CAMMLS 2025
Chemicals and Materials Machine Learning School

Newcastle University

Machine Learning Interatomic Potentials I

Ioan-Bogdan Magdău
ioan.magdau@ncl.ac.uk

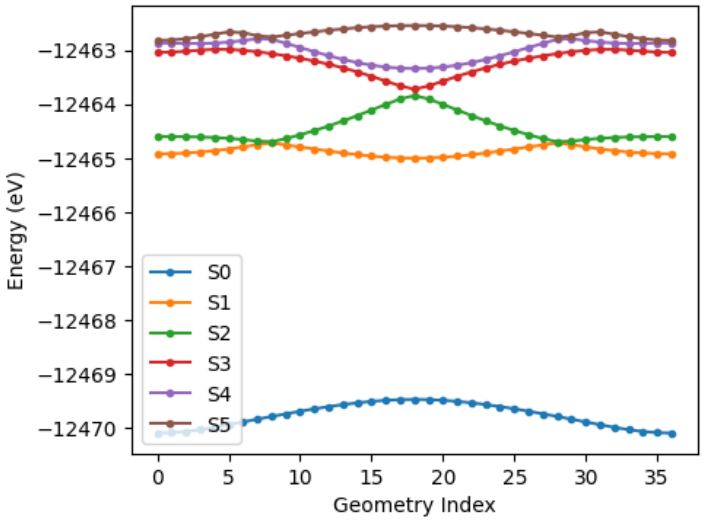
CAMMLS Spring School 2025

Few words about myself.

Lecturer in Computational Chemistry (3rd April 2023)
Newcastle University

Few words about myself.

Modelling Excited States Dynamics

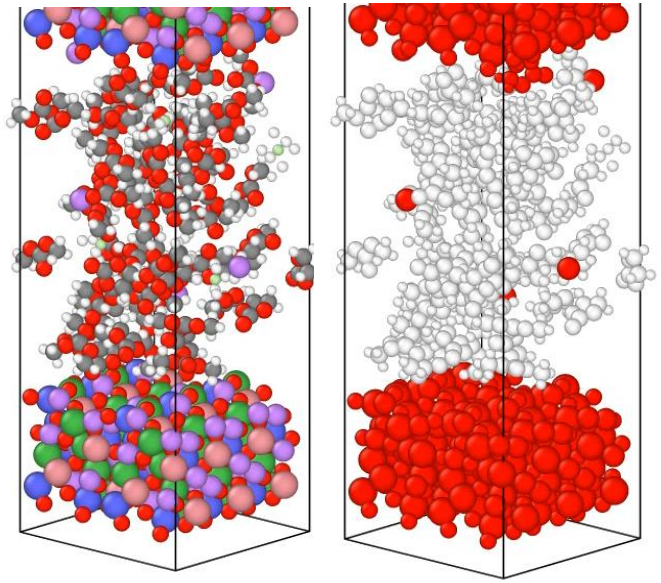


Adam O'Hare

Julien Eng

Leon Ashley

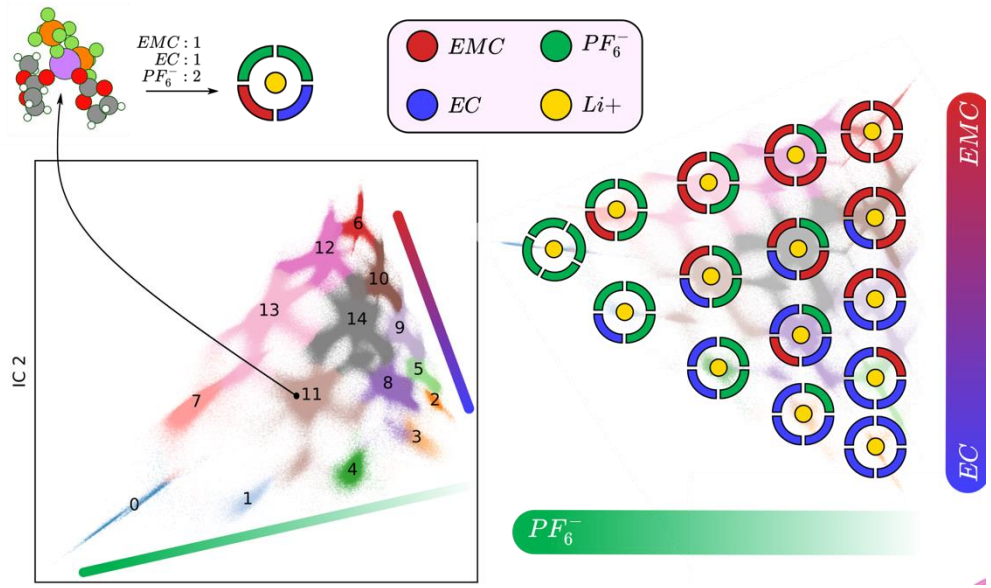
Foundational MLIPs for complex interfaces



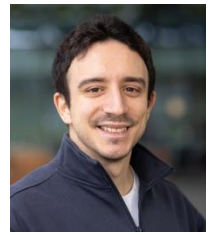
Samuel Niblett

Panagiotis Kourtis

Modelling ion transport in electrolytes



Zitong Wu

