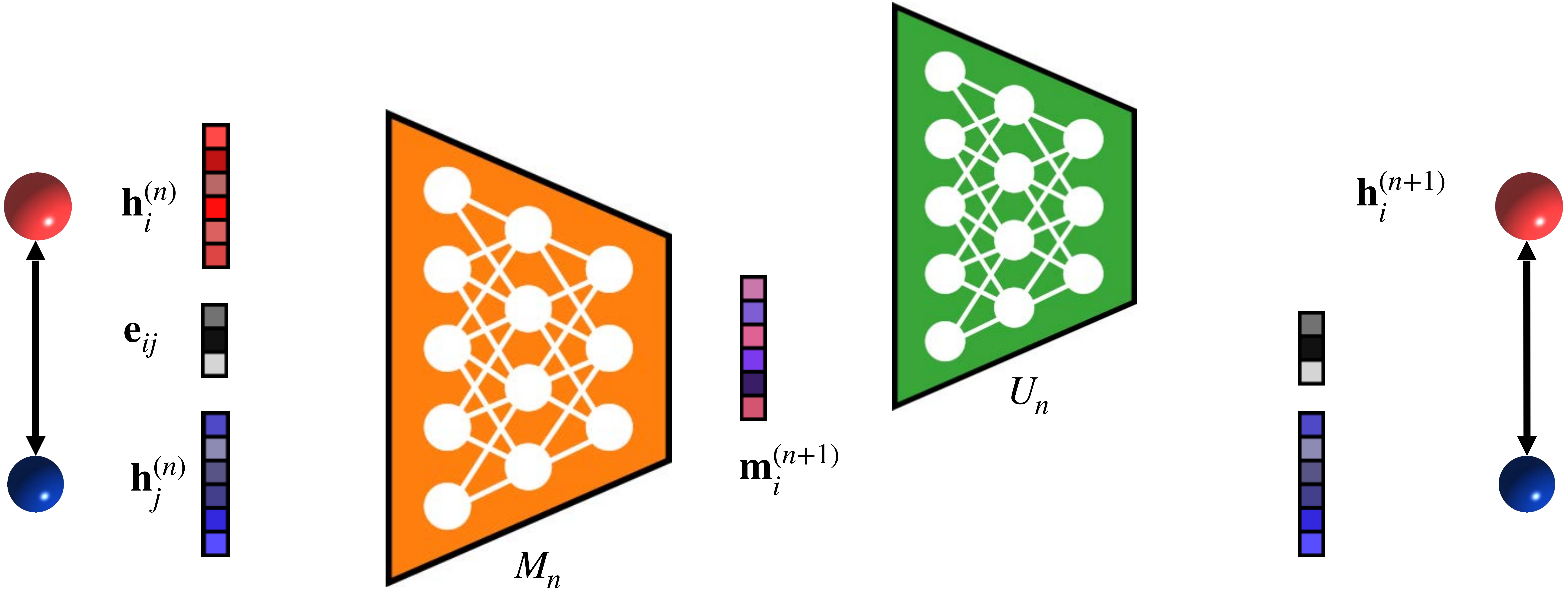
d_{13}

d_{14}

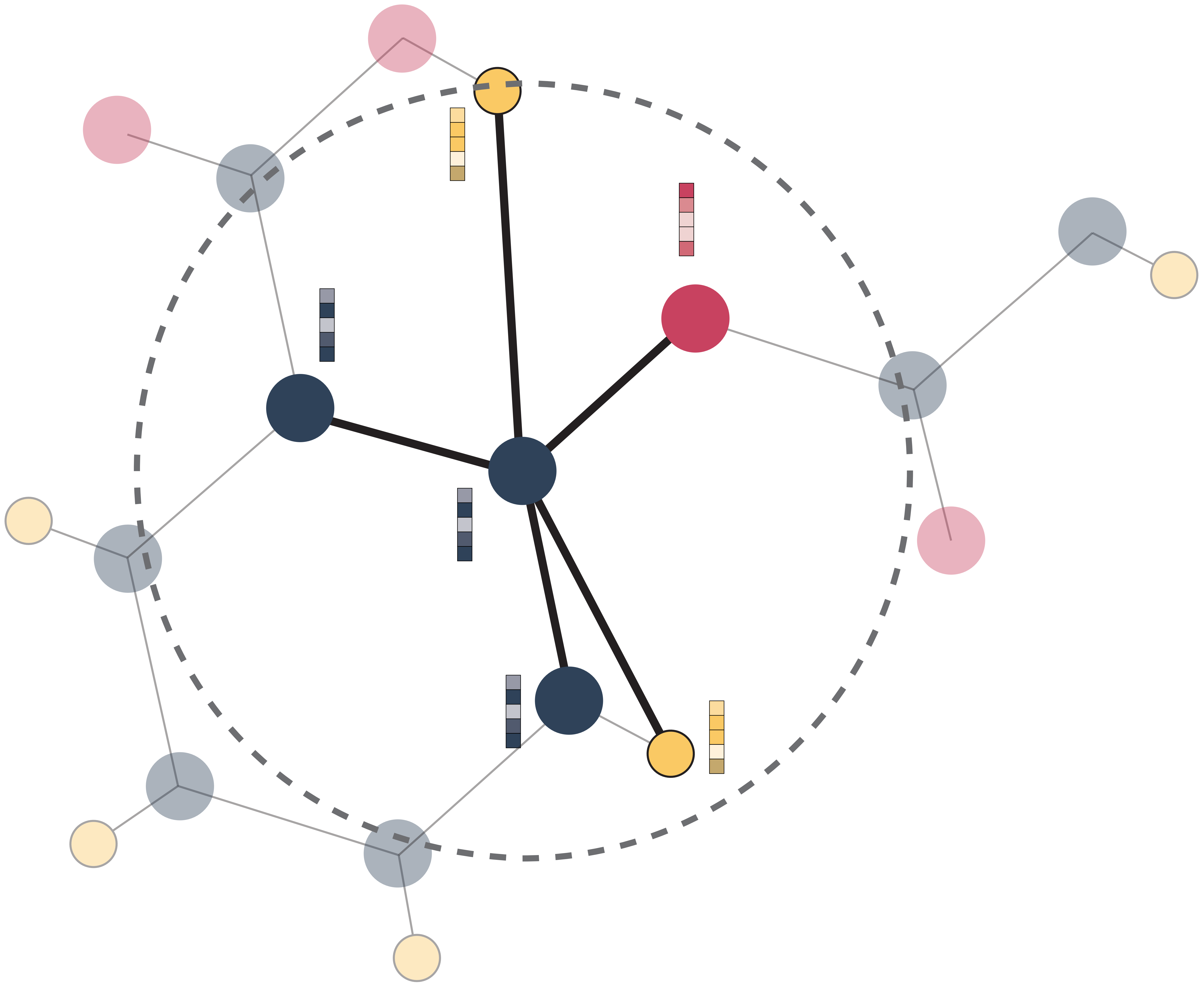
$\vdots$

Structural Fingerprints

Then, we update the representations using a specific neural network architecture



The whole process is called “message passing” framework

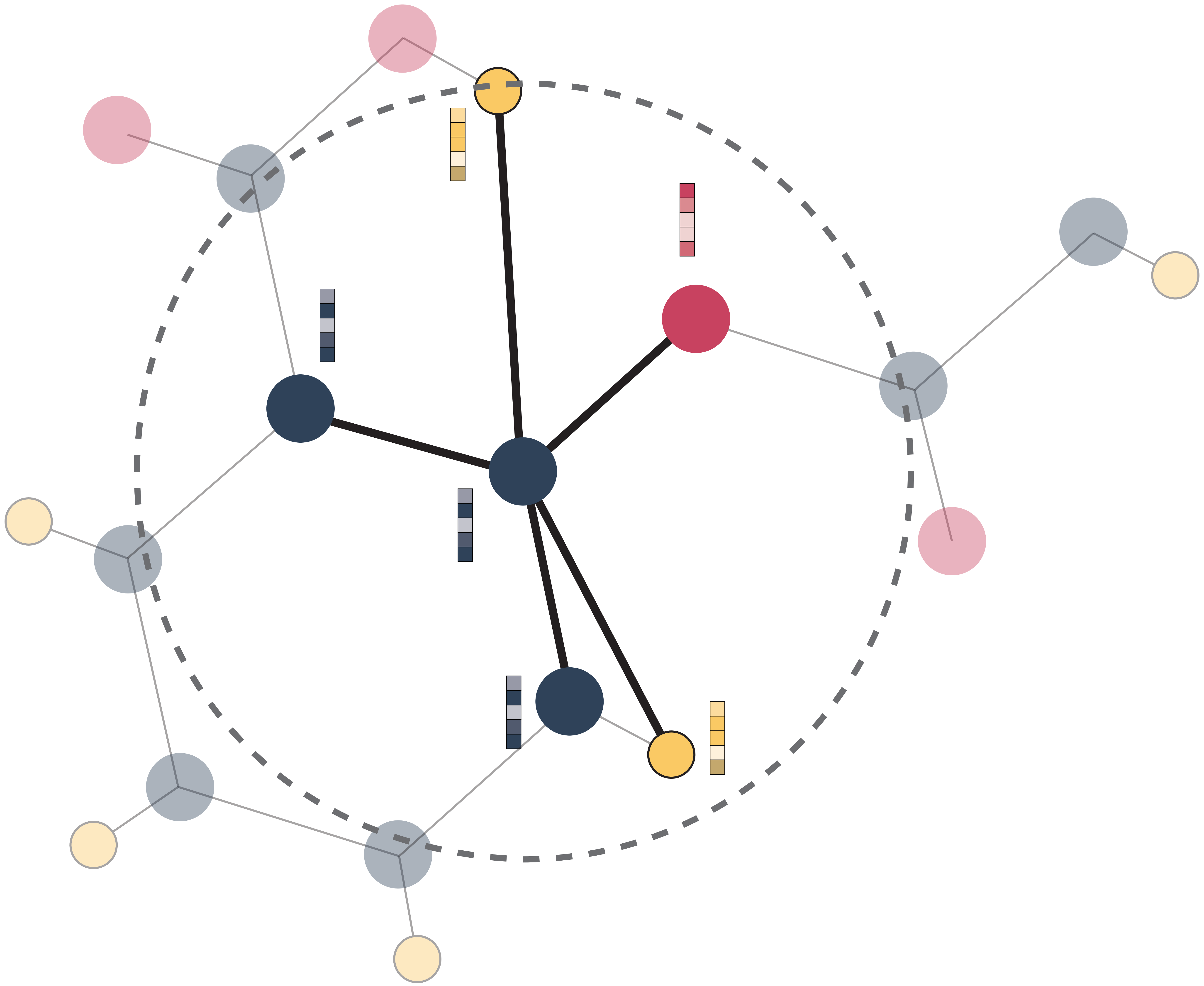


What we are doing is combining the atomic environments in the graph, just like we saw for the CNNs.

The figure on the left shows a central atom and its neighborhood. Atoms which are faded away are not in the neighborhood of this central atom.

A message passing neural network takes the initial graph and representations and creates a node-based representation for each environment.

The architecture of an MPNN



Mathematically, at the layer *n* for the node *i*, the message vector $\boldsymbol{m}_i^{(n+1)}$ is given by

$$\mathbf{m}_i^{(n+1)} = \sum_{j \in \mathcal{N}(i)} M_n\left(\mathbf{h}_i^{(n)}, \mathbf{h}_j^{(n)}, \mathbf{e}_{ij}\right)$$

where $\boldsymbol{h}_i^{(n)}$ is the representation of node *i* at layer *n*, M_n is a neural network, and *N(i)* is the neighborhood of *i*.

The new representation $\boldsymbol{h}_i^{(n+1)}$ is given by

$$\mathbf{h}_i^{(n+1)} = U_n\left(\mathbf{h}_i^{(n)}, \mathbf{m}_i^{(n+1)}\right)$$

where U_n is a neural network.

J. Gilmer et al. *arXiv*:1704.01212 (2017)

This concept of “graph convolution” is very similar to a CNN

CNN on image

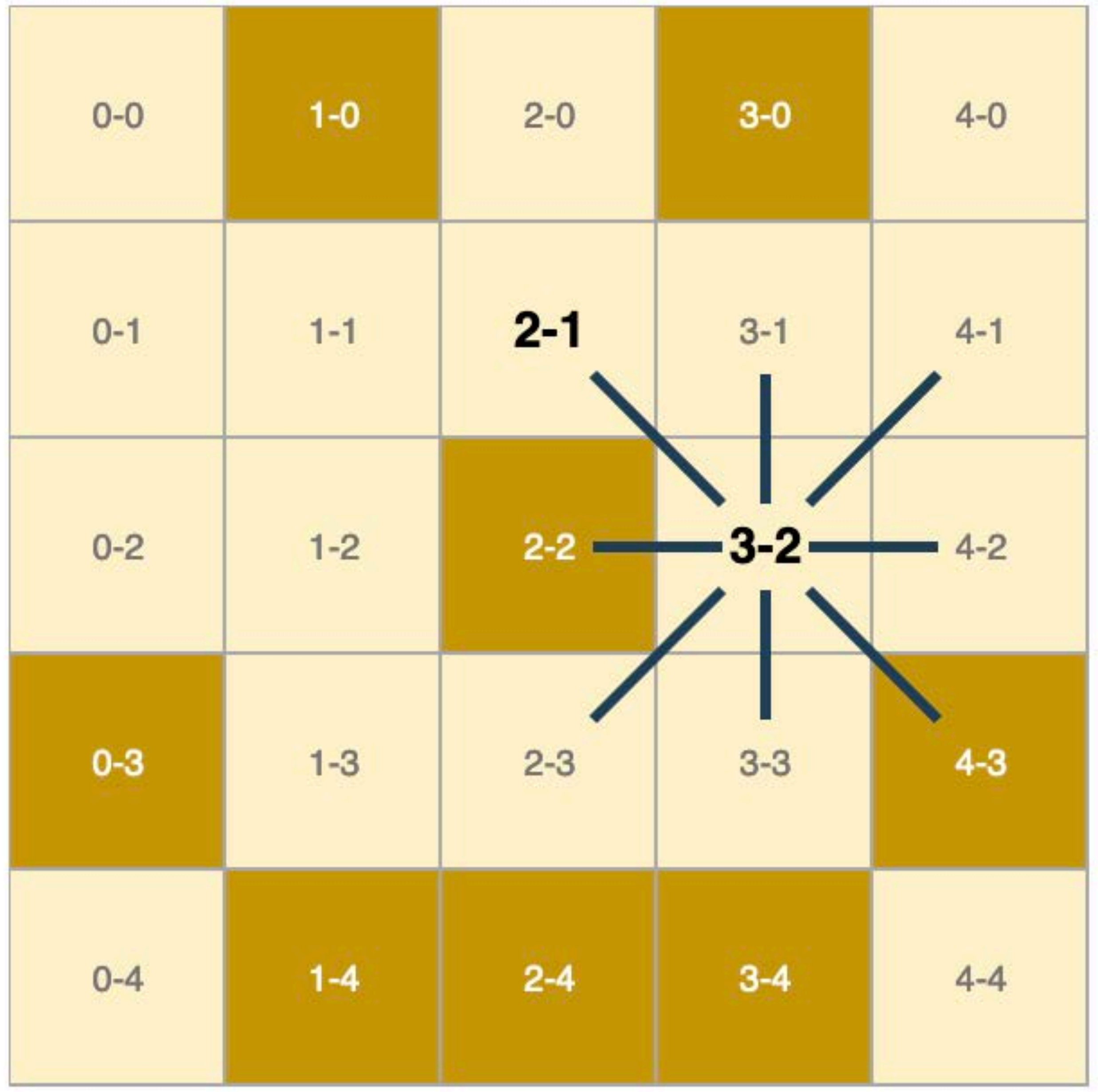
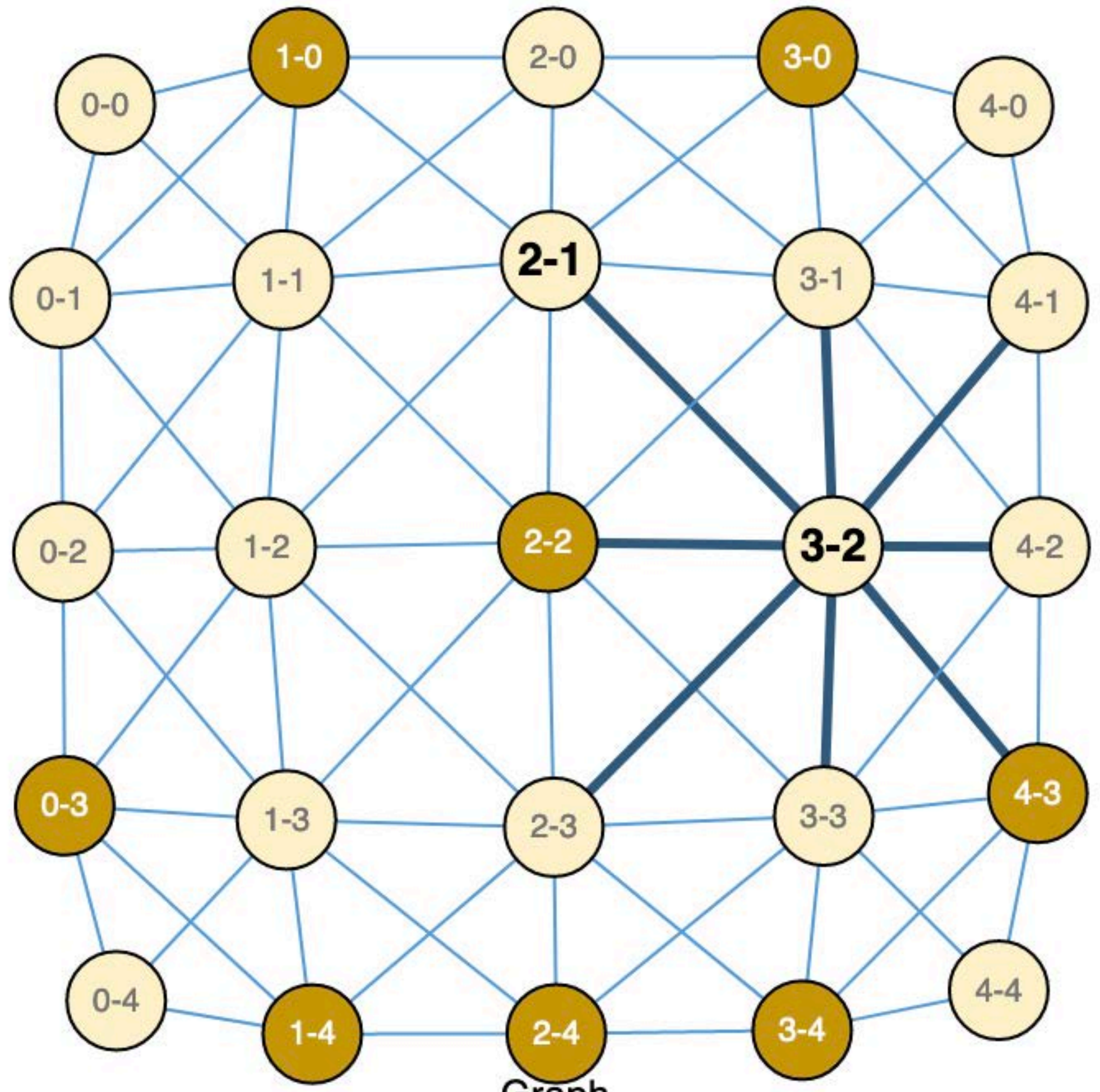


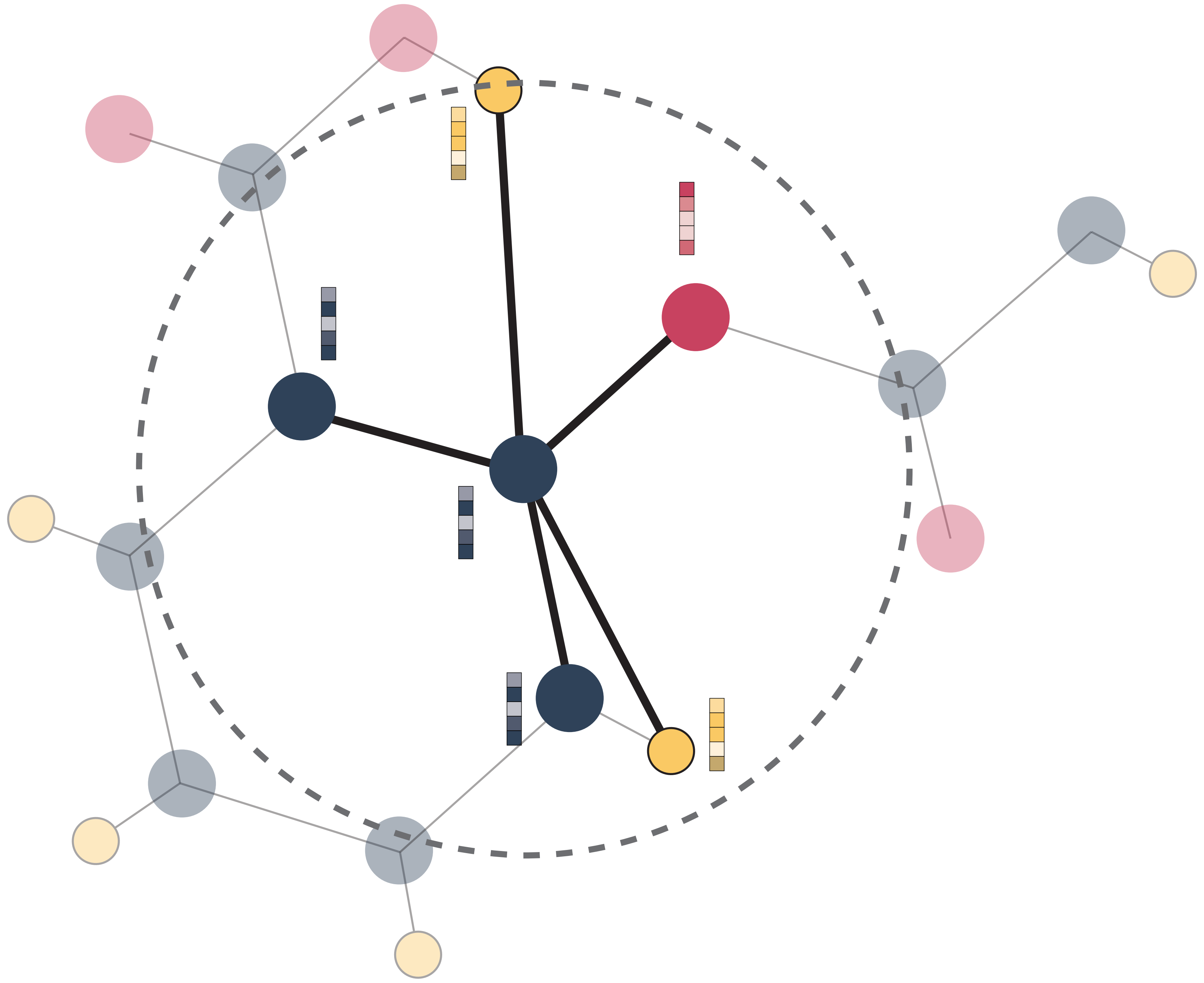
Image Pixels

Graph convolution on image



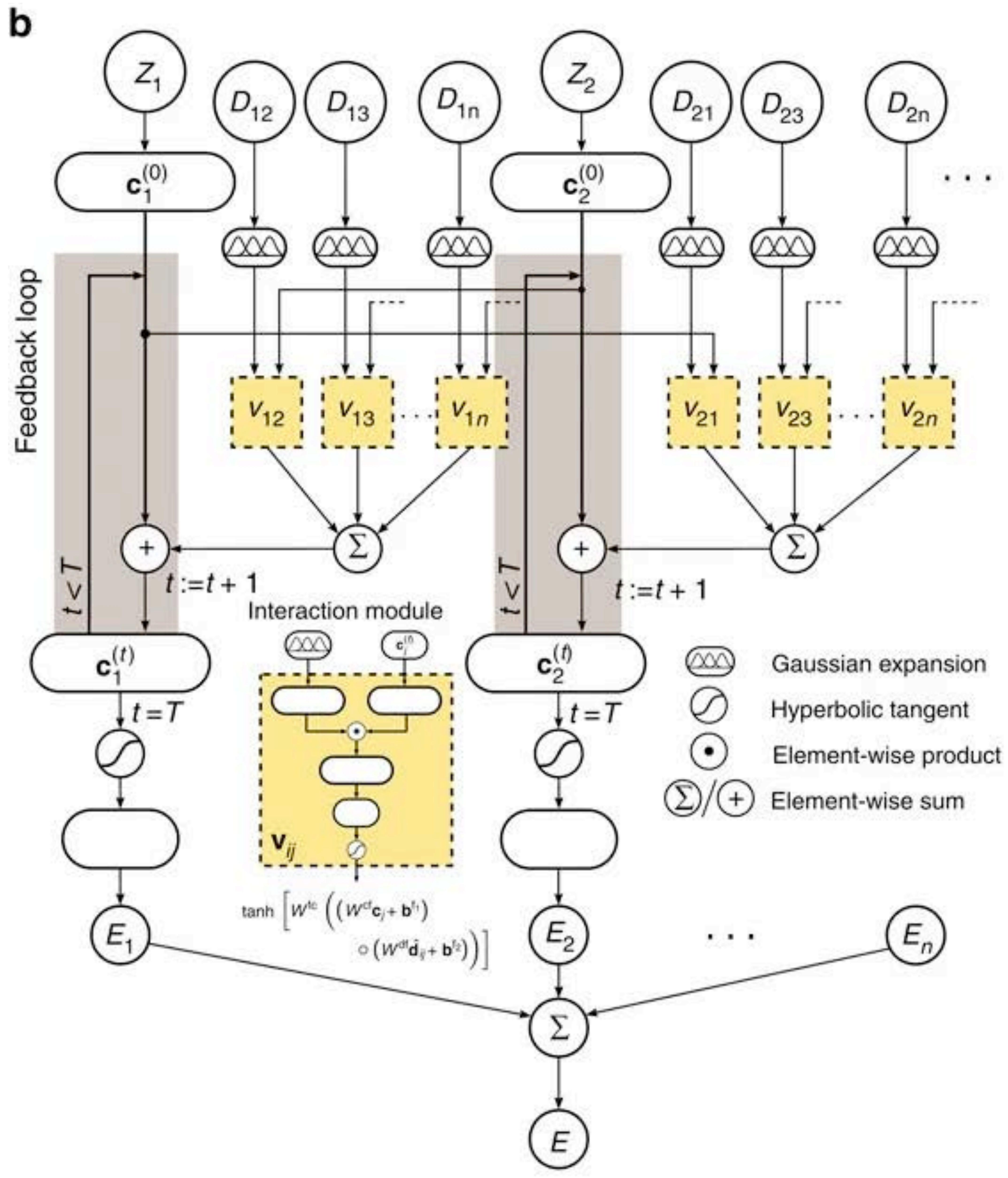
Graph

Graph convolution on geometry

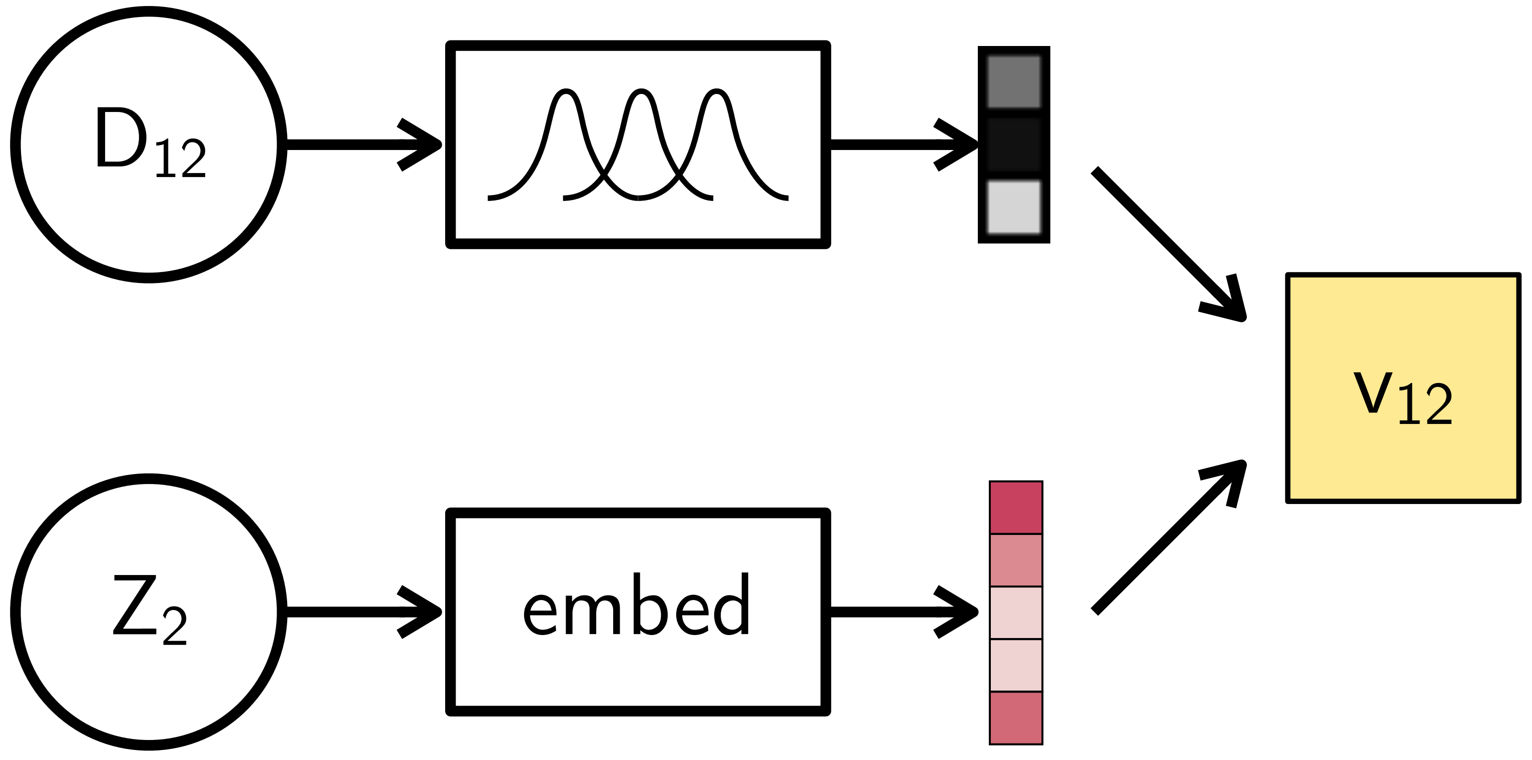


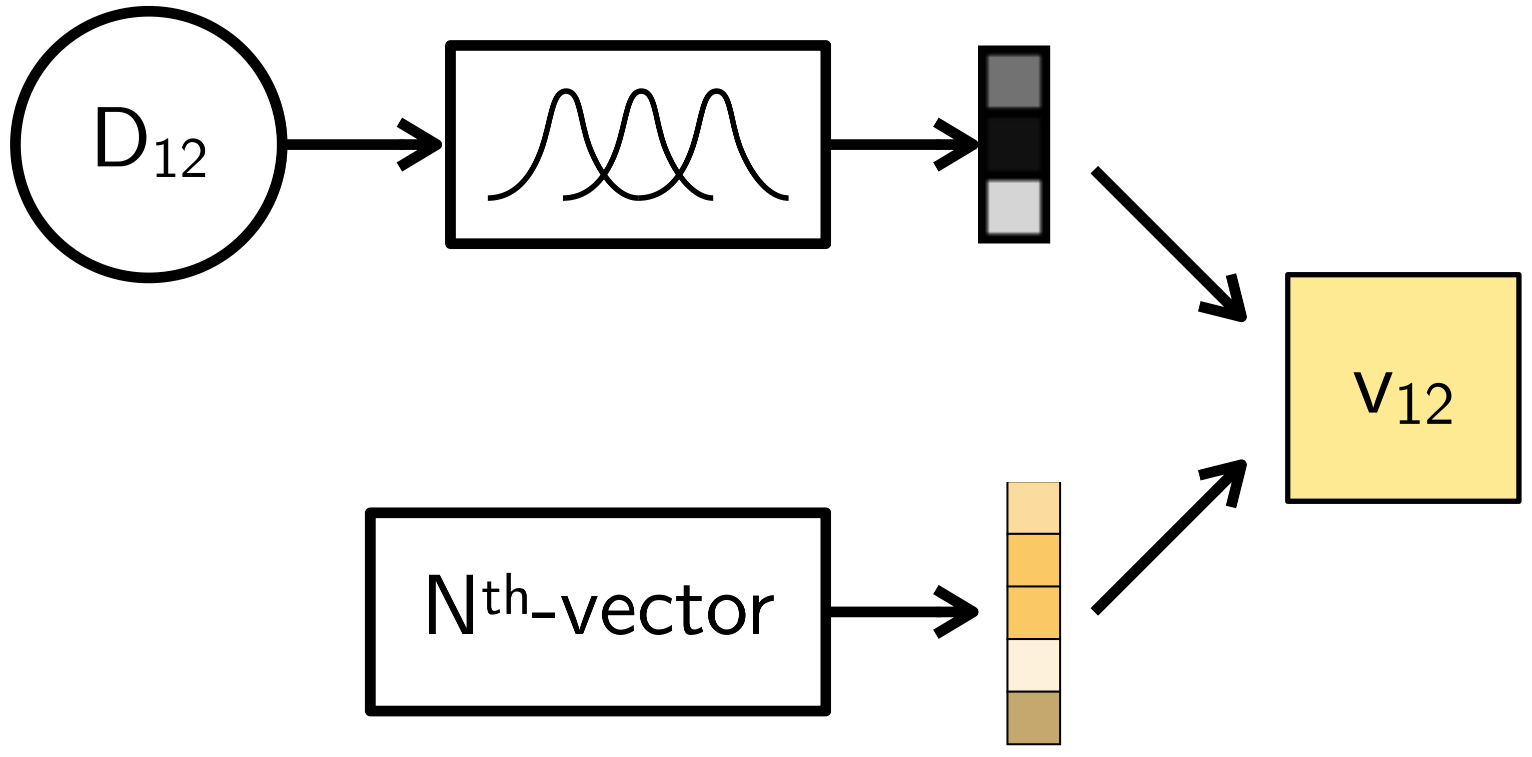
CNN images: B. Sanchez-Lengeling et al. *Distill.pub* (2021)

Application to NNIP: the Deep Tensor Neural Network (DTNN)



The architecture looks complicated! Let's break it down:

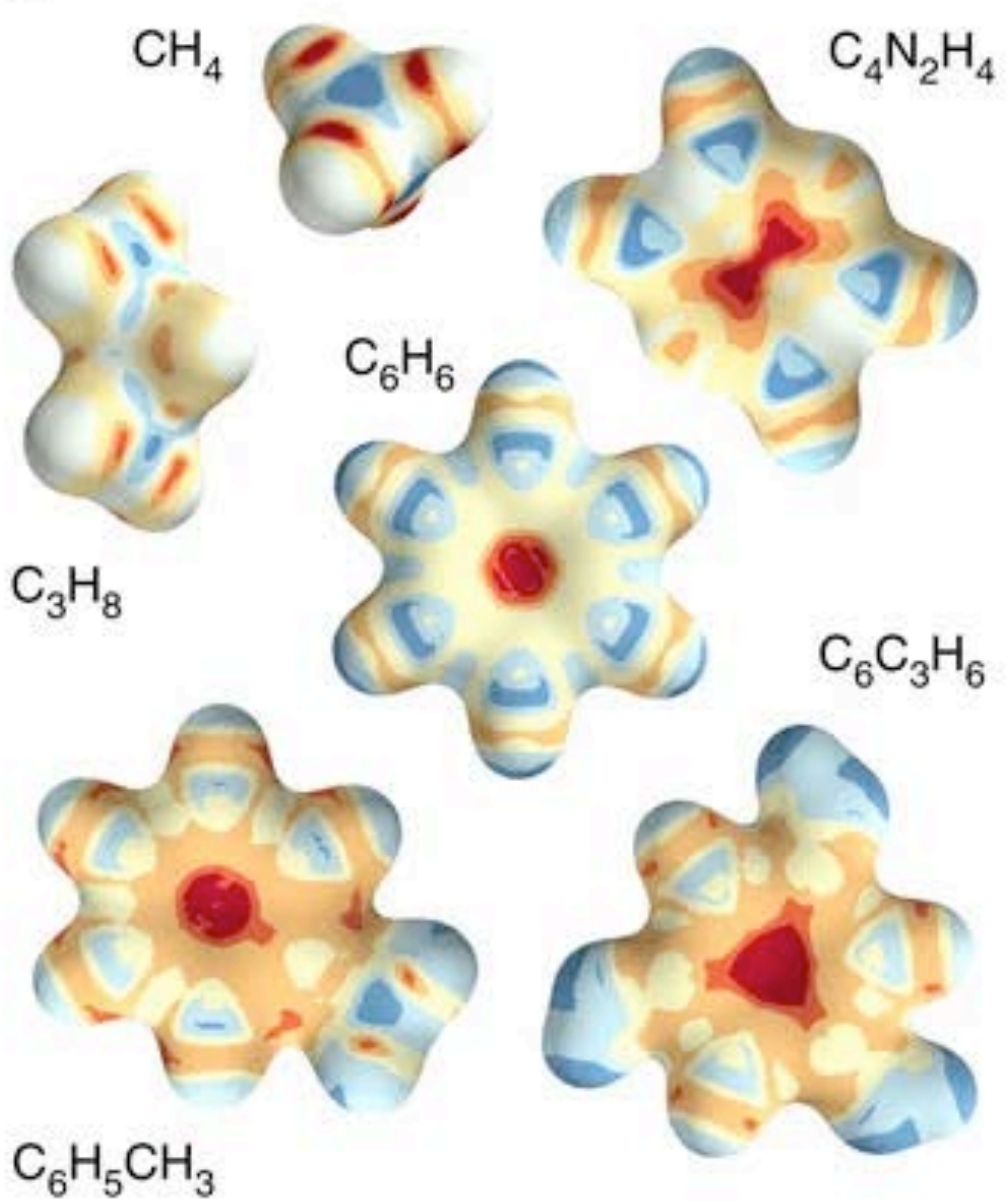




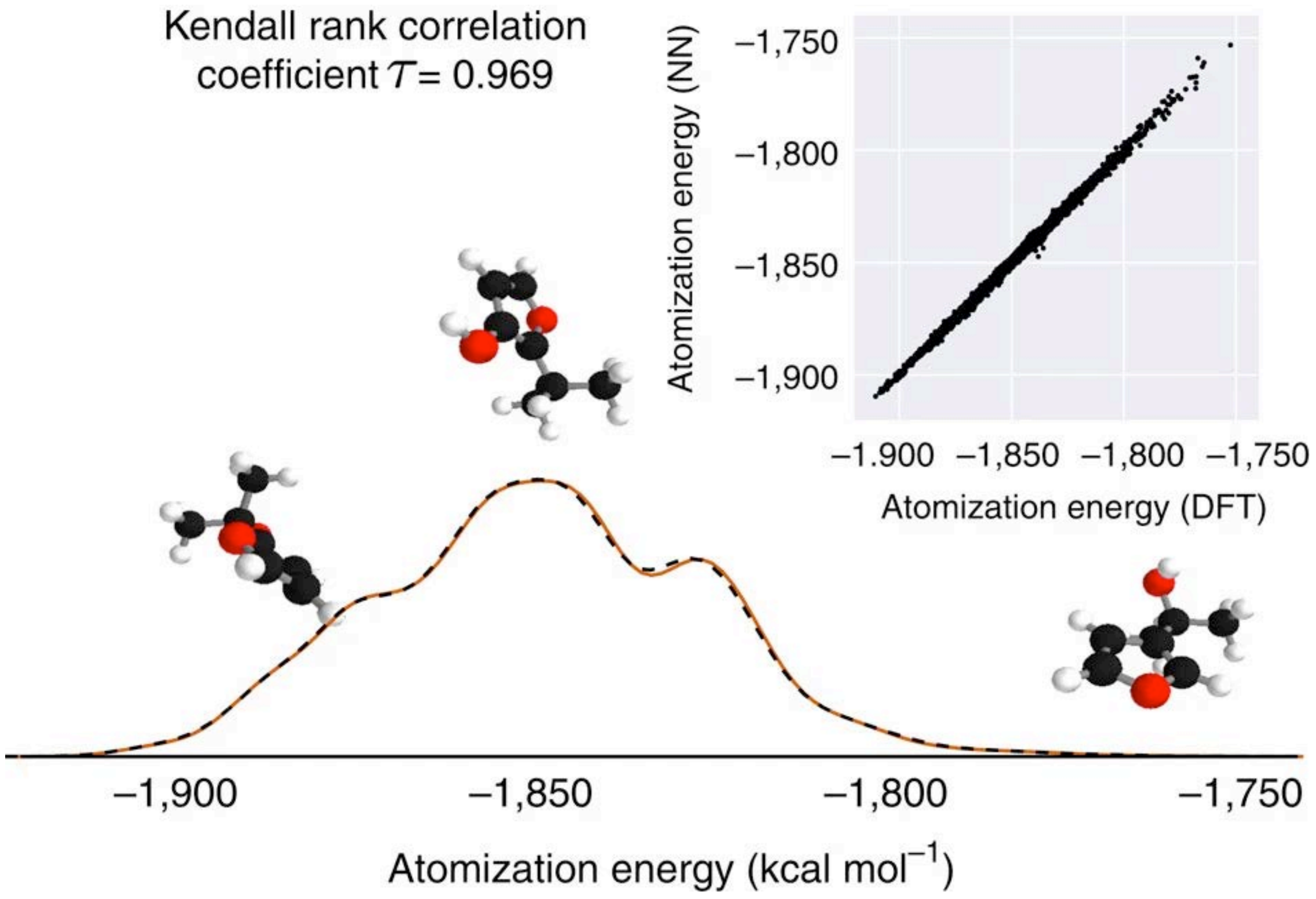
K. Schütt et al. *Nat. Commun.* **8**, 13890 (2017)

DTNN showed interpretable filters and excellent prediction of molecular properties

Local contribution of a test charge (probing the NN):



Different isomers of $C_7O_2H_{10}$



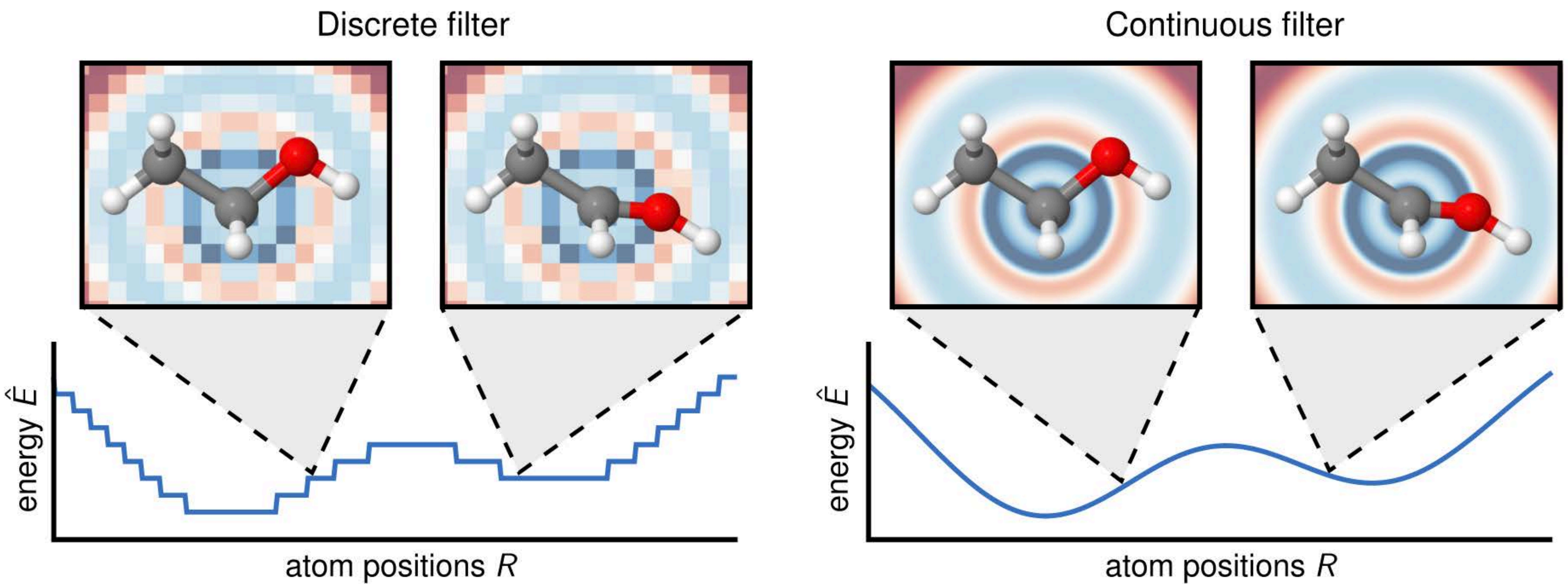
K. Schütt et al. *Nat. Commun.* **8**, 13890 (2017)

The limitations of the previous models for performing simulations at T > 0 K

So far, the graph-based NNs have treated molecular graphs or ground-state properties. What if we wanted to perform MD simulations?

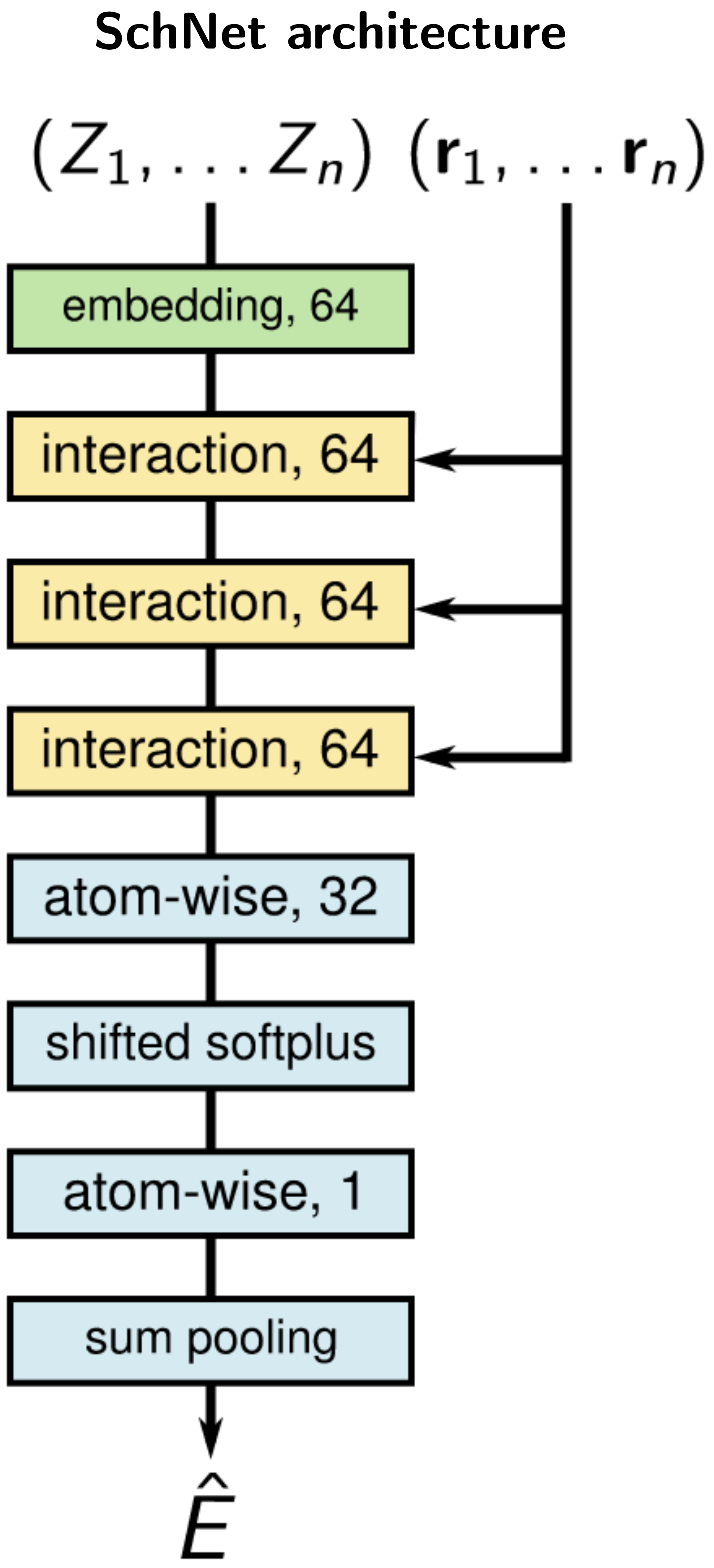
$$\longrightarrow \qquad \mathbf{F} = -\nabla_{\mathbf{r}} E$$

The problem with the previous NN architectures is that they mostly predict properties of a static graph or 3D structure. If the atoms move, it is not guaranteed to vary the energy continuously.



K. Schütt et al. *NeurIPS* **30** (2017)

SchNet as a continuous-filter NNIP

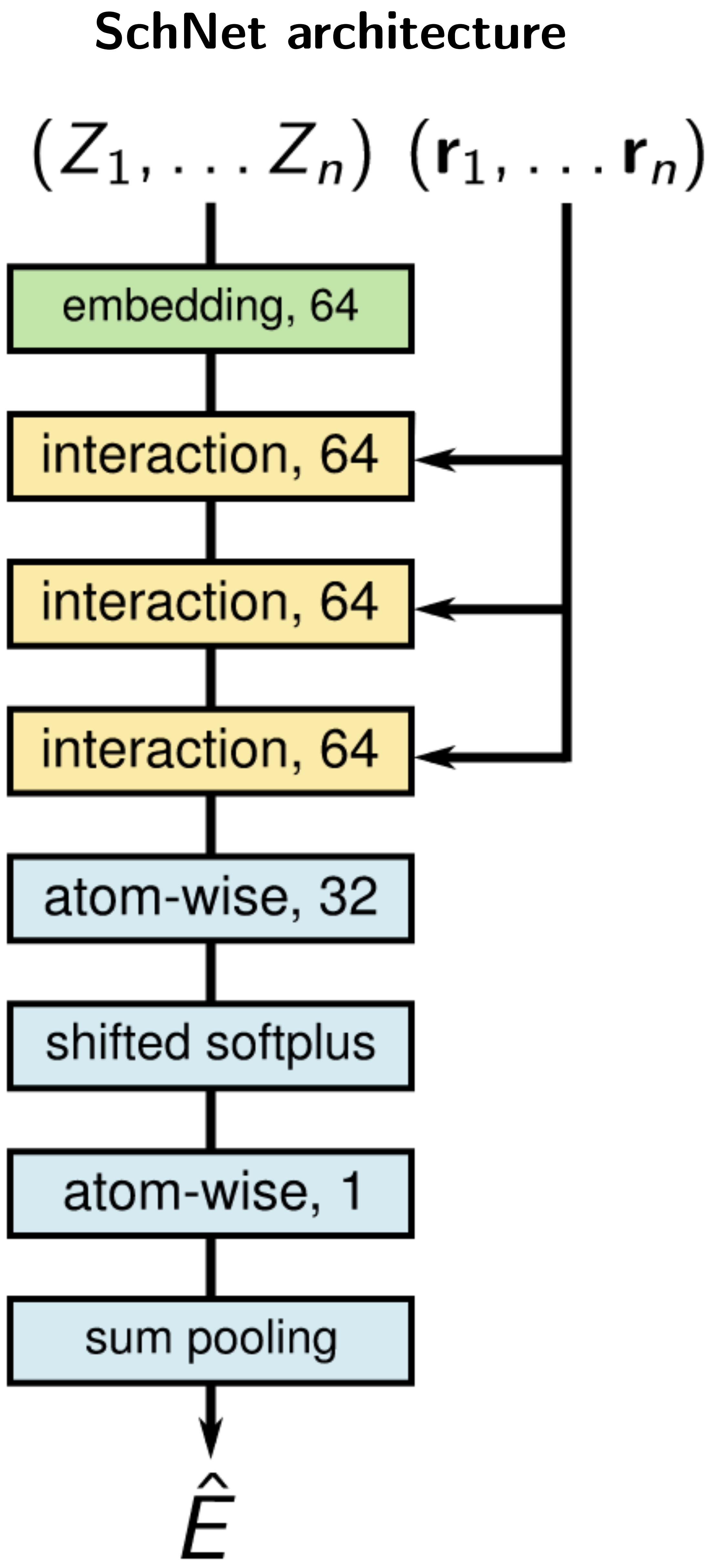


The SchNet architecture is not too different from what we learned. On the image on the left, we can see:

1. embedding layers, that map an atomic number to a vector.
2. interaction blocks for representation learning with message passing.
3. fully connected NNs, for predicting atom-wise energies (or properties).
4. a sum at the end, pooling all the contributions from the atoms of the systems.

K. Schütt et al. *NeurIPS* **30** (2017)

How to fit to energies and forces?



The trick to training this NN is to use information not only about the energy of the PES, but also the forces:

$$\mathscr{L} = \lambda_E \| E - \hat{E} \|^2 + \lambda_F \frac{1}{n} \sum_{i=0}^{n} \| \mathbf{F}_i - \hat{\mathbf{F}}_i \|^2$$

Where the predicted forces can be obtained by differentiating the predicting energy with respect to the input coordinates:

$$\hat{\mathbf{F}}_i = - \frac{\partial \hat{E}}{\partial \mathbf{R}_i}$$

K. Schütt et al. *NeurIPS* **30** (2017)

How does it perform?

Predictions of energies + forces is better than just energies.

Why do you think this is the case?

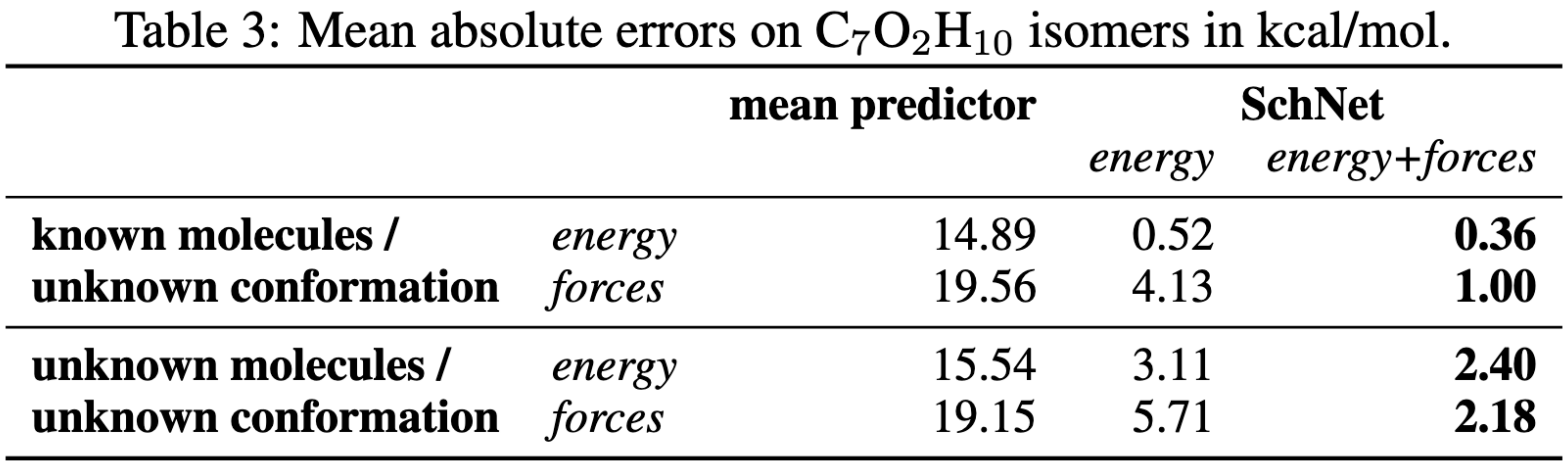
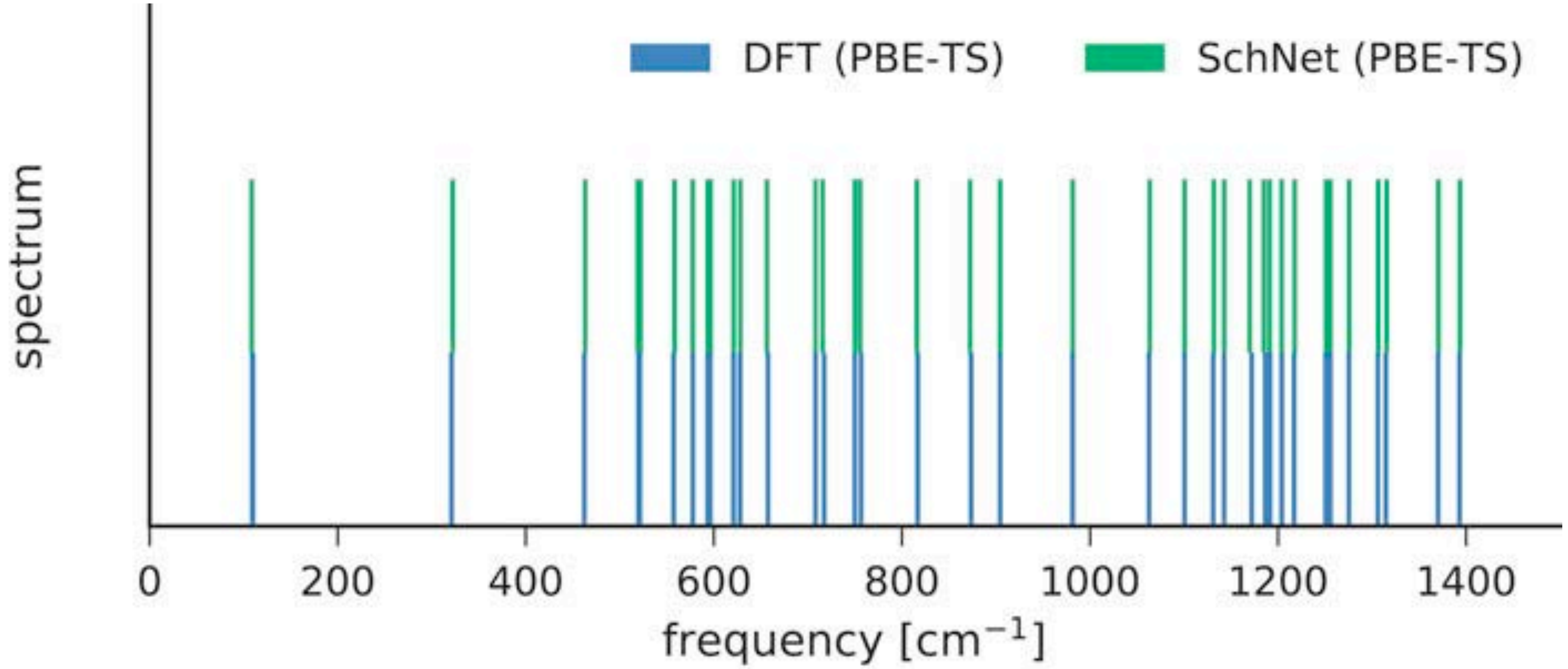


Table 3: Mean absolute errors on $C_7O_2H_{10}$ isomers in kcal/mol.

		mean predictor	SchNet	
			energy	*energy+forces*
known molecules / unknown conformation	*energy*	14.89	0.52	**0.36**
	forces	19.56	4.13	**1.00**
unknown molecules / unknown conformation	*energy*	15.54	3.11	**2.40**
	forces	19.15	5.71	**2.18**

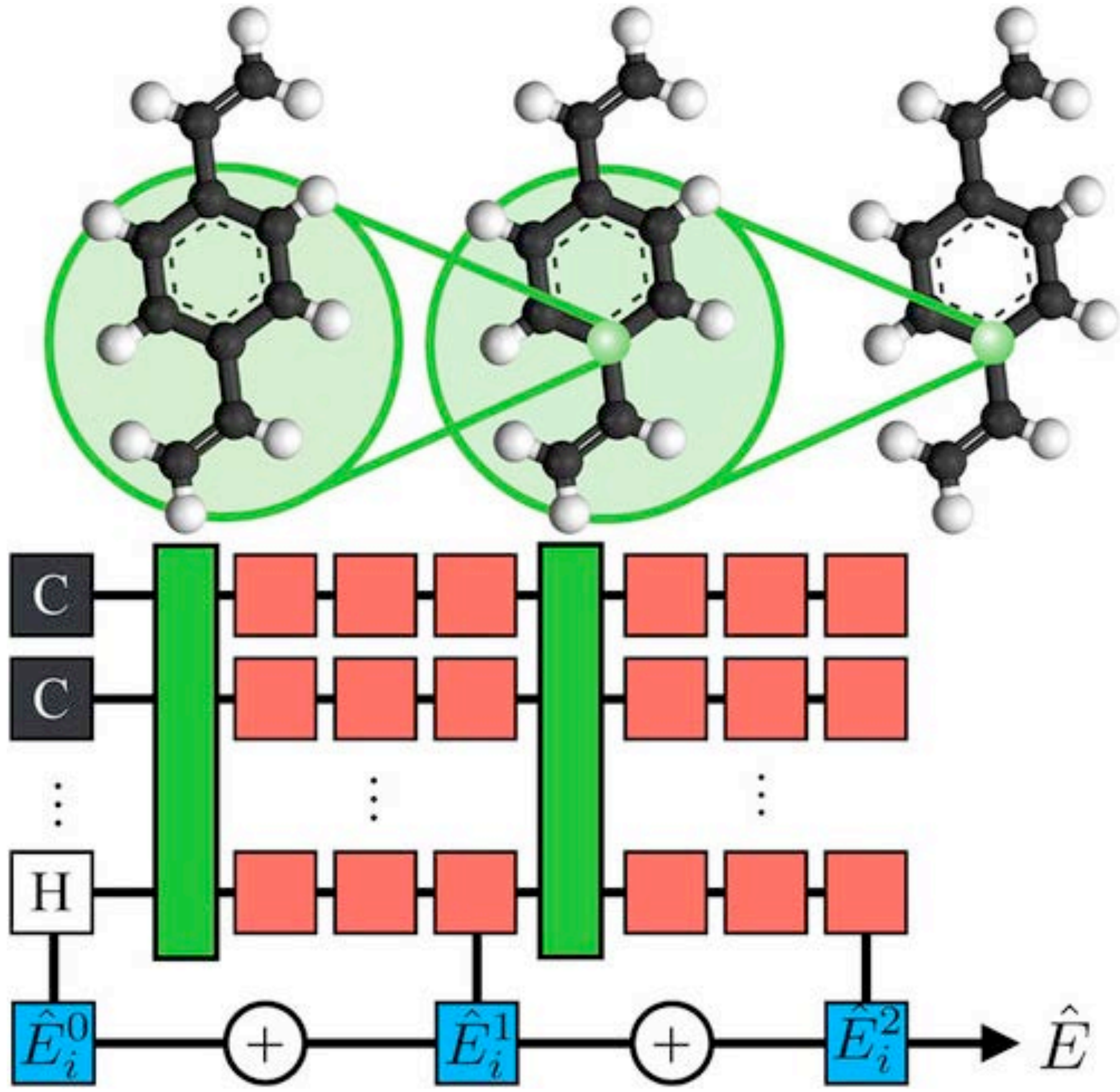
Predictions of forces enables other properties to be obtained, such as vibrational spectra.



K. Schütt et al. *NeurIPS* **30** (2017)
K. Schütt et al. *J. Chem. Phys.* ***148***, 241722 (2018)

Many more models were proposed in the field

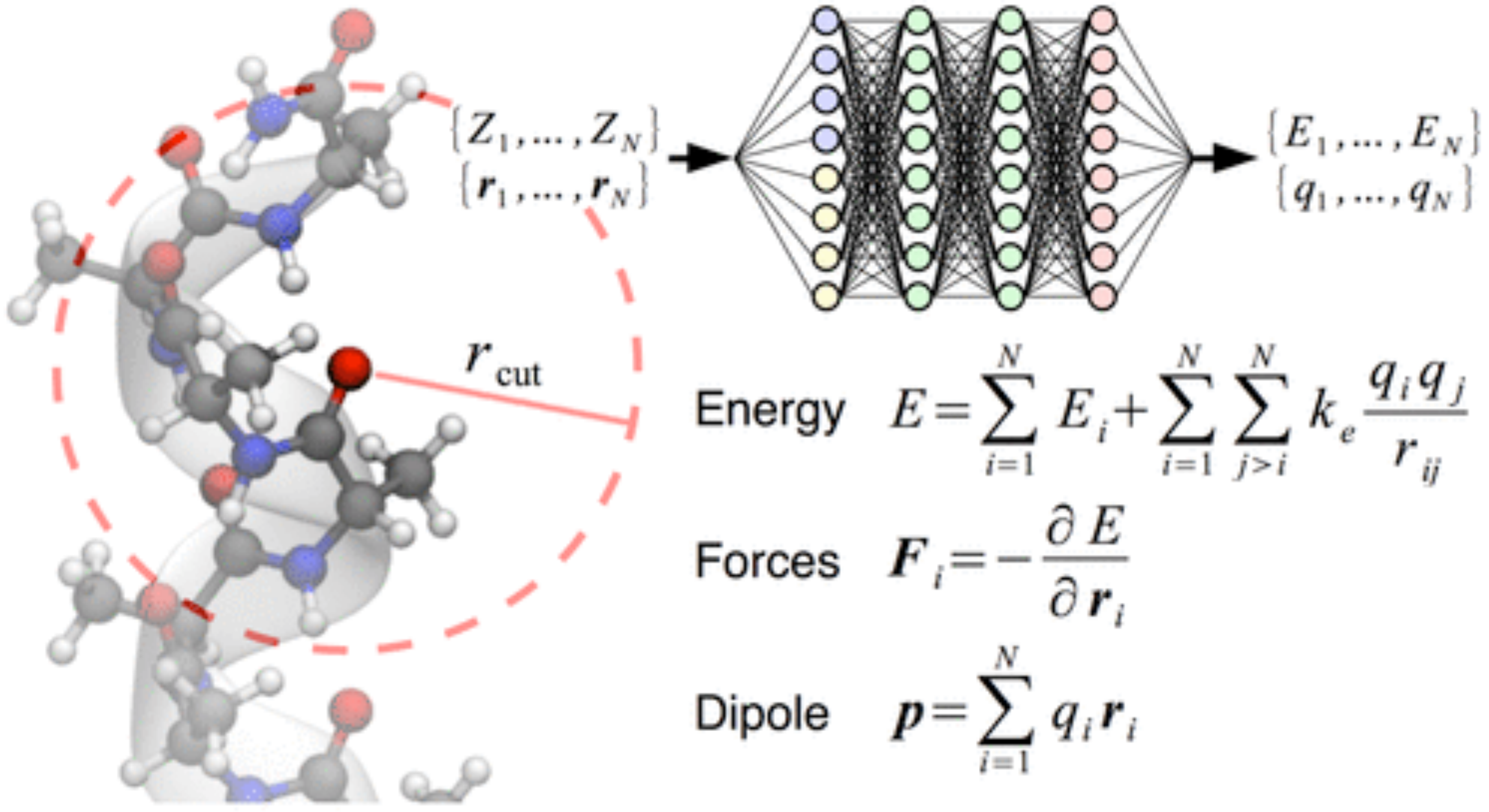
HIP-NN



N. Lubbers et al. *J. Chem. Phys* **148**, 241715 (2018)

Architecture inspired in many-body expansions

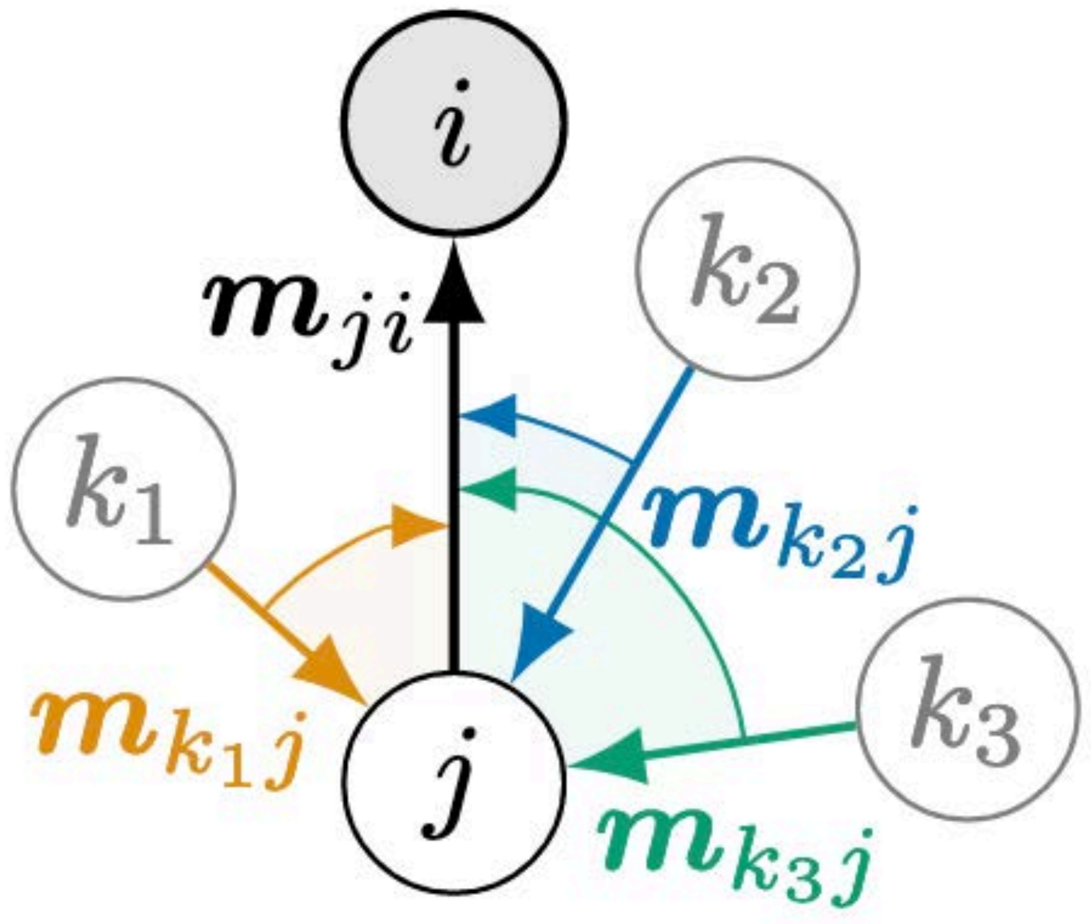
PhysNet


r_{cut}
$\{Z_1,\dots,Z_N\}$
$\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_N\}$
$\{E_1,\dots,E_N\}$
$\{q_1,\dots,q_N\}$
Energy $E=\sum_{i=1}^{N}E_i+\sum_{i=1}^{N}\sum_{j>i}^{N}k_e\frac{q_iq_j}{r_{ij}}$
Forces $\boldsymbol{F}_i=-\frac{\partial E}{\partial \boldsymbol{r}_i}$
Dipole $\boldsymbol{p}=\sum_{i=1}^{N}q_i\boldsymbol{r}_i$

O. Unke and M. Meuwly. *JCTC* **15** (6), 3678 (2019)

Prediction of atomic charges

DimeNet



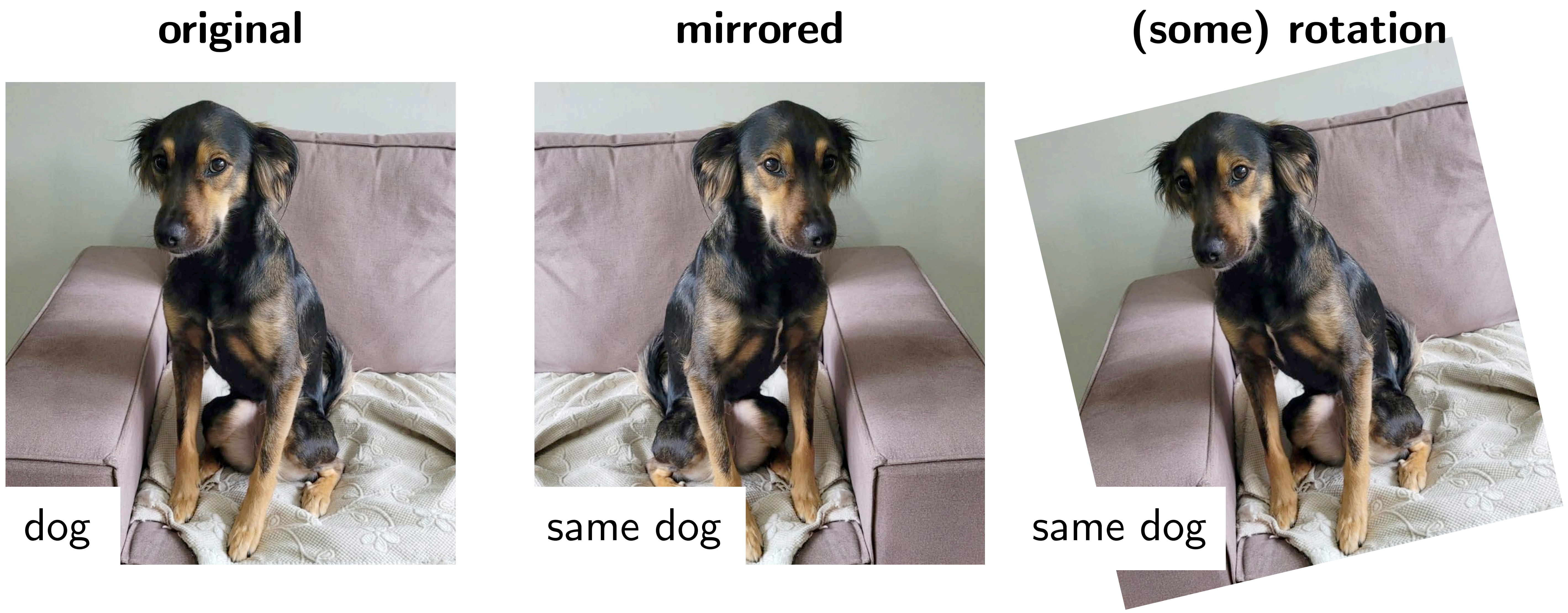
J. Gasteiger et al. *ICLR* (2020), *arXiv:2003.03123*

Explicit treatment of three-body terms

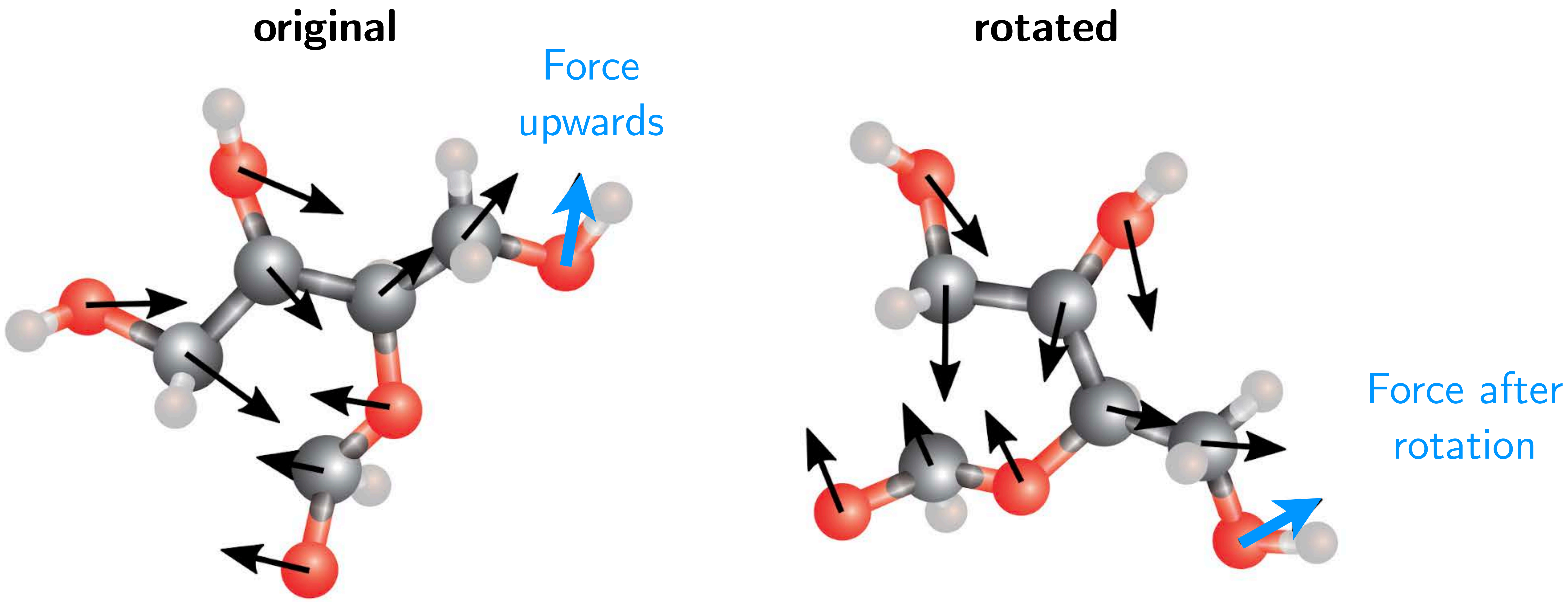
There are many more models in the field nowadays. What is being improved?

First improvement: symmetry, invariance, and equivariance

Images are invariant to translation, mirror (often), and rotation (sometimes).



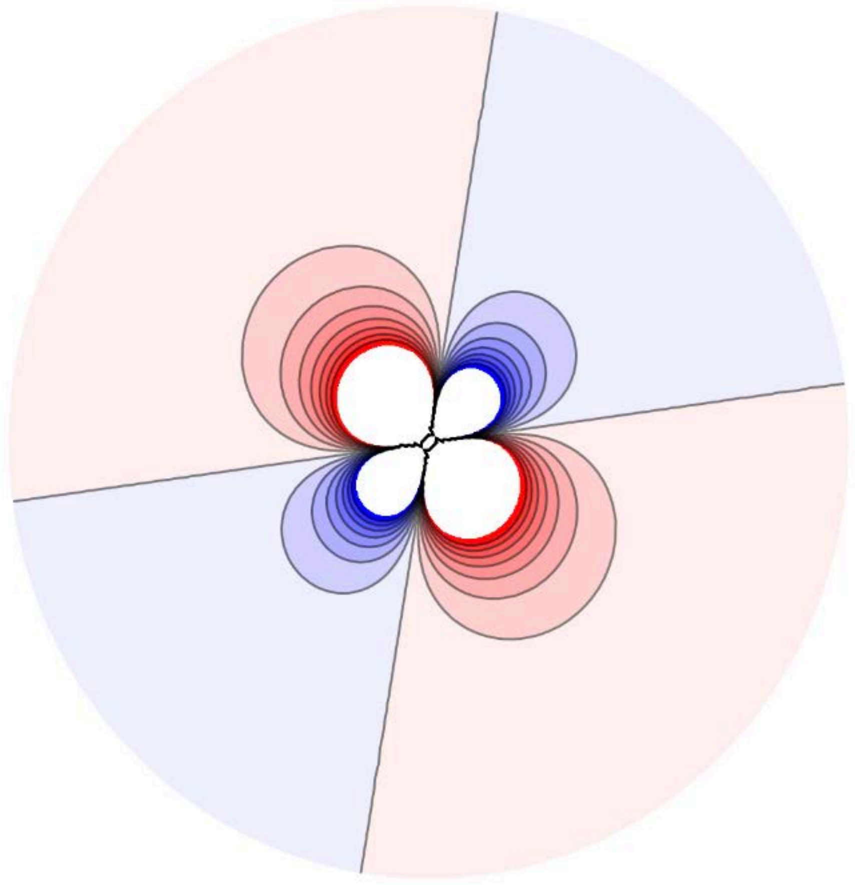
Forces are equivariant to rotation: they transform according to the operation



Molecule figure from: T. Smidt, e3nn (2021). https://e3nn.org

Recent improvements in this area

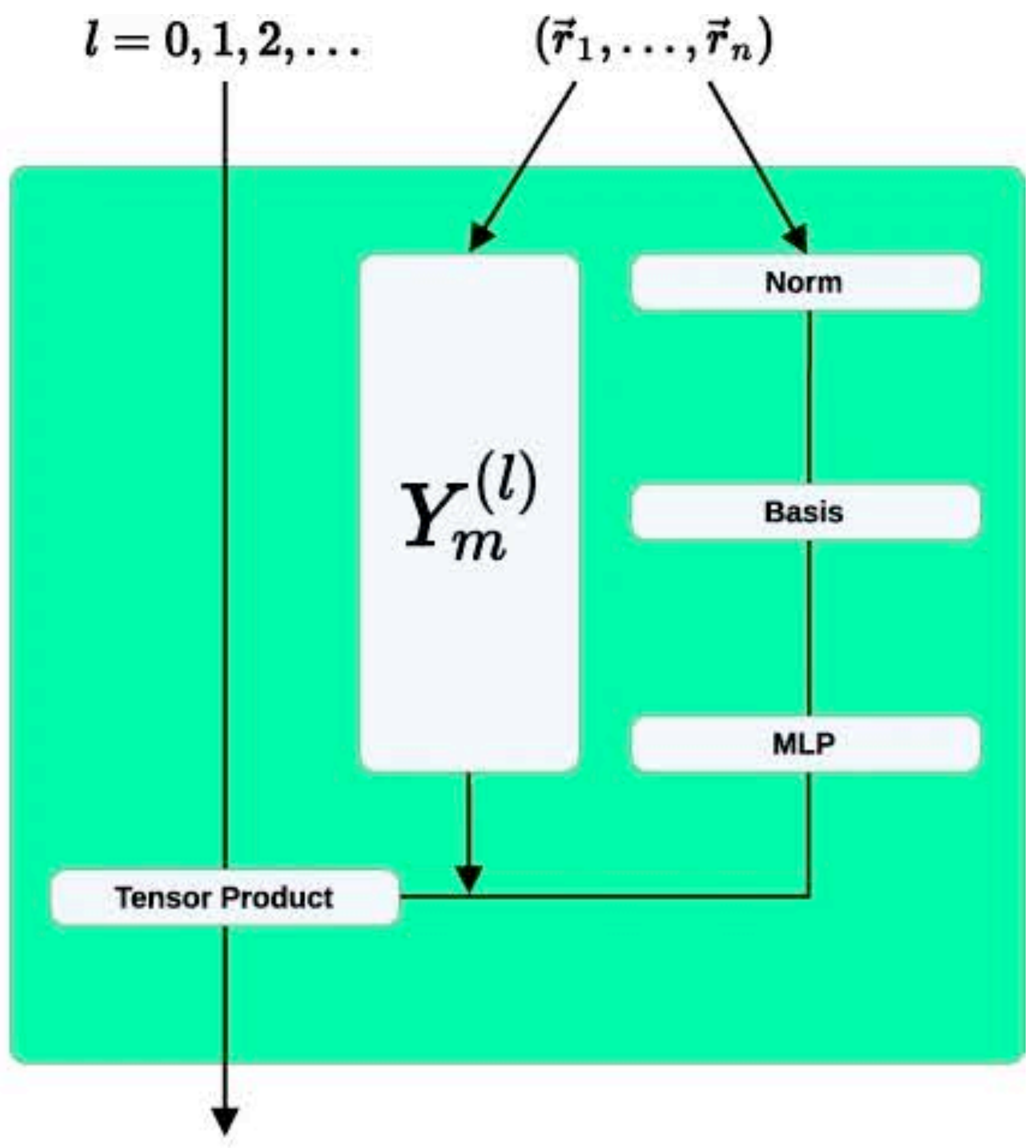
Cormorant



B. Anderson et al. *arXiv:1906.04015* (2019)

Covariant "neurons" for SO(3) symmetry

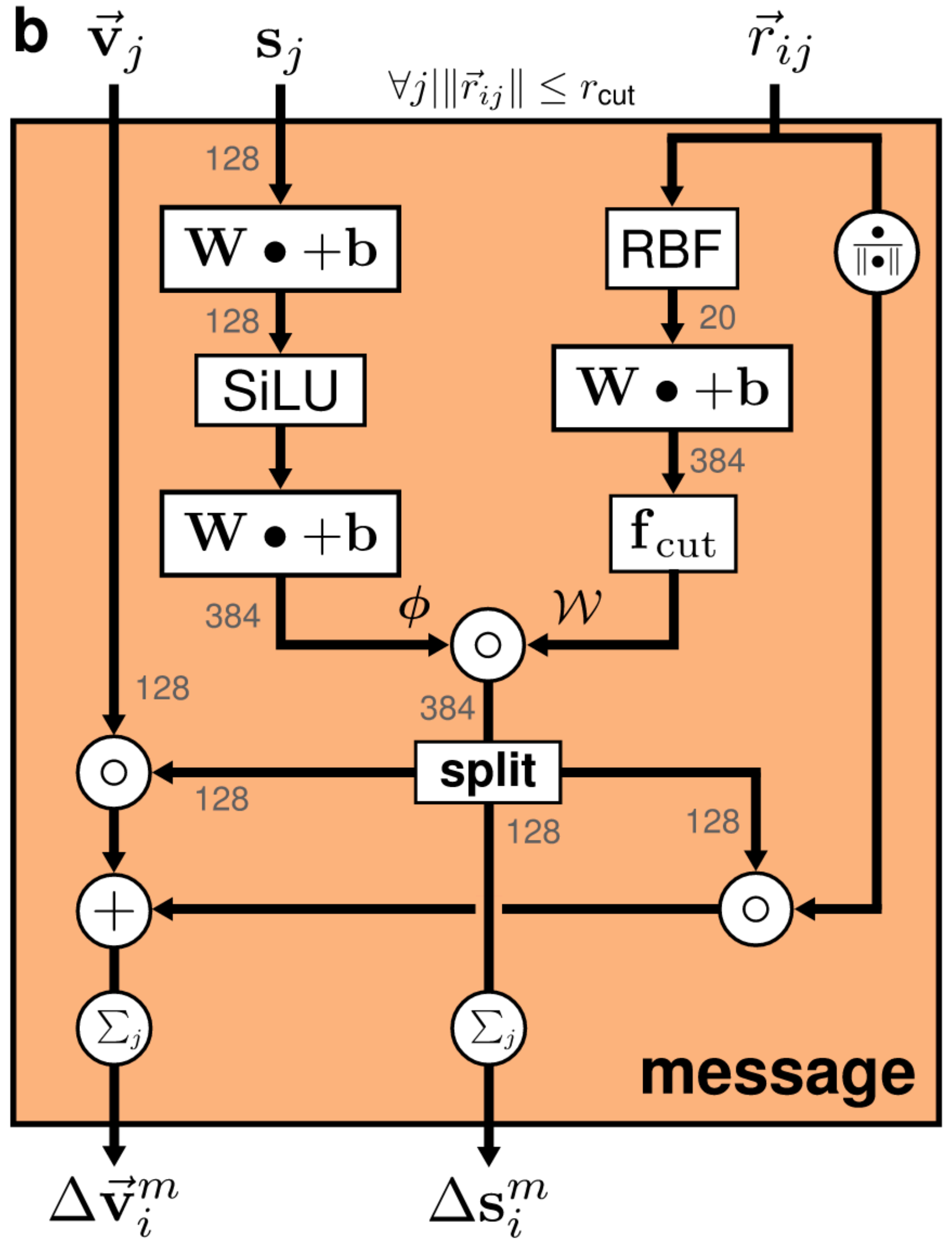
NequIP



S. Batzner et al. *Nat. Commun.* **13**, 2453 (2022)

E(3)-equivariant GNN

PaiNN


b
$\vec{\mathbf{v}}_j$
$\mathbf{s}_j$
$\vec{r}_{ij}$
$\forall j | \|\vec{r}_{ij}\| \le r_{\text{cut}}$
128
$\mathbf{W} \bullet + \mathbf{b}$
128
SiLU
$\mathbf{W} \bullet + \mathbf{b}$
384
RBF
20
$\mathbf{W} \bullet + \mathbf{b}$
384
$\mathbf{f}_{\text{cut}}$
ϕ
$\mathcal{W}$
$\circ$
128
384
split
128
128
128
$\circ$
$+$
$\circ$
Σ_j
Σ_j
message
$\Delta\vec{\mathbf{v}}_i^m$
$\Delta\mathbf{s}_i^m$

K. Schütt et al. *ICML* **139** (2021)

Uses directional message-passing and vector repr.

These models are close to the state-of-the-art for several datasets

Other improvements: many-body terms and better scaling

Allegro

A. Musaelian et al. *arXiv:2204.05249 (2022)*

$$
\begin{aligned}
\mathbf{V}^{ij,L}_{n,(\ell_1,p_1,\ell_2,p_2)\to(\ell_{\text{out}},p_{\text{out}})} &= \sum_{k\in\mathcal{N}(i)} w^{ik,L}_{n,\ell_2,p_2}\left(\mathbf{V}^{ij,L-1}_{n,\ell_1,p_1}\otimes\vec{Y}^{ik}_{\ell_2,p_2}\right)\\
&= \sum_{k\in\mathcal{N}(i)} \mathbf{V}^{ij,L-1}_{n,\ell_1,p_1}\otimes\left(w^{ik,L}_{n,\ell_2,p_2}\vec{Y}^{ik}_{\ell_2,p_2}\right)\\
&= \mathbf{V}^{ij,L-1}_{n,\ell_1,p_1}\otimes\left(\sum_{k\in\mathcal{N}(i)} w^{ik,L}_{n,\ell_2,p_2}\vec{Y}^{ik}_{\ell_2,p_2}\right)
\end{aligned}
$$

MACE

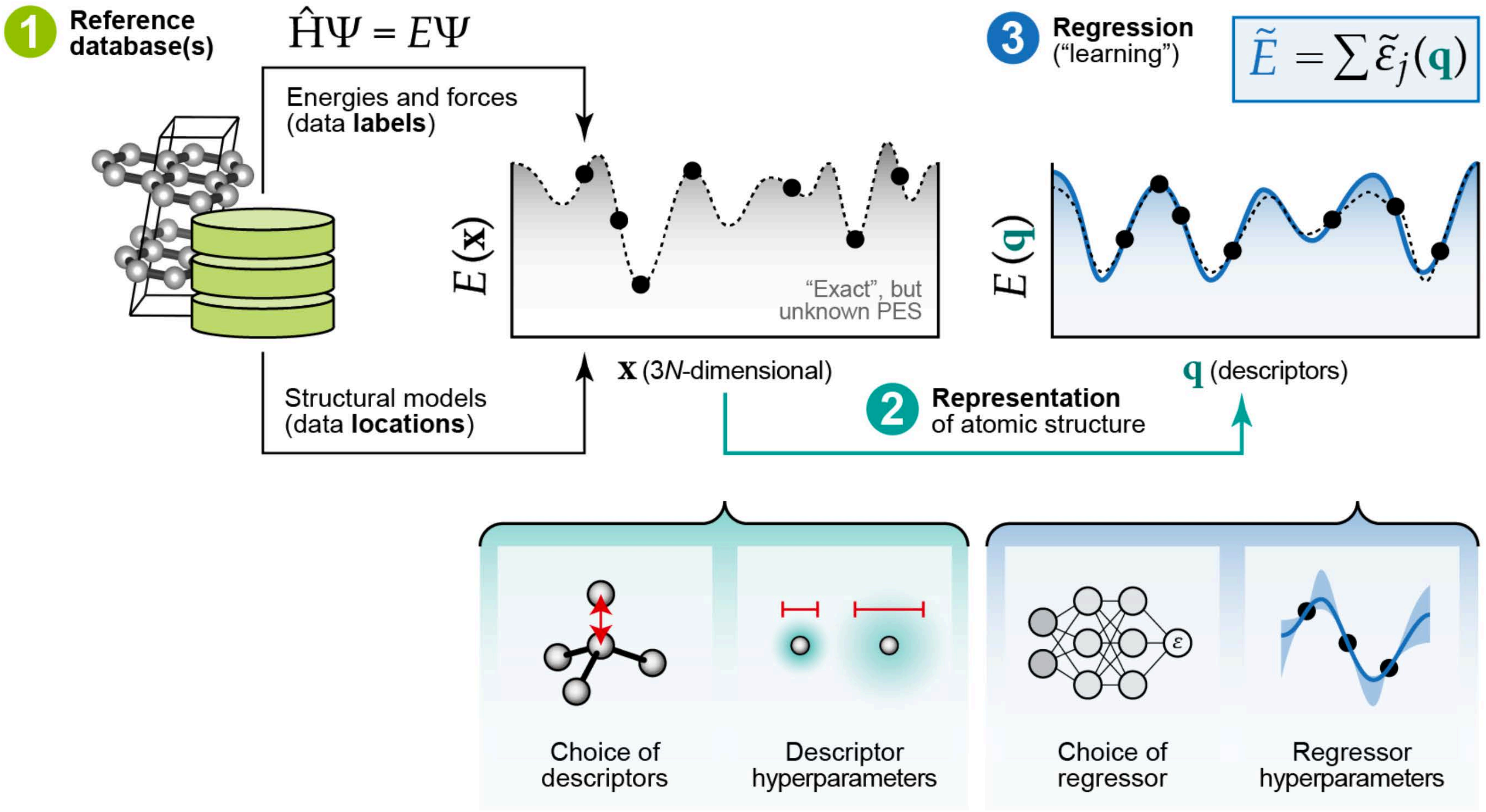
I. Batatia et al. *arXiv:2206.07697* (2022)

$$
\boldsymbol{m}_i^{(t)} = \sum_j \boldsymbol{u}_1\left(\sigma_i^{(t)};\sigma_j^{(t)}\right) + \sum_{j_1,j_2}\boldsymbol{u}_2\left(\sigma_i^{(t)};\sigma_{j_1}^{(t)},\sigma_{j_2}^{(t)}\right) + \cdots + \sum_{j_1,\ldots,j_\nu}\boldsymbol{u}_\nu\left(\sigma_i^{(t)};\sigma_{j_1}^{(t)},\ldots,\sigma_{j_\nu}^{(t)}\right),
$$

$$
A^{(1)}_{i,kl_1m_1} = \sum_{j\in\mathcal{N}(i)} R^{(1)}_{kl_1}(r_{ji})Y^{m_1}_{l_1}(\hat{\boldsymbol{r}}_{ji})\mathcal{W}^{(1)}_{kz_j}.
$$

The math gets complicated, but the models get more accurate (and scale better)

To summarize what we have learned so far


1
Reference database(s)
$\hat{H}\Psi = E\Psi$
Energies and forces (data **labels**)
Structural models (data **locations**)
$E(\mathbf{x})$
"Exact", but unknown PES
$\mathbf{x}$ (3N-dimensional)
2
Representation of atomic structure
$\mathbf{q}$ (descriptors)
3
Regression ("learning")
$\tilde{E} = \sum \tilde{\varepsilon}_j(\mathbf{q})$
$E(\mathbf{q})$
ε
Choice of descriptors
Descriptor hyperparameters
Choice of regressor
Regressor hyperparameters

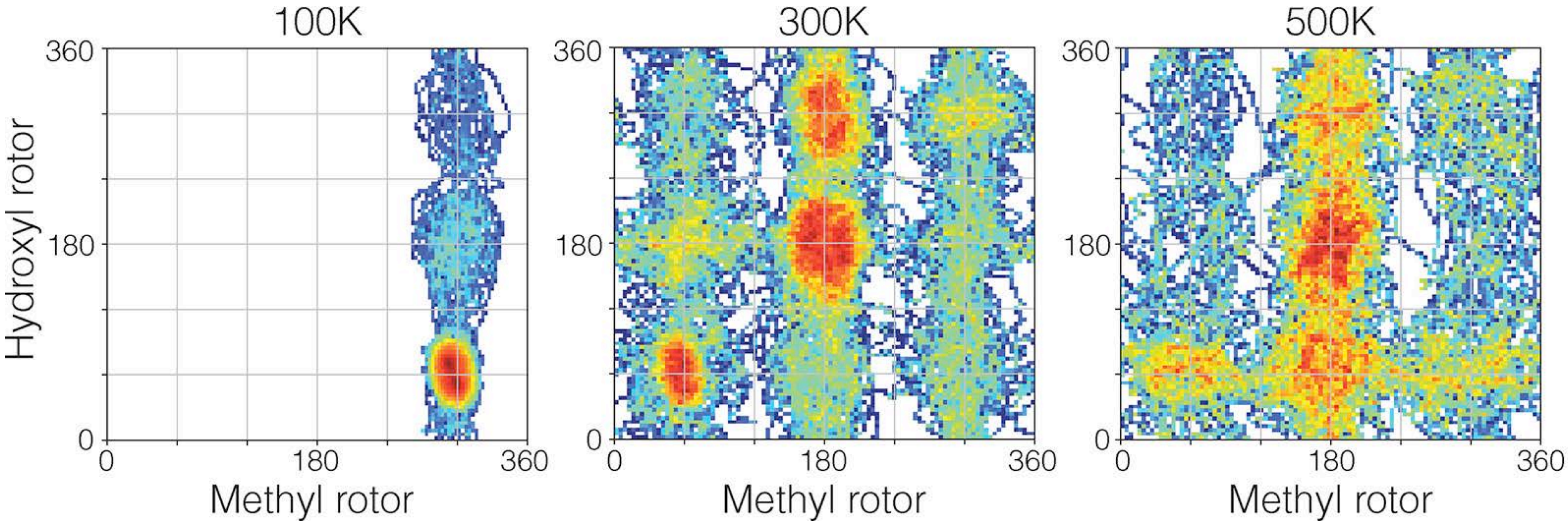
J. Morrow et al. *arXiv*:2211.12484 (2022)

4. Data

How to use, construct, and validate datasets in NNIPs?

Machine learning needs data

For example, if we wanted to perform an MD simulation for ethanol, we we would observe that the sampling of configuration space changes with the temperature, as expected:



O. Unke et al. *Chem. Rev.* **121** (16), 10142 (2021)

To train a NN force field, we have to use the right datasets for our application of interest.

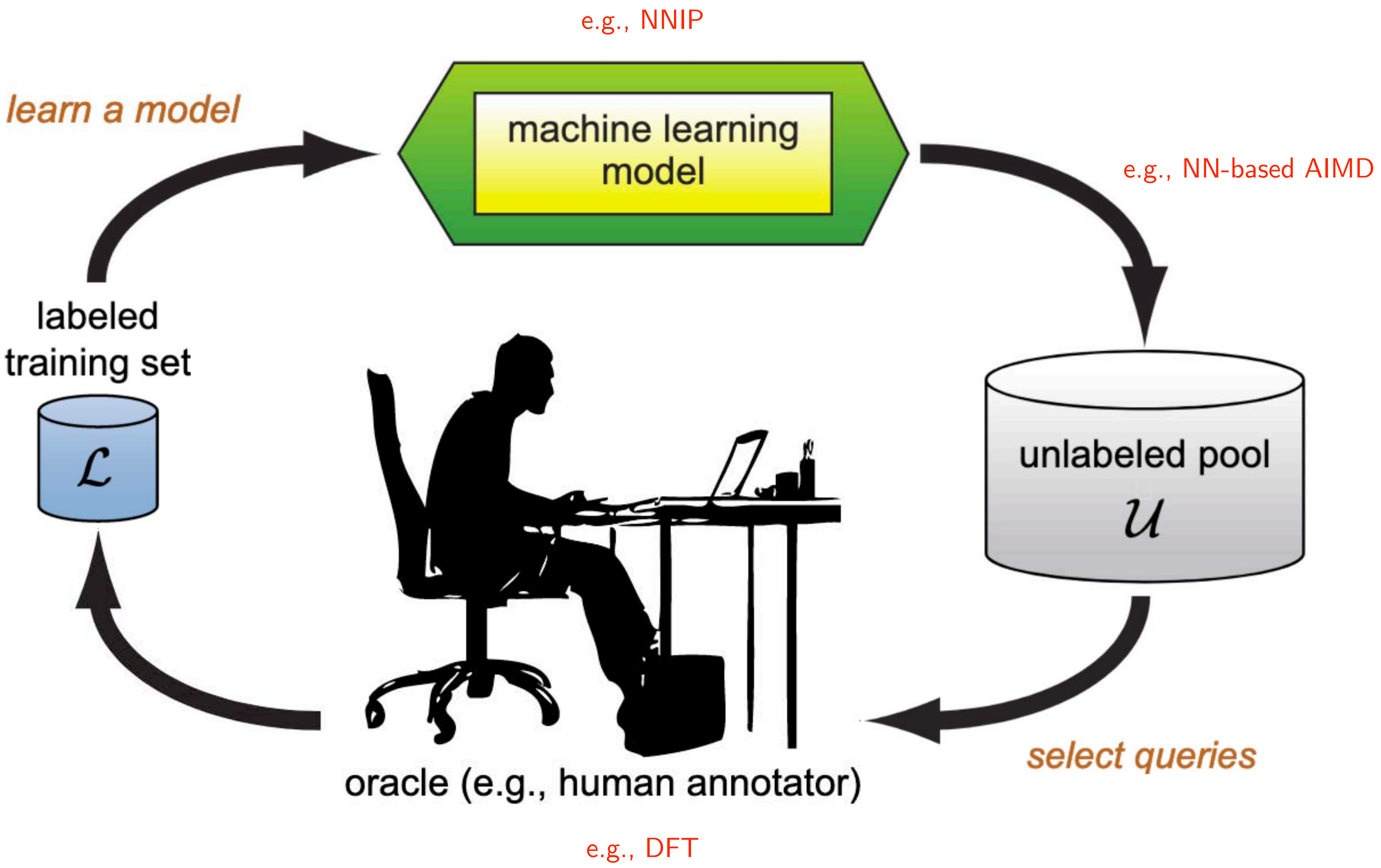
How to create data?

Create data as you usually would: QM, DFT, etc.

AIMD	ℹ️: use AIMD trajectories as dataset
	😁: easy to perform
	😞: high cost, correlated samples
normal mode sampling	ℹ️: displace atoms randomly along the eigenvectors of the Hessian
	😁: easy to perform
	😞: small distortions only
enhanced sampling	ℹ️: explore the PES with enhanced sampling methods (e.g., metadynamics)
	😁: better exploration of the PES
	😞: harder to implement, often relying on *ab initio*
active learning	ℹ️: improve the dataset over time by analyzing the uncertainties
	😁: good quality datasets, may be cheaper to produce
	😞: requires uncertainty metric, long process

let's focus on this one

What is active learning?



e.g., NNIP

learn a model

machine learning model

e.g., NN-based AIMD

labeled training set

$\mathcal{L}$

unlabeled pool

$\mathcal{U}$

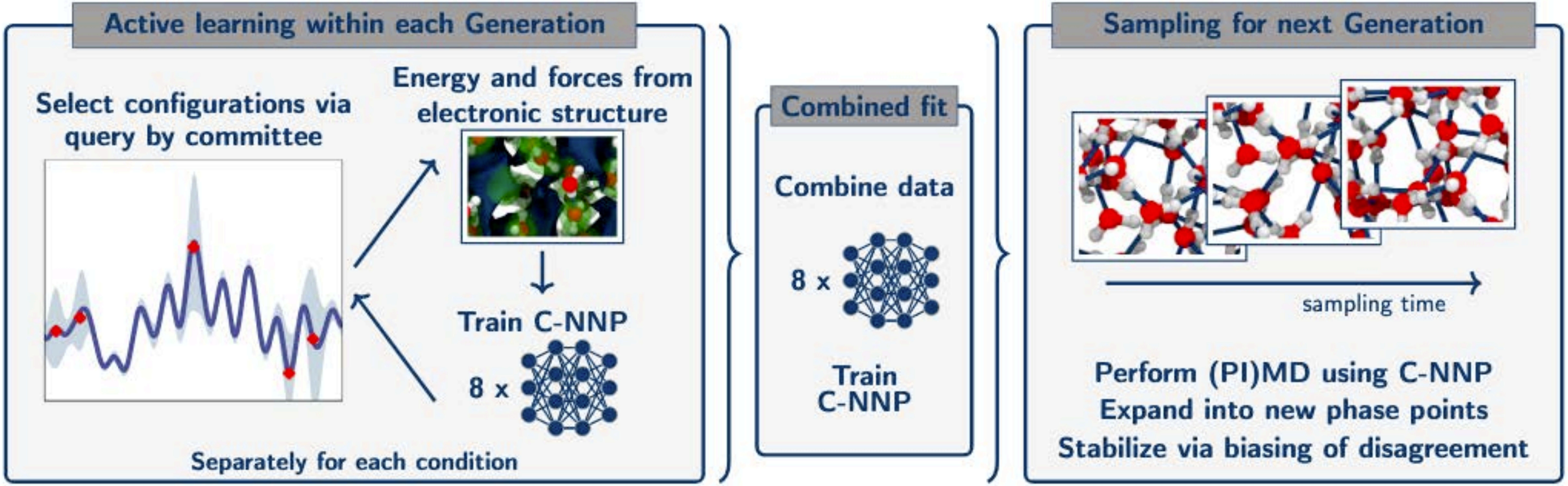
select queries

oracle (e.g., human annotator)

e.g., DFT

B. Settles. *Active Learning Literature Survey* (2009)

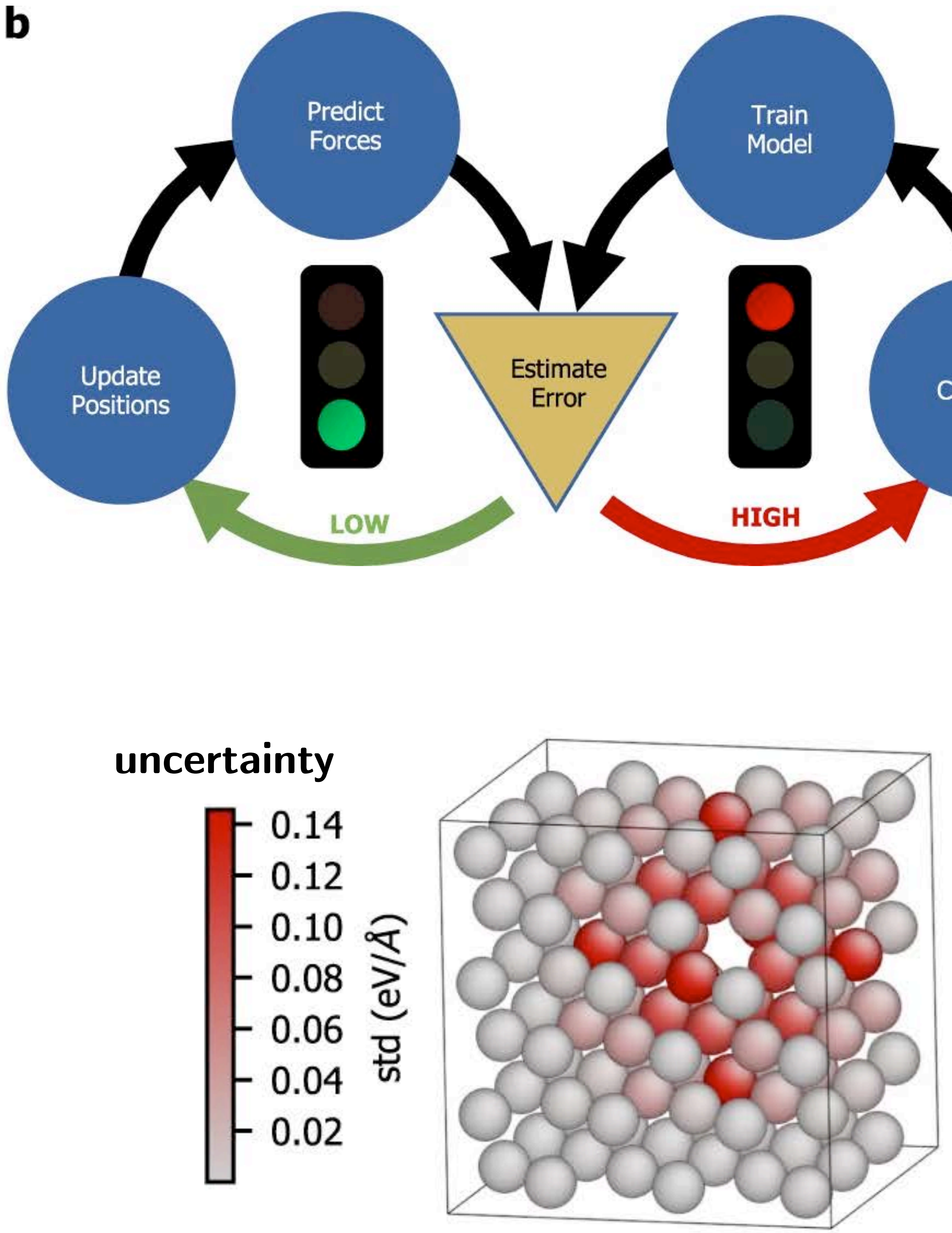
Active learning requires uncertainty quantification to identify "unlabeled" configurations

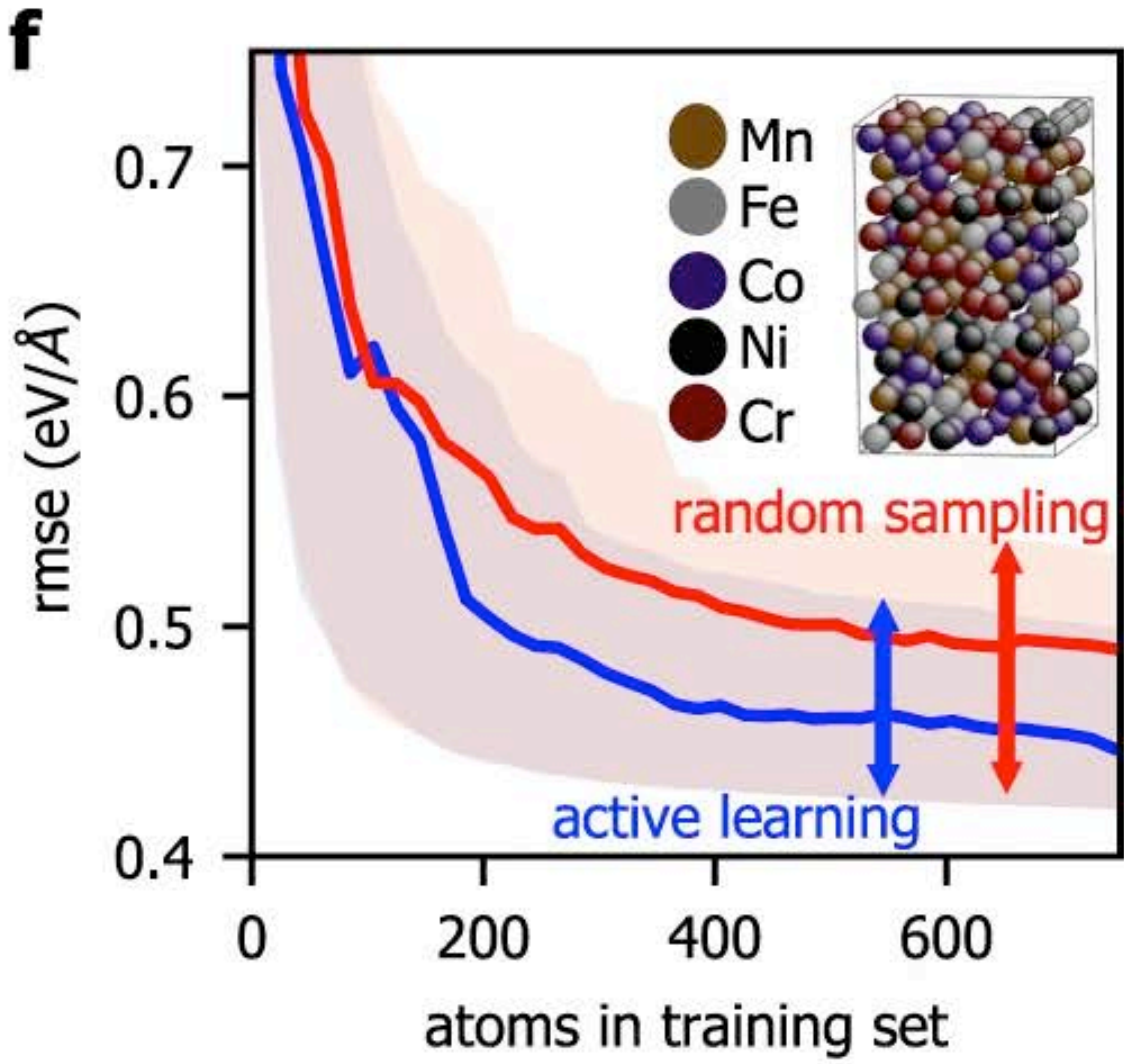

Active learning within each Generation
Select configurations via query by committee
Energy and forces from electronic structure
Train C-NNP
8 x
Separately for each condition
Combined fit
Combine data
8 x
Train C-NNP
Sampling for next Generation
sampling time
Perform (PI)MD using C-NNP
Expand into new phase points
Stabilize via biasing of disagreement

What is the problem of this?

C. Schran et al. *J. Chem. Phys.* **153**, 104105 (2020)

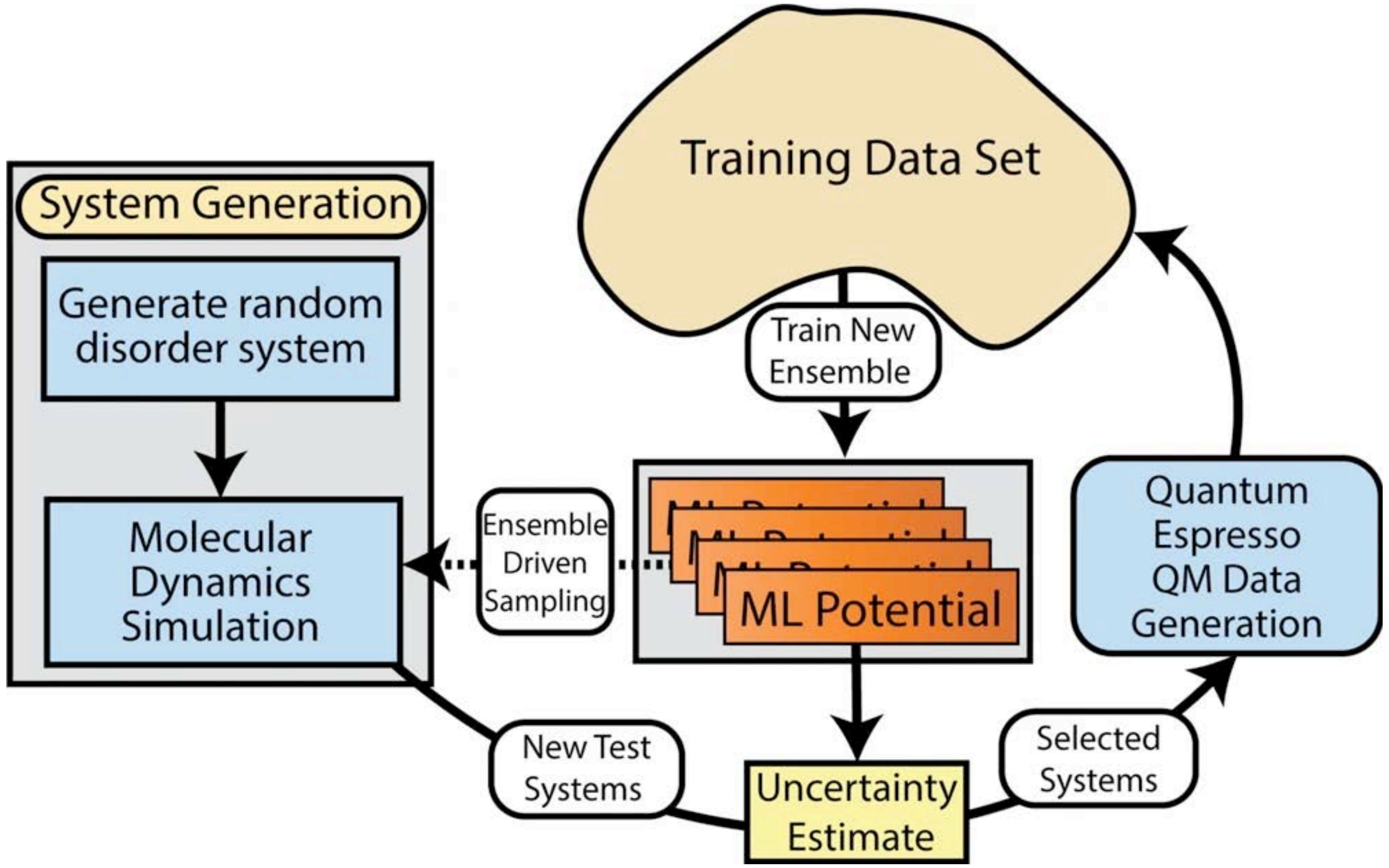
Examples of interesting dataset constructions: GPR, FLARE and high-entropy alloys

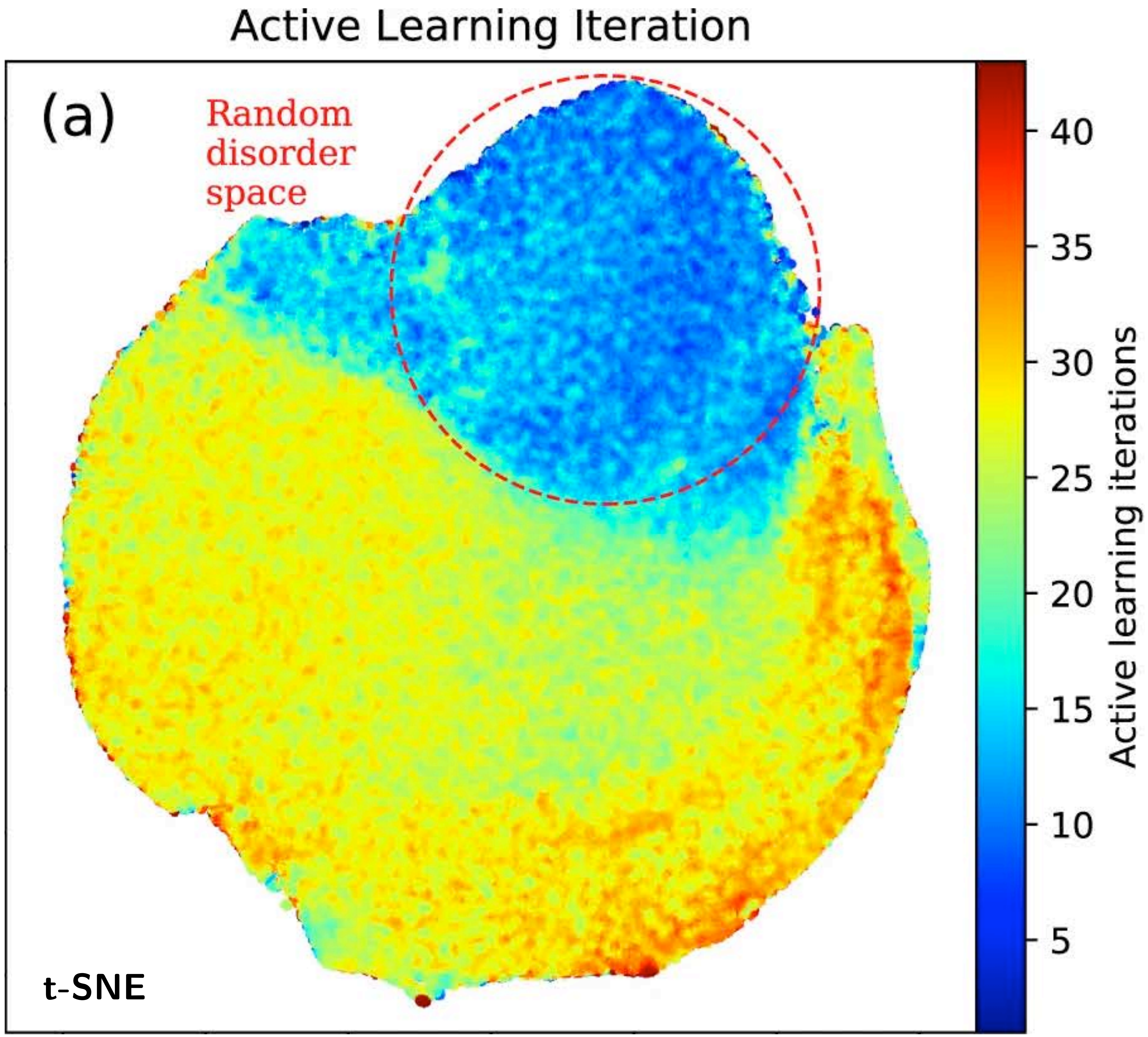




J. Vandermause et al. *npj Comp. Mater.* **6**, 20 (2020)

Examples of interesting dataset constructions: aluminum dataset from ANI potential

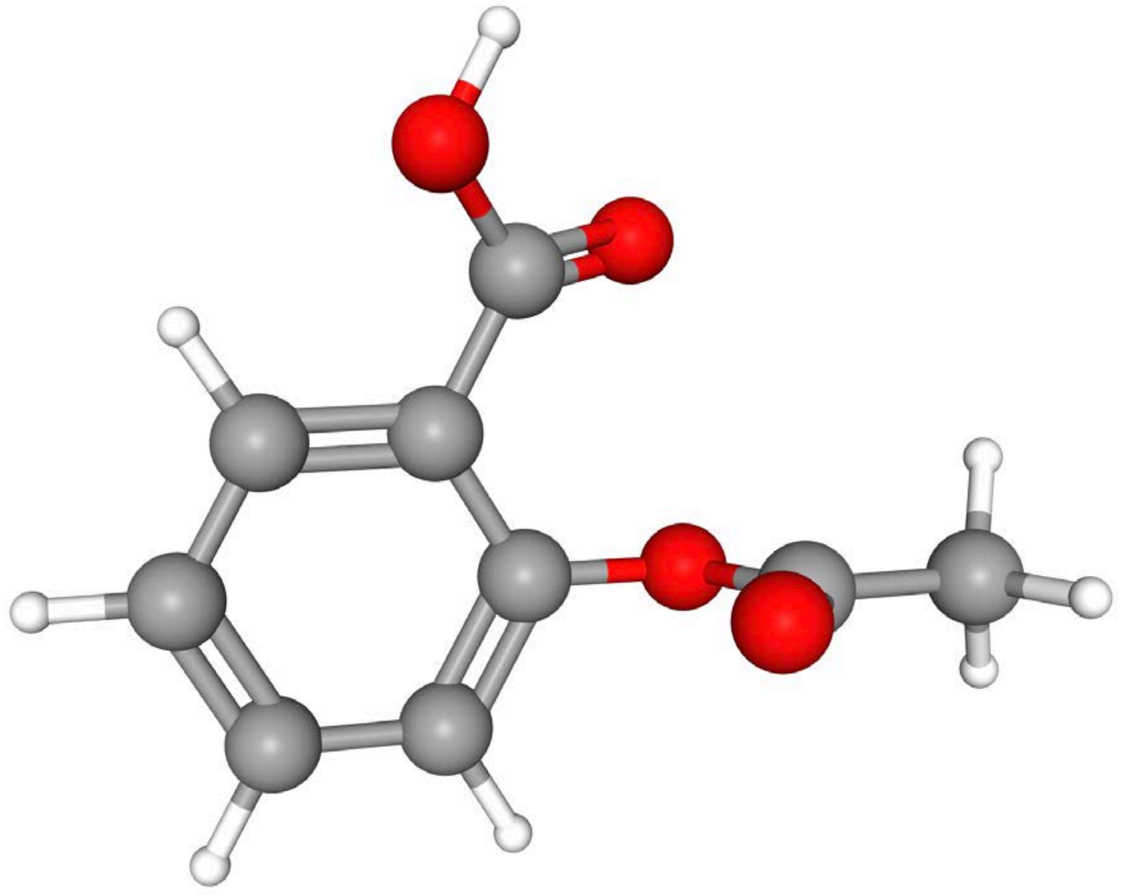




J. Smith et al. *Nat. Commun.* **12**, 1257 (2021)

Some datasets typically used out there

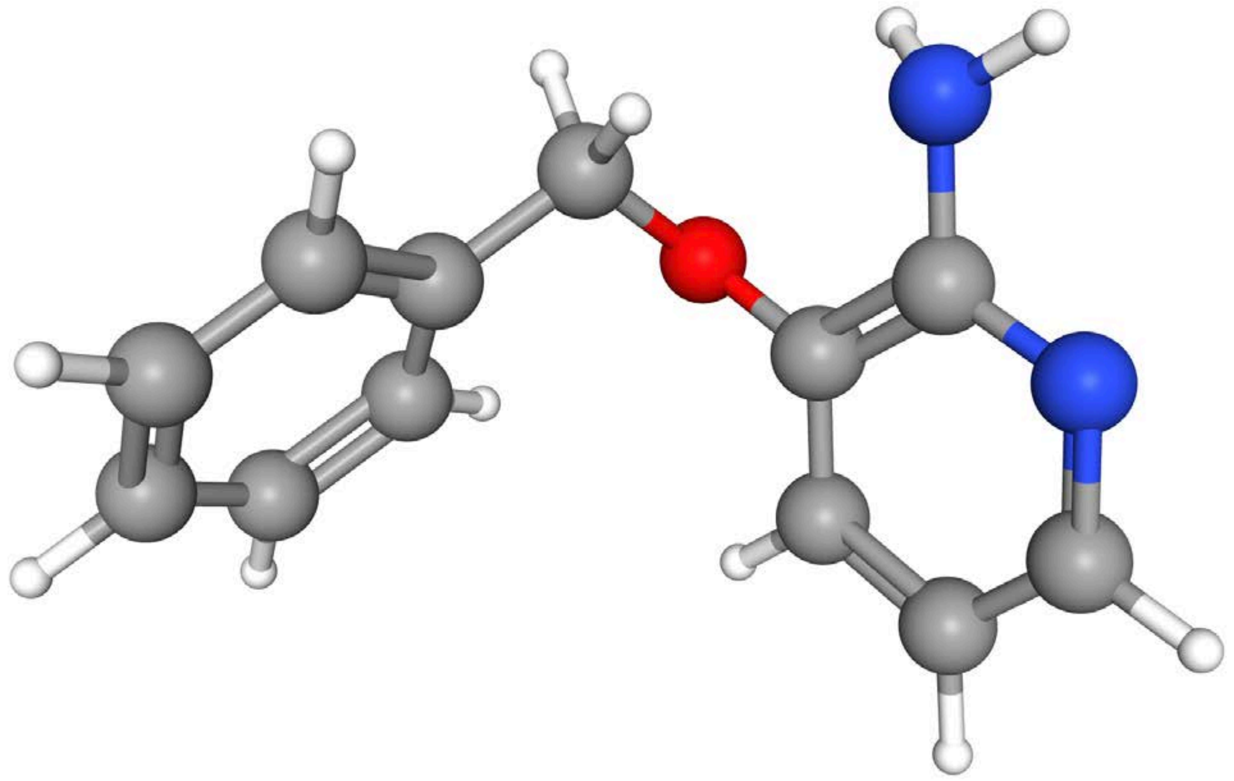
rMD17



S. Chmiela et al. *Sci. Adv.* **3** (5), e1603015 (2017)
A. Christiansen et al. *MLST* **1**, 045018 (2020)

Small molecules with their conformers

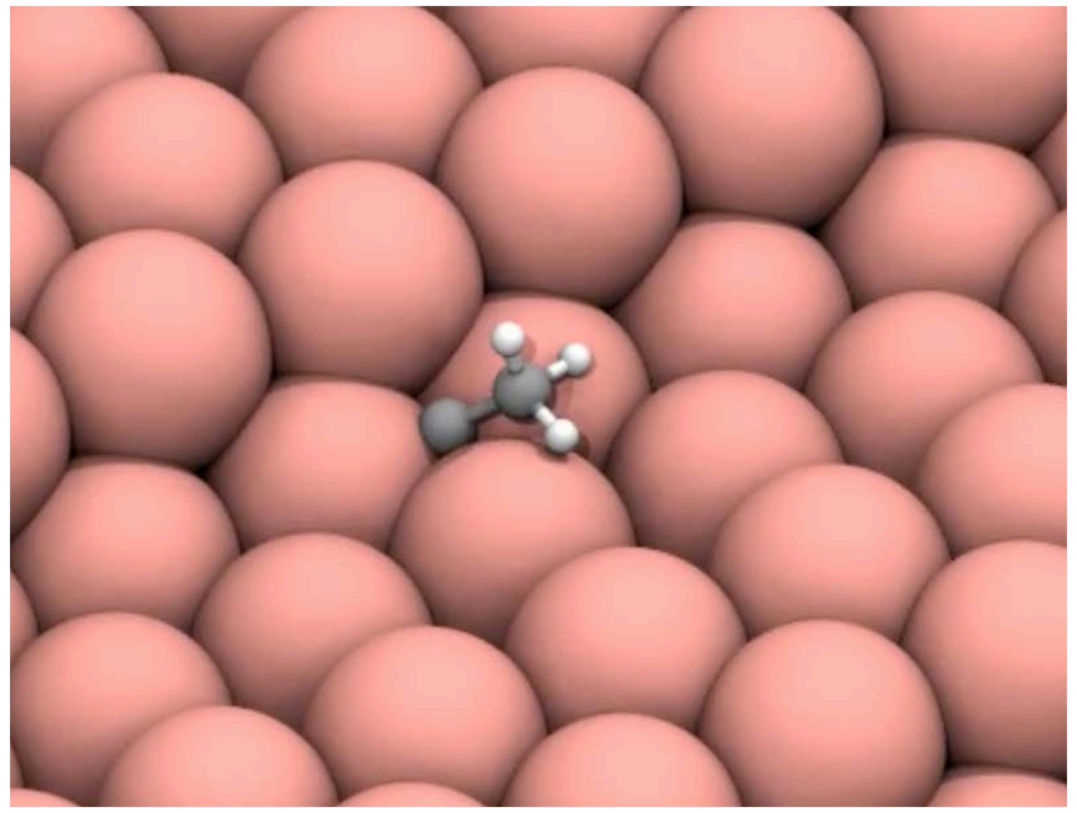
3BPA



D. Kovács et al. *JCTC* **17** (12), 7696 (2021)

Flexible molecule sampled at different temperatures

Open Catalyst (OC20)



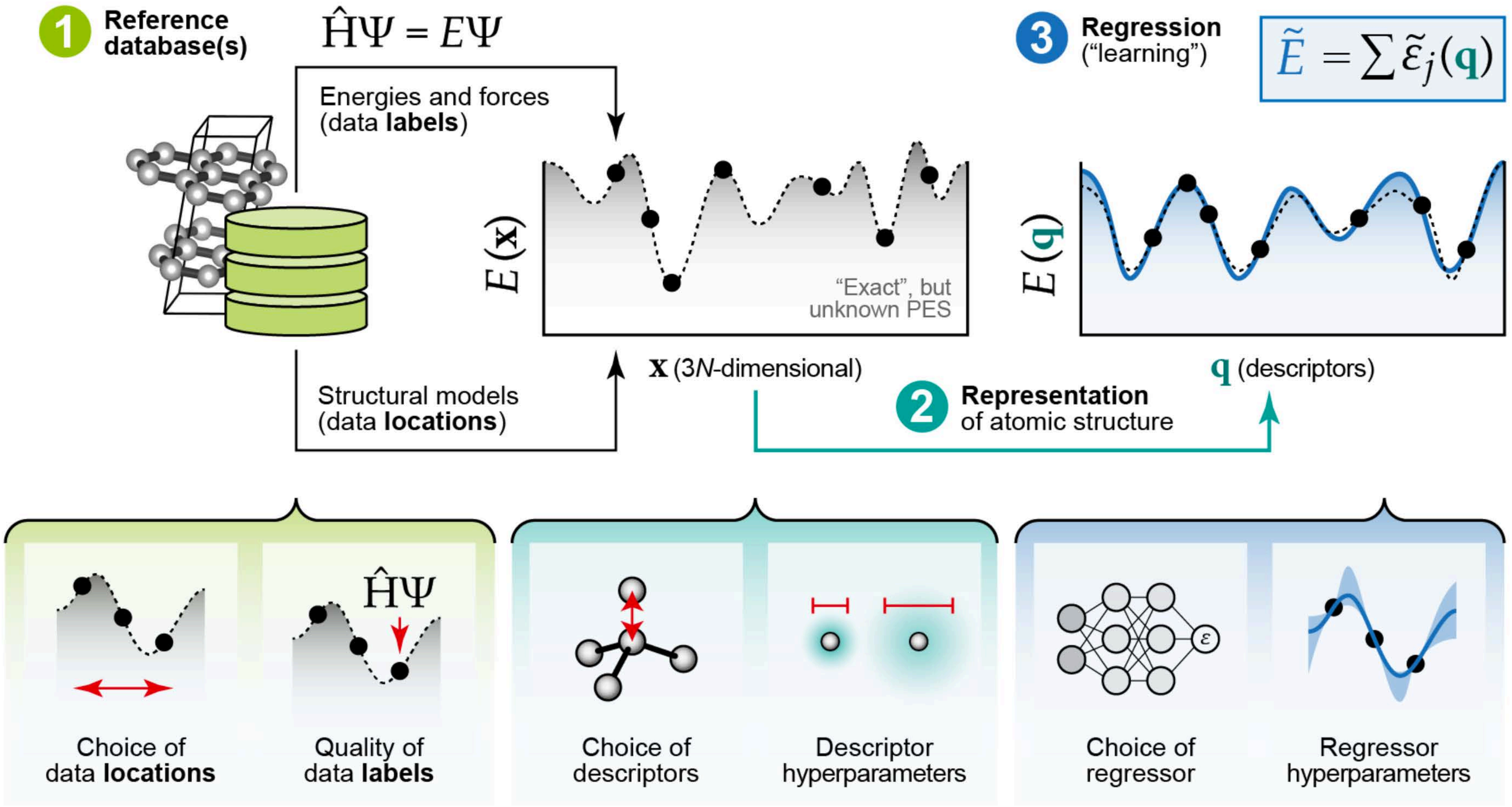
L. Chanussot et al. *ACS Catalysis* **11**, 6059 (2021)

Adsorption energies on inorganic catalysts

Remember: train-validation-test, k-fold CV, error metrics and many more.

There are many, many datasets...

To summarize what we have learned so far


1 Reference database(s)
$\hat{H}\Psi = E\Psi$
Energies and forces (data **labels**)
Structural models (data **locations**)
$E(\mathbf{x})$
"Exact", but unknown PES
$\mathbf{x}$ (3N-dimensional)
3 Regression ("learning")
$\tilde{E} = \sum \tilde{\varepsilon}_j(\mathbf{q})$
$E(\mathbf{q})$
$\mathbf{q}$ (descriptors)
2 Representation of atomic structure
$\hat{H}\Psi$
ε
Choice of data **locations**
Quality of data **labels**
Choice of descriptors
Descriptor hyperparameters
Choice of regressor
Regressor hyperparameters

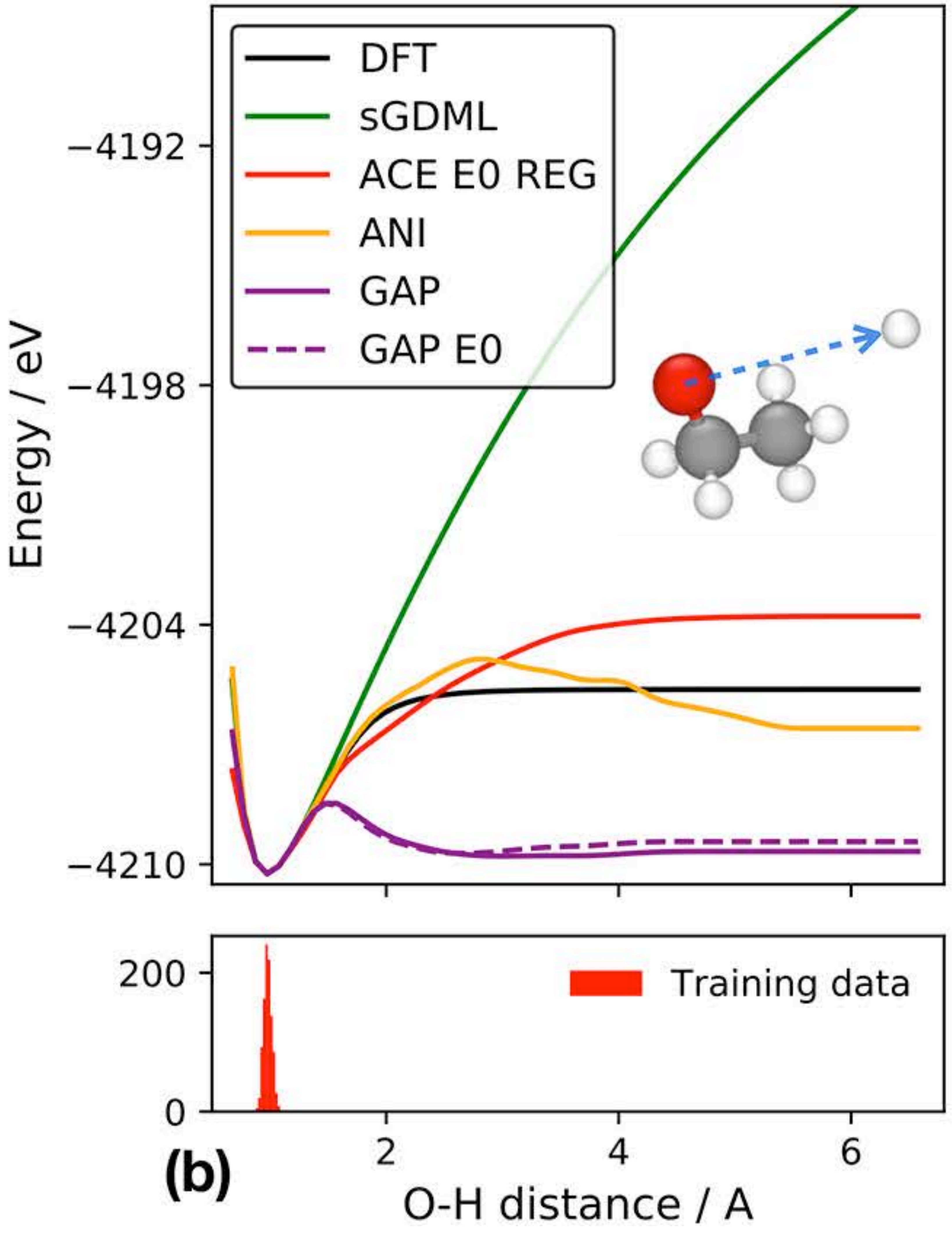
J. Morrow et al. *arXiv*:2211.12484 (2022)

5. Frontiers of NNIPs

What’s next for ML interatomic potentials?

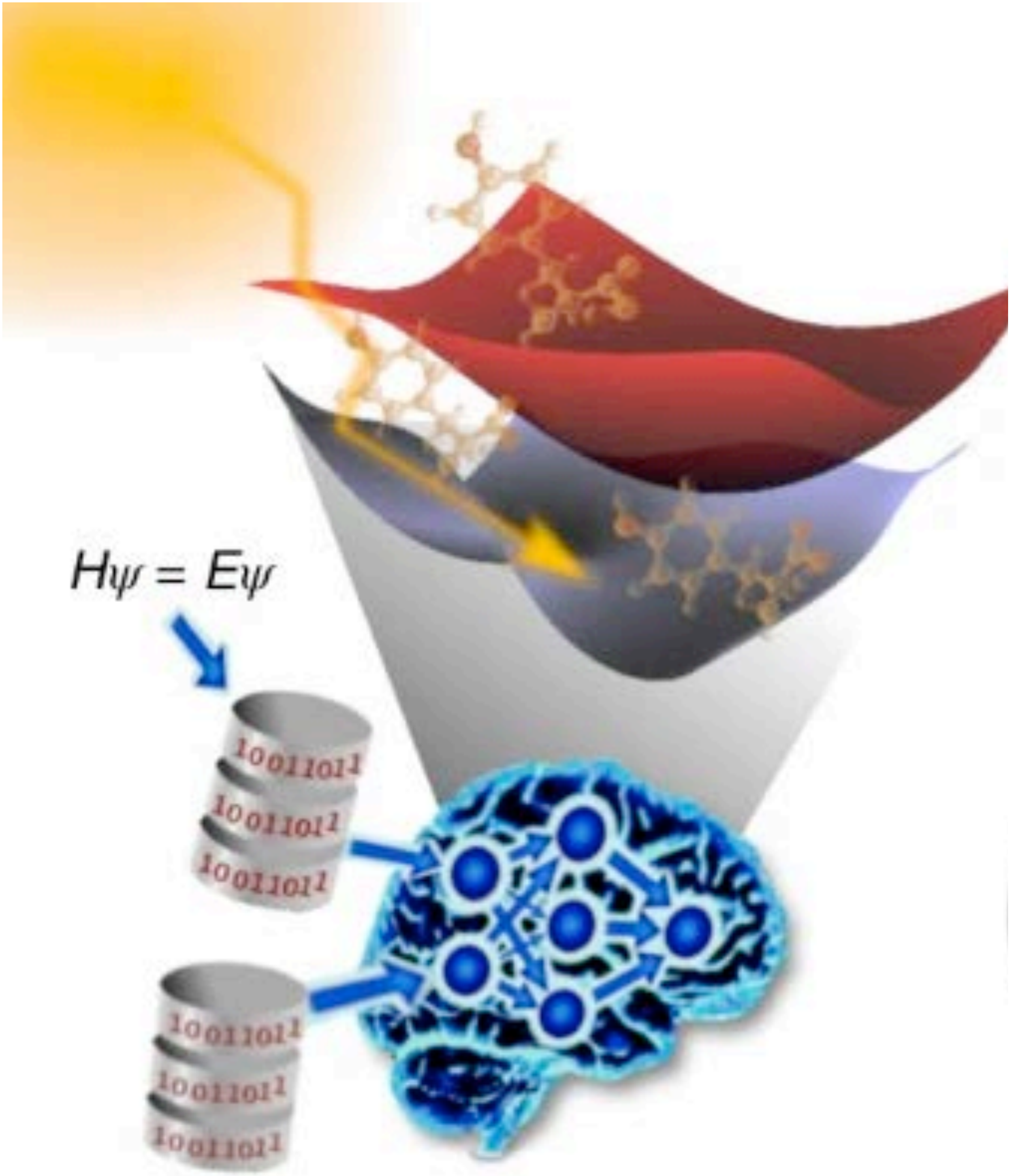
Interesting trends in the field

Uncertainty & extrapolation



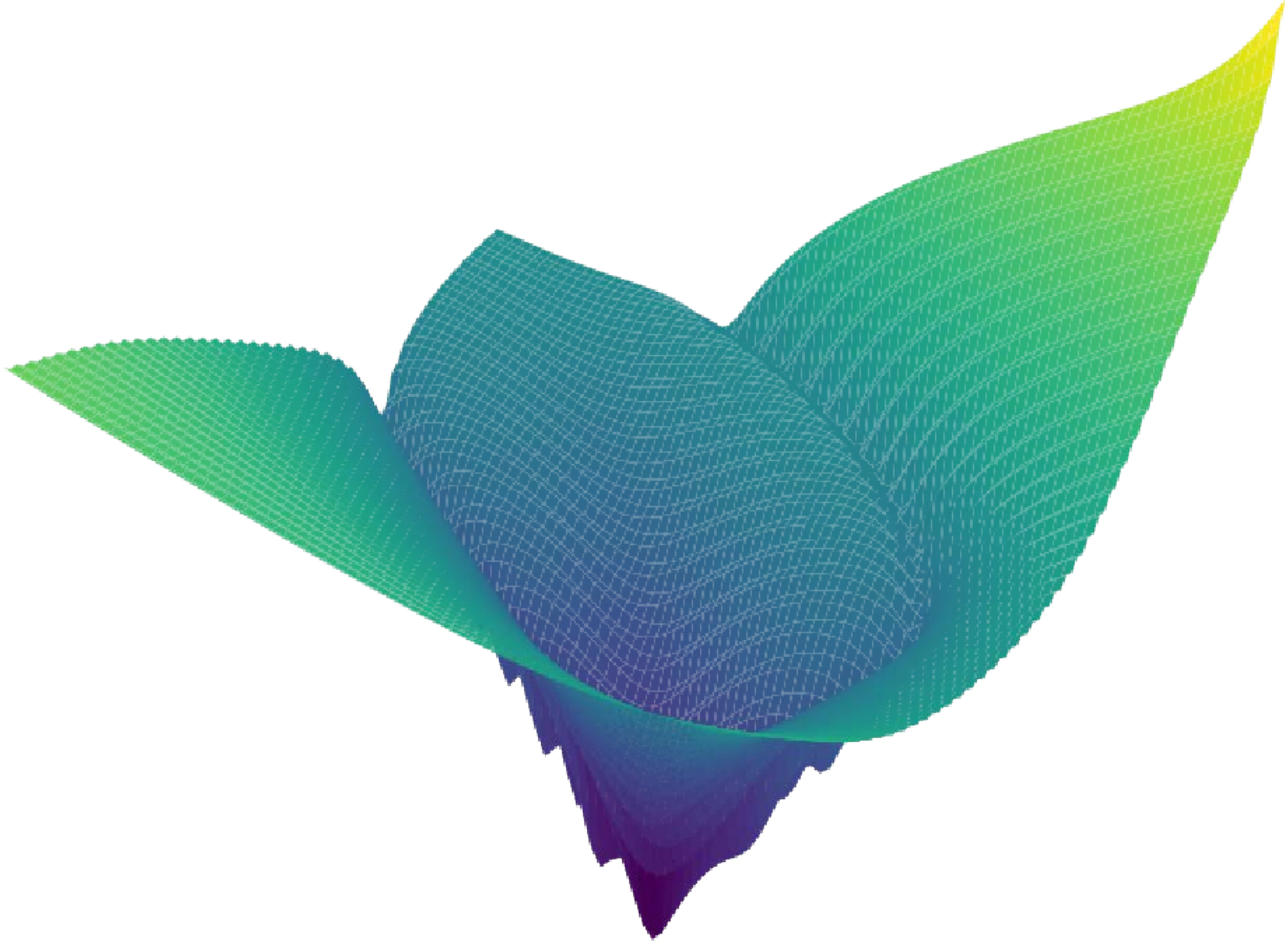
D. Kovács et al. *JCTC* **17** (12), 7696 (2021)

Challenging applications



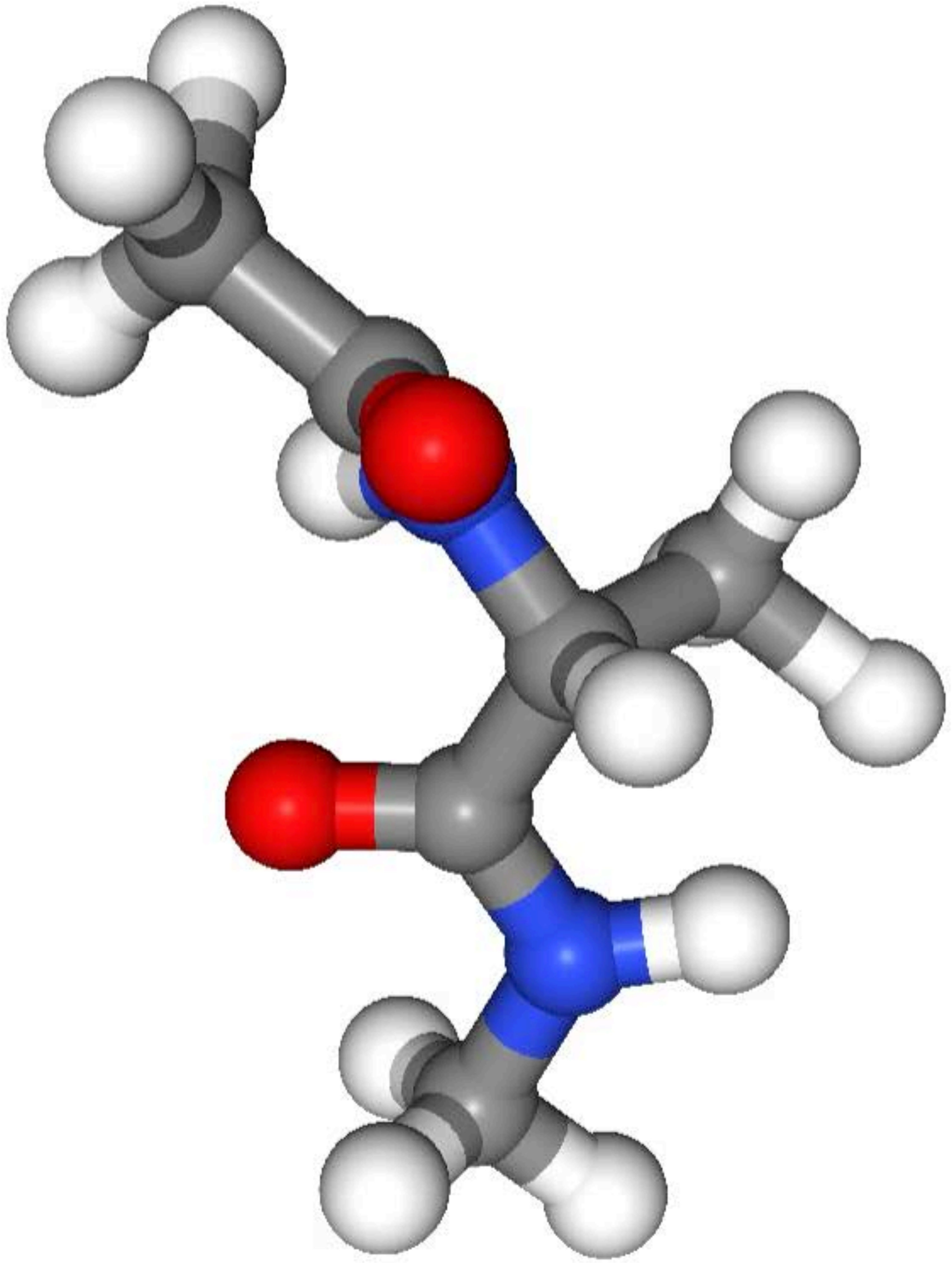
J. Westermayr et al. *Nat. Chem.* **14**, 914 (2022)

Better metrics & interpretability

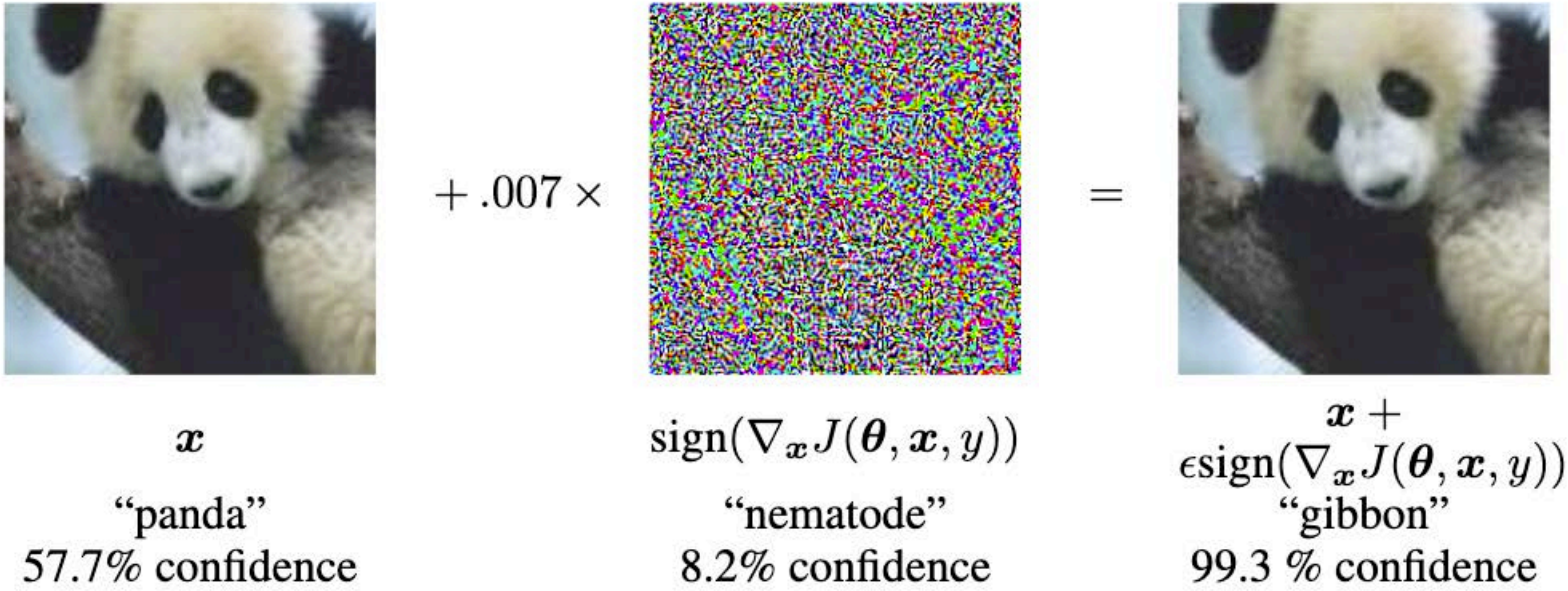


J. Vita and D. Schwalbe-Koda (2023)

For example: ML potentials suffer when extrapolating...



Neural networks are susceptible to adversarial attacks


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} +$
$\epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

C. Szegedy et al. (2013), *arXiv:1312.6199*
I. Goodfellow et al. ICLR (2014), *arXiv:1412.6572*

Increasing NN robustness to adversarial attacks

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathscr{D}} \left[\max_{\delta\in\Delta} \mathscr{L}(h_{\theta}(x+\delta), y) \right]$$

find the NN weights that minimize

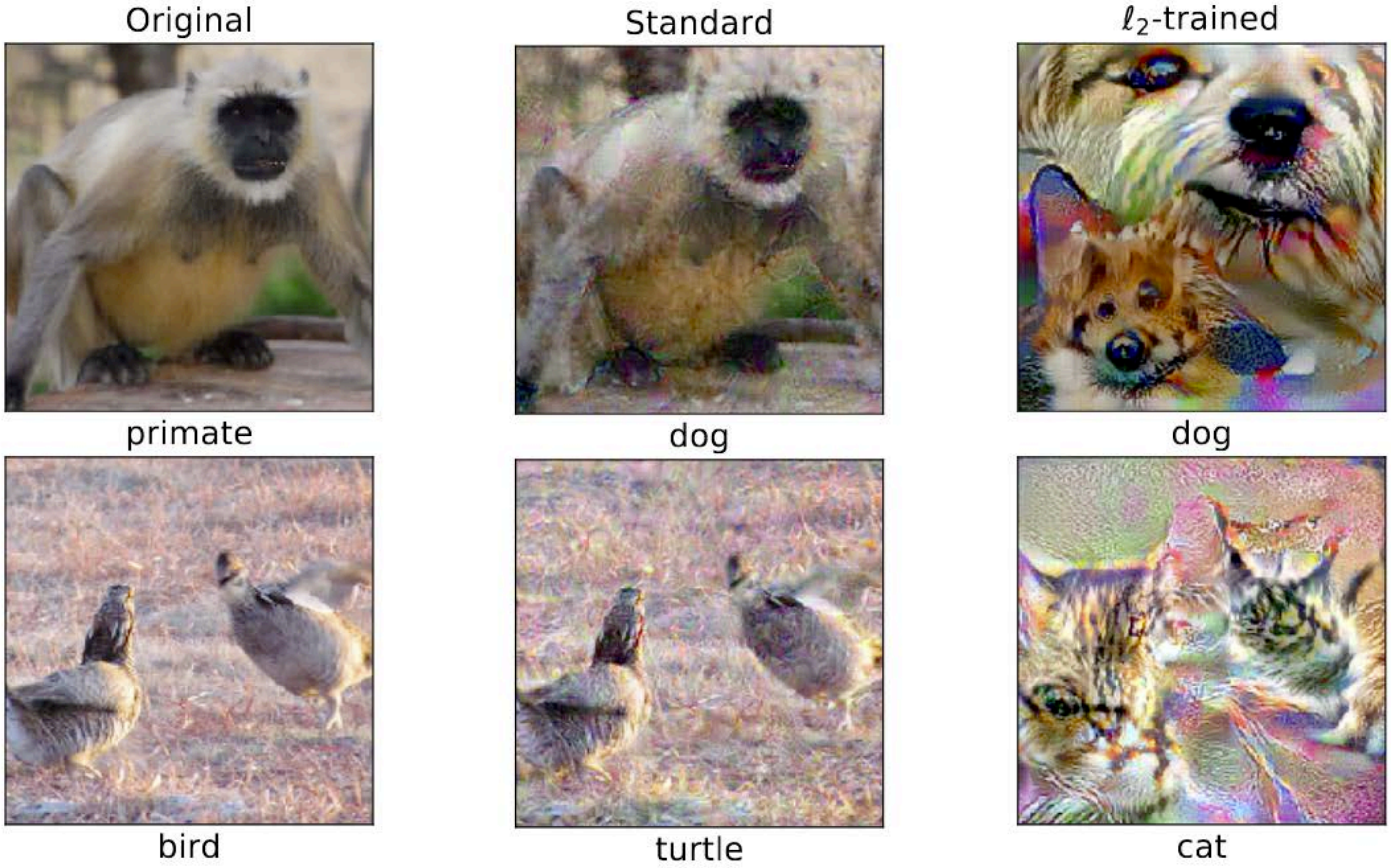
across the whole dataset under study

and for perturbations δ in the set of allowed perturbations Δ

the perturbed loss function

D. Tsipras et al. ICLR (2019)

Qualitative results of robust NNs



Question: how to do this for NN potentials?

D. Tsipras et al. ICLR (2019)

Objective of a robust neural network regressor

E$_\delta$, F$_\delta$ come from reference calculations in an AL loop

$$\min_{\theta} \mathbb{E}_{(X,E,\mathbf{F})\sim\mathscr{D}} \left[\max_{\delta\in\Delta} \mathscr{L}(X_\delta, E_\delta, \mathbf{F}_\delta; \theta) \right]$$

find the NN weights that minimize

across the whole dataset under study

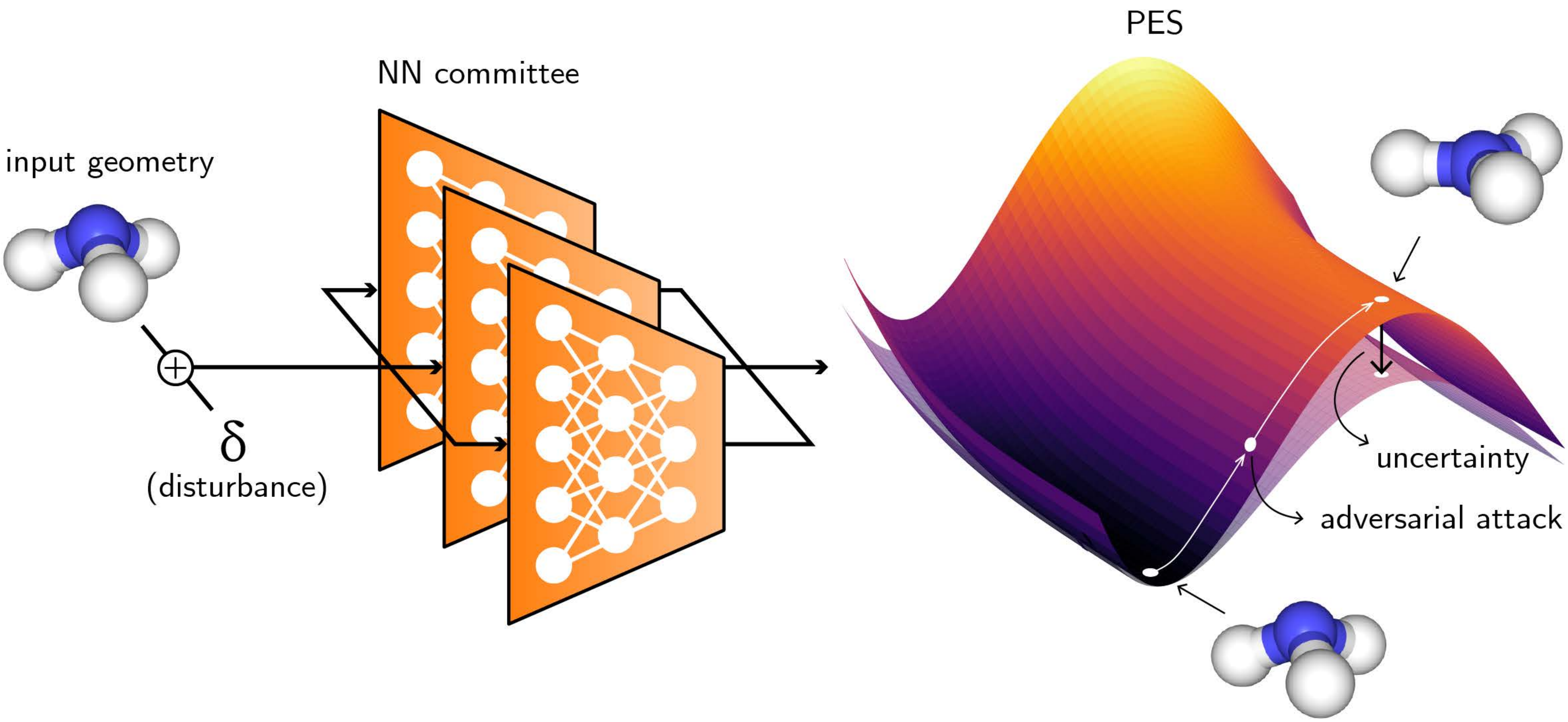
and for perturbations δ in the set of allowed perturbations Δ

the perturbed loss function of the regressor

Question: how to generate the perturbed samples and their ground truth values?

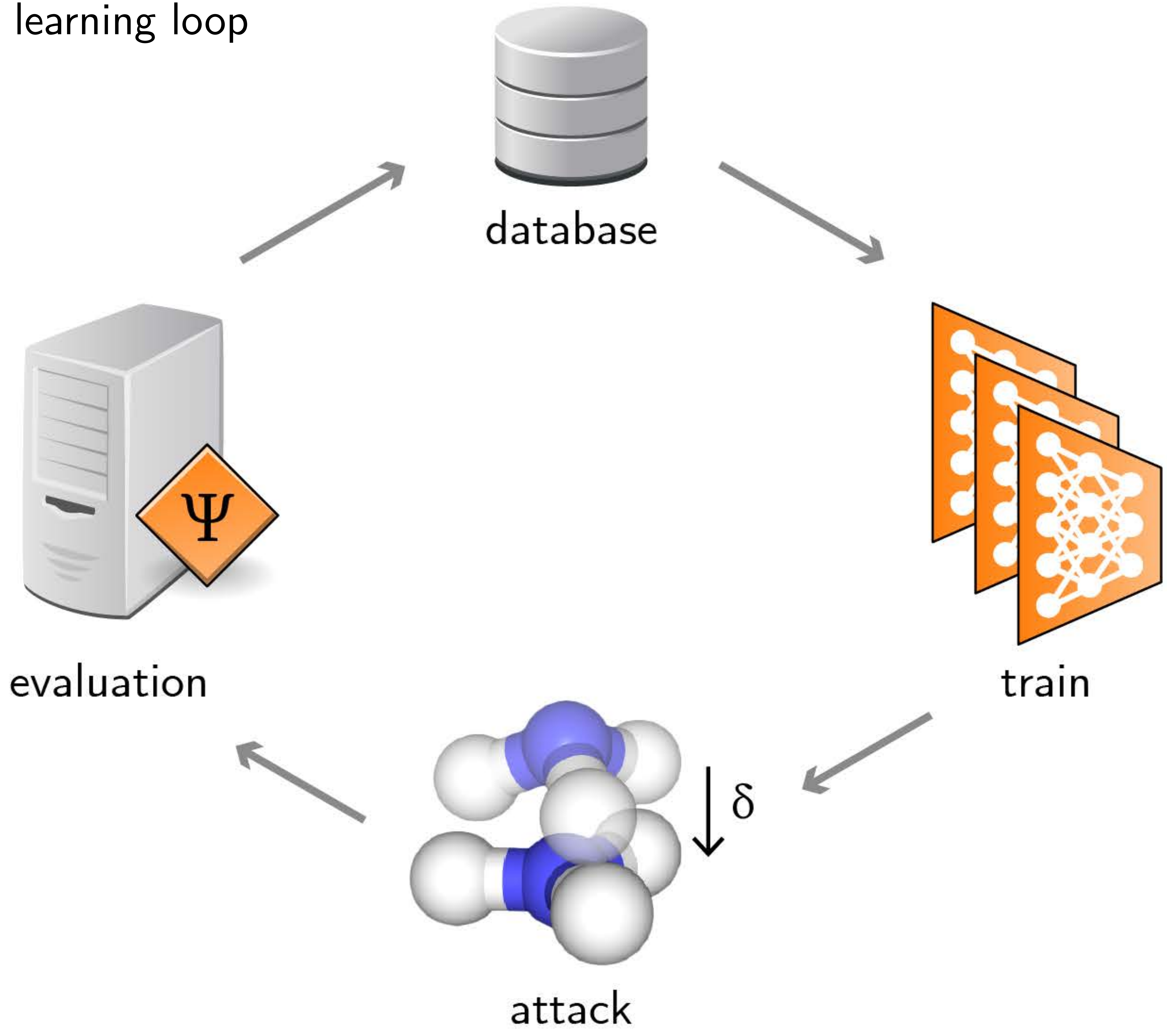
Idea: find geometries that maximize the epistemic uncertainty of the NN potential!

Adversarial loss depends on the uncertainty



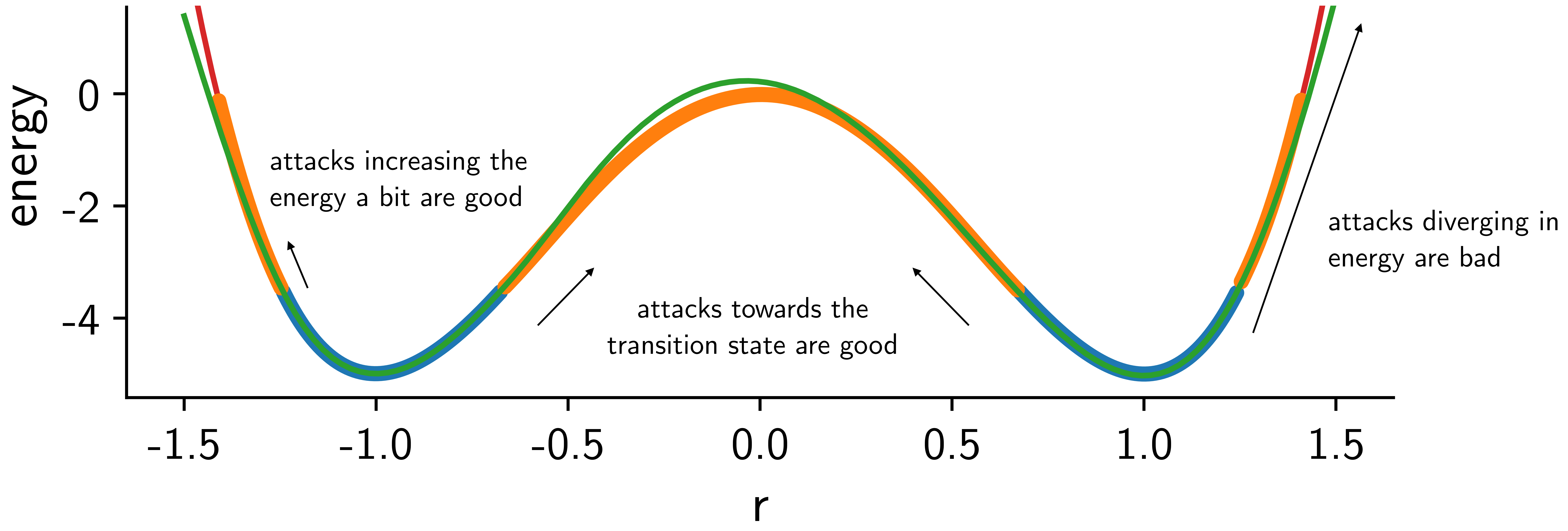
D. Schwalbe-Koda et al. *Nat. Commun.* **12**, 5104 (2021)

Robust training is an active learning loop



D. Schwalbe-Koda et al. *Nat. Commun.* **12**, 5104 (2021)

Sample new points through adversarial attacks



attacks increasing the energy a bit are good

attacks towards the transition state are good

attacks diverging in energy are bad

Goals of a good adversarial attack:

- Find points of maximum uncertainty
- Penalize going towards crazy high energies

$$\min_{\theta} \mathbb{E}_{(X,E,\mathbf{F})\sim\mathscr{D}} \left[\max_{\boxed{\delta\in\Delta}} \mathscr{L}(X_\delta, E_\delta, \mathbf{F}_\delta; \theta) \right]$$

D. Schwalbe-Koda et al. *Nat. Commun.* **12**, 5104 (2021)

Loss function for adversarial attack

Construct partition function from training set:

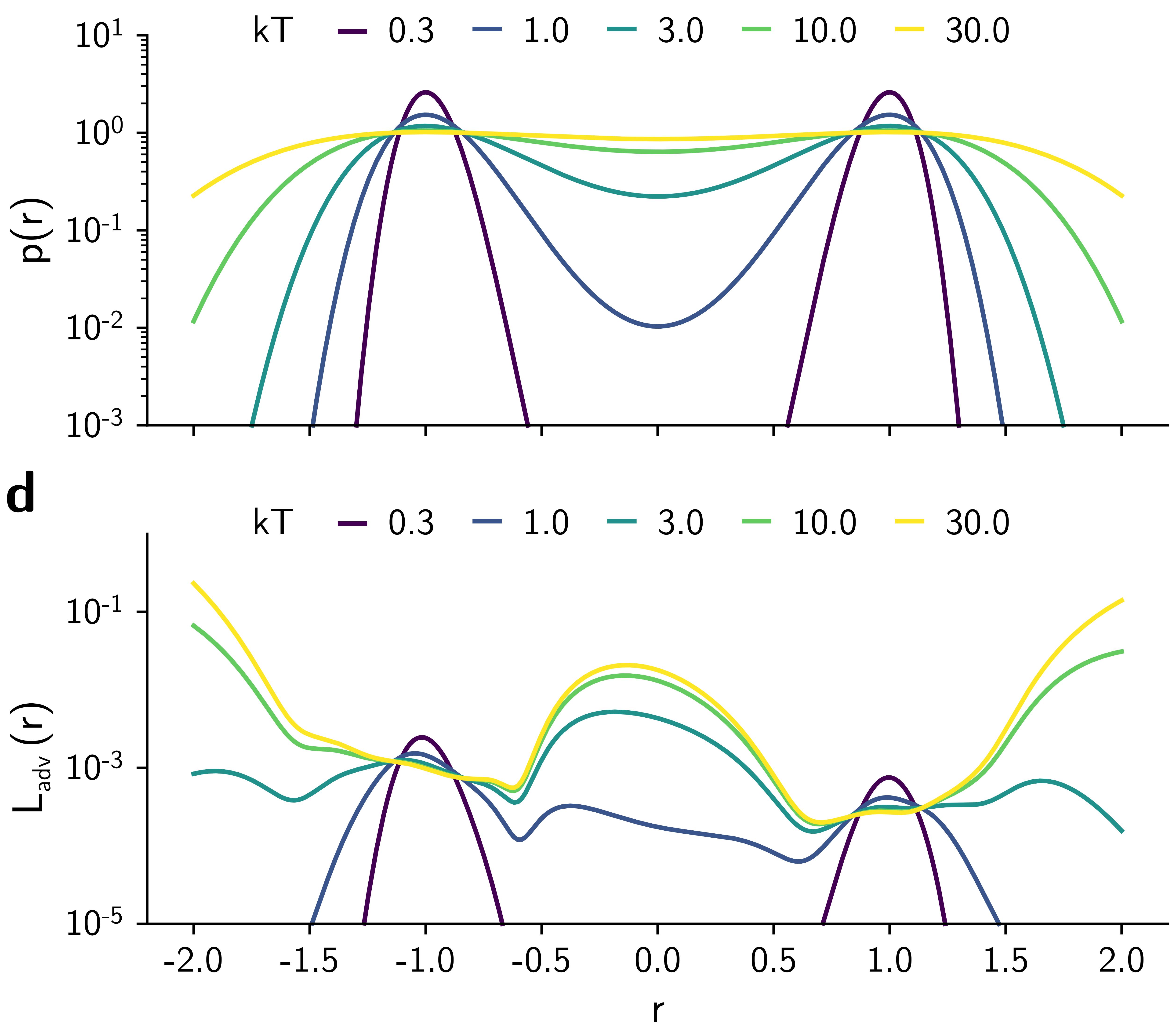
$$Q = \sum_{(X,E,\mathbf{F})\in\mathscr{D}} \exp\left(\frac{-E}{kT}\right)$$

Estimate Boltzmann probability given the mean energy from NN ensemble:

$$p(X_\delta) = \frac{1}{Q}\exp\left(\frac{-\bar{E}(X_\delta)}{kT}\right)$$

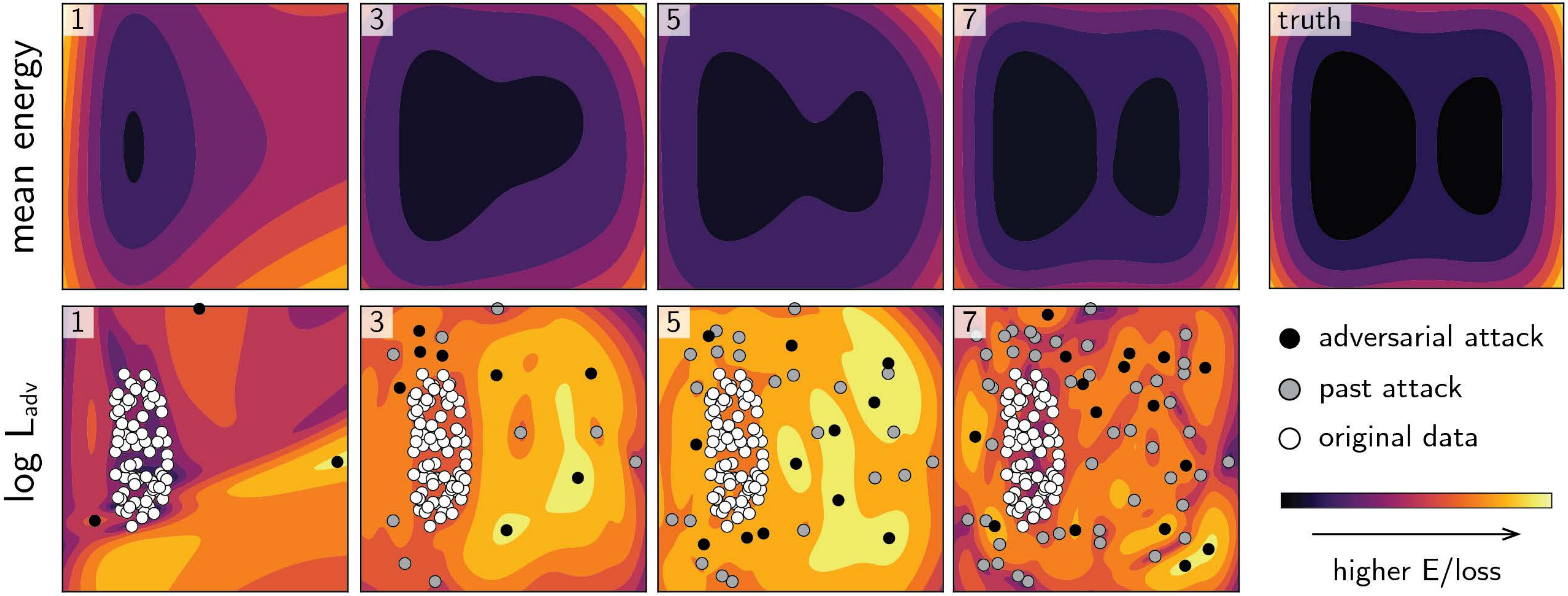
The final adversarial objective then becomes

$$\max_\delta \mathscr{L}(X,\delta;\theta) = \max_\delta p(X_\delta)\cdot\sigma_F^2(X_\delta)$$



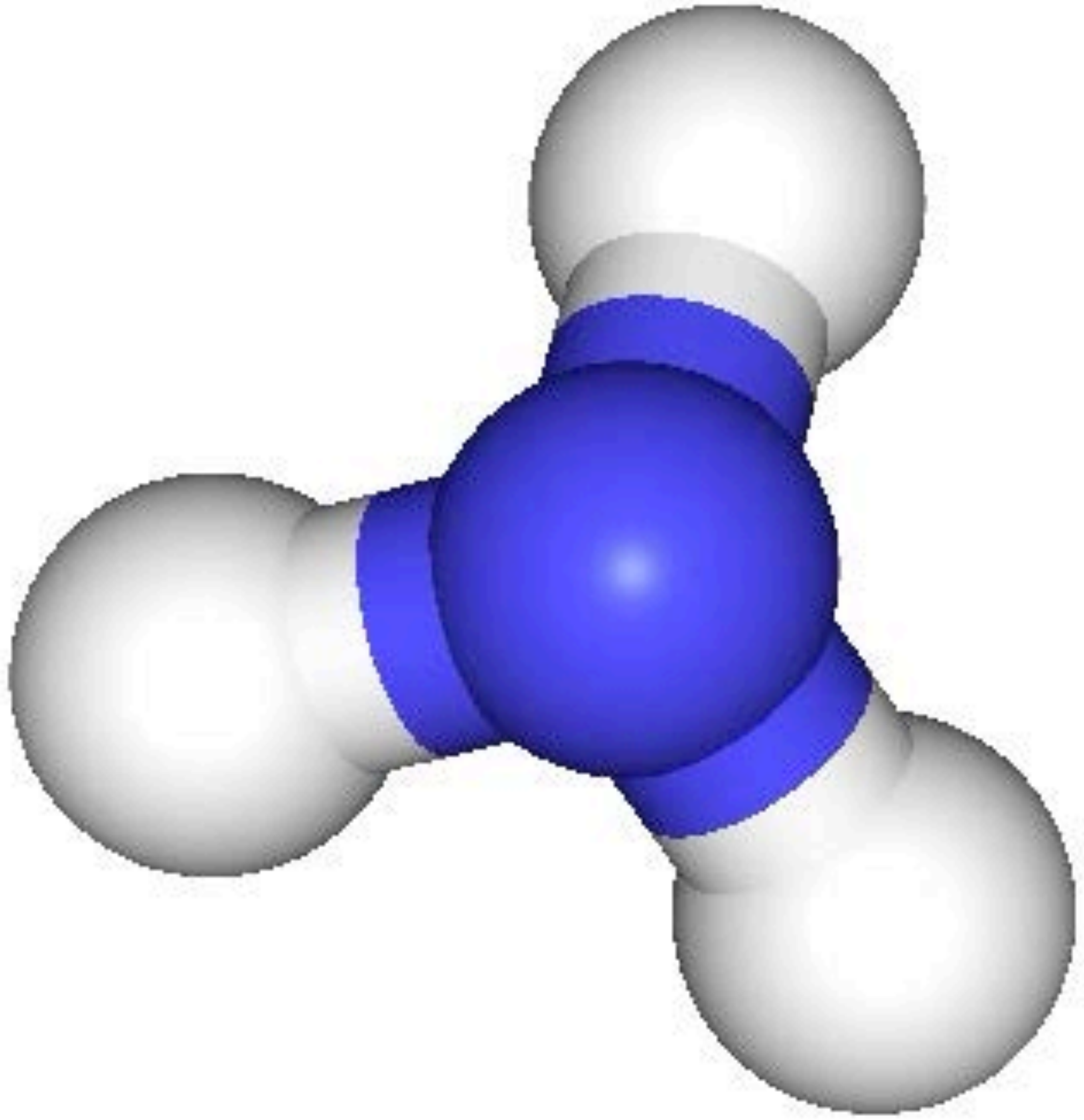
D. Schwalbe-Koda et al. *Nat. Commun.* **12**, 5104 (2021)

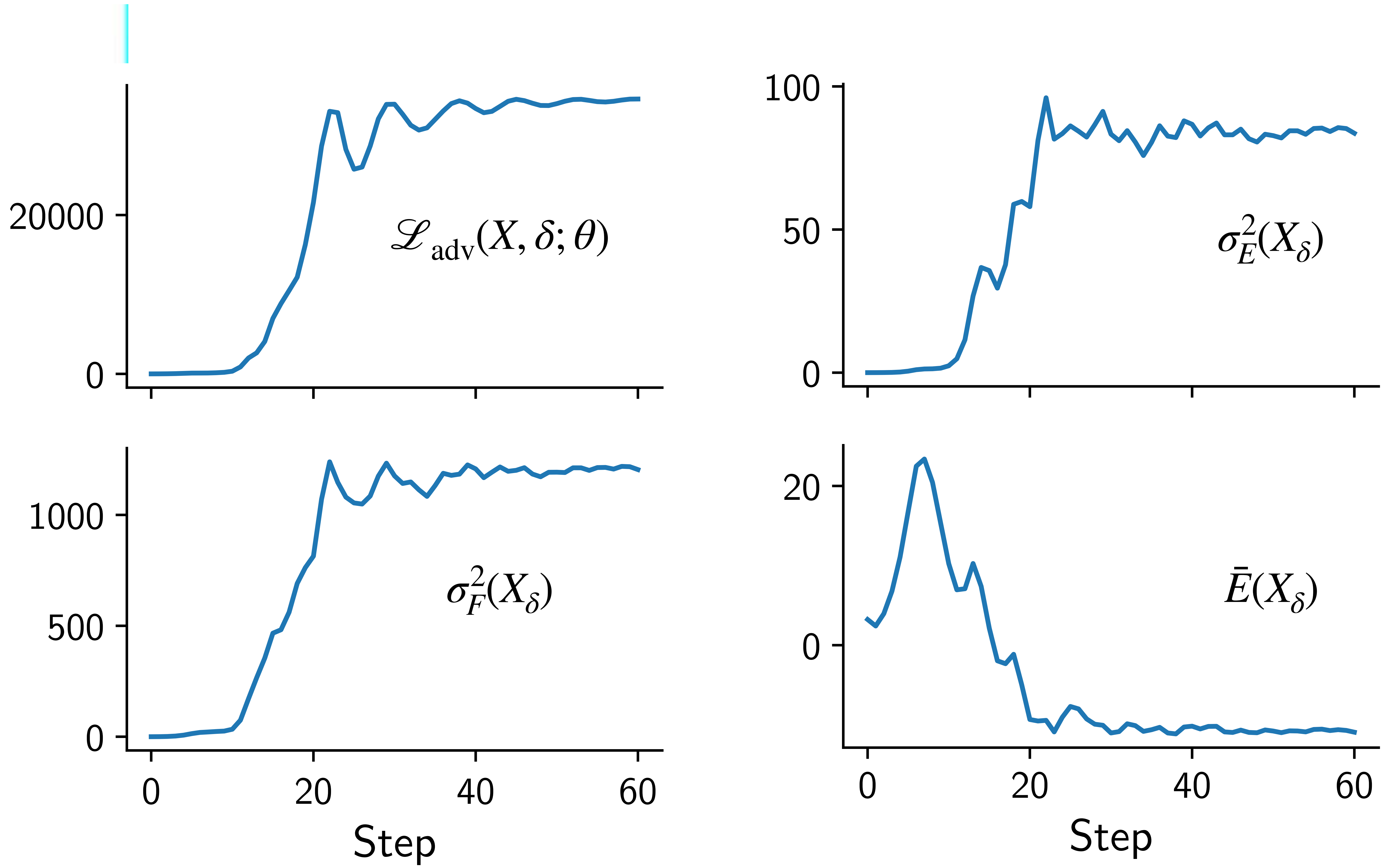
Adversarial attacks for 2D double well



D. Schwalbe-Koda et al. *Nat. Commun.* **12**, 5104 (2021)

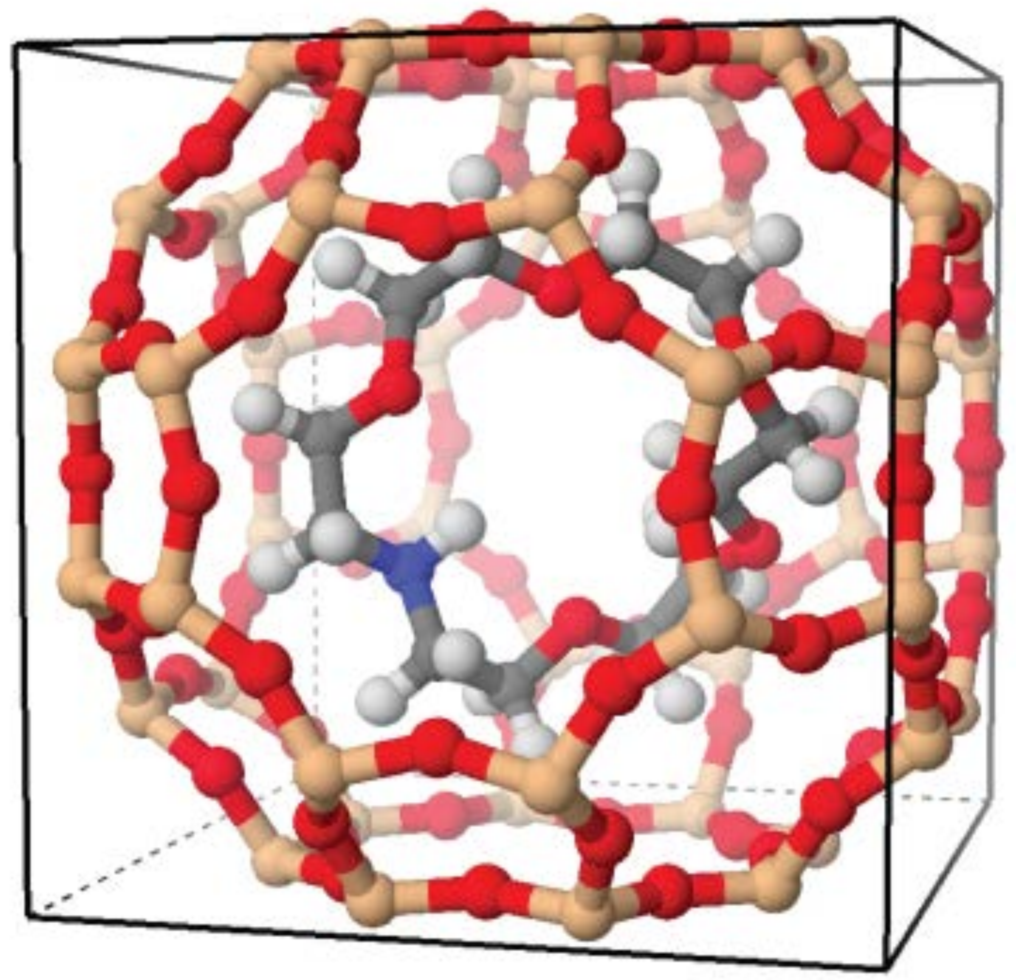
How adversarial attacks look like for molecules?





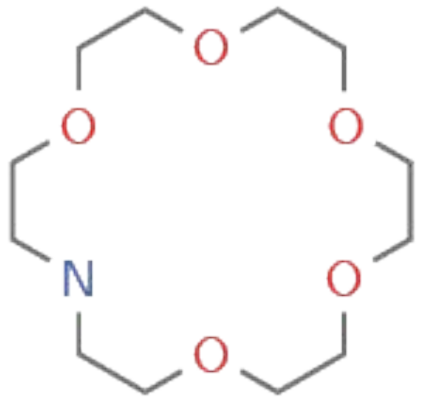
D. Schwalbe-Koda et al. *Nat. Commun.* **12**, 5104 (2021)

How efficient is the active learning with this technique?



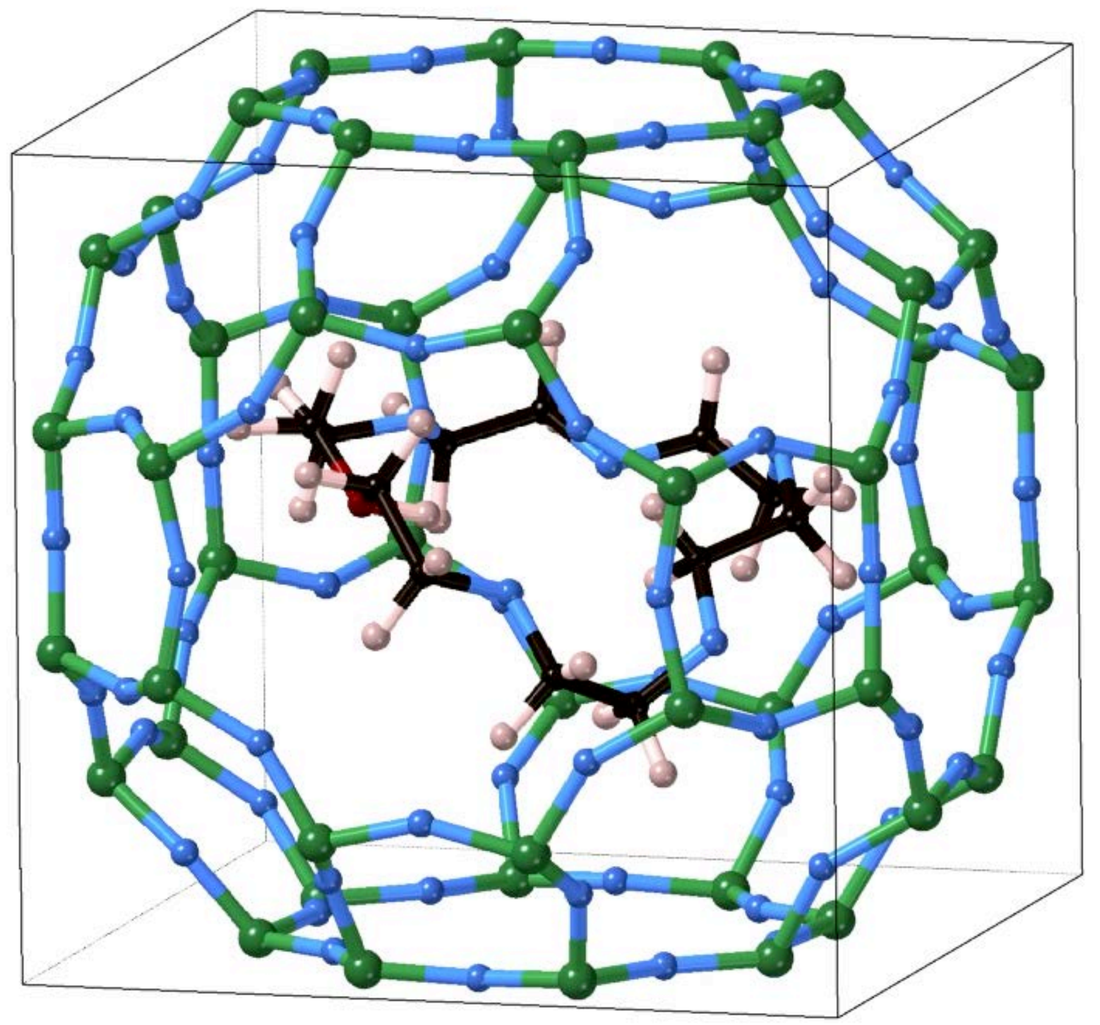
LTA

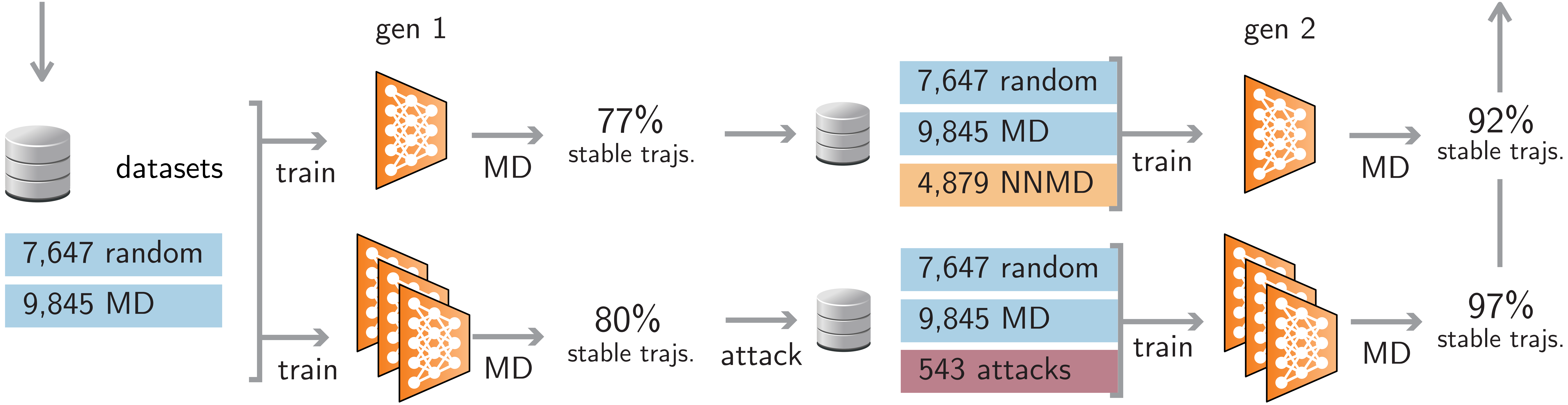
+



and 542 other zeolite-OSDA pairs

Tutorial: performing atomistic adversarial attacks





datasets

7,647 random

9,845 MD

train

gen 1

MD

77%
stable trajs.

7,647 random

9,845 MD

4,879 NNMD

train

gen 2

MD

92%
stable trajs.

train

MD

80%
stable trajs.

attack

7,647 random

9,845 MD

543 attacks

train

MD

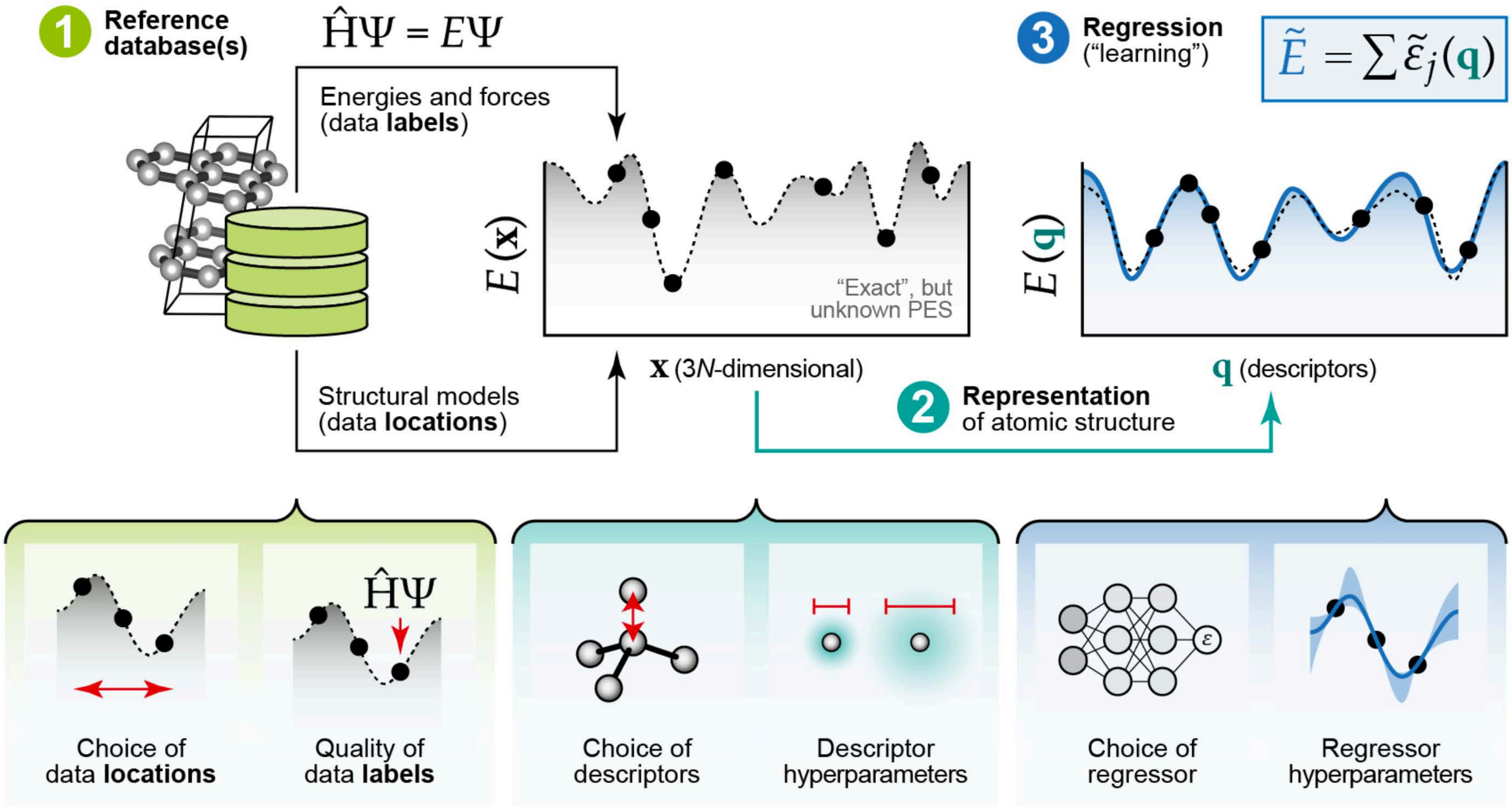
97%
stable trajs.

D. Schwalbe-Koda et al. *Nat. Commun.* **12**, 5104 (2021)

Summary

What did we learn today?

To summarize what we have learned today


1 Reference database(s)
$\hat{H}\Psi = E\Psi$
Energies and forces (data **labels**)
Structural models (data **locations**)
$E(\mathbf{x})$
"Exact", but unknown PES
$\mathbf{x}$ (3N-dimensional)
2 **Representation** of atomic structure
$\mathbf{q}$ (descriptors)
3 **Regression** ("learning")
$\tilde{E} = \sum \tilde{\varepsilon}_j(\mathbf{q})$
$E(\mathbf{q})$
$\hat{H}\Psi$
ε
Choice of data **locations**
Quality of data **labels**
Choice of descriptors
Descriptor hyperparameters
Choice of regressor
Regressor hyperparameters

J. Morrow et al. *arXiv*:2211.12484 (2022)

A few resources to learn more

Chemical Reviews **121** (16) (2021): Several reviews on ML for materials

Papers cited in this presentation: In-depth discussion on the advances of NNIPs and much more.

Andrew White's dmol.pub (https://dmol.pub/): interactive resources to learn more about ML, deep learning, and their applications to molecules and materials

Michael Nielsen's online book (http://neuralnetworksanddeeplearning.com/index.html): several explanations on the math/workings of neural networks

3blue1brown's videos on NNs: excellent visualizations and explanations on NNs (https://youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi)

I. Goodfellow et al. *Deep Learning.* MIT Press (2016): in-depth discussion of deep learning theory (https://www.deeplearningbook.org/)

Interatomic Potentials Enabled by Machine Learning

Daniel Schwalbe-Koda

Lawrence Livermore National Laboratory
https://dskoda.com, <dskoda@llnl.gov>

Image created with DALL-E 2

Lecture: 4th IKZ-FAIRmat Winter School
Jan. 24, 2023

LLNL-PRES-843937
Prepared by LLNL under Contract DE-AC52-07NA27344. LDRD # 22-ERD-055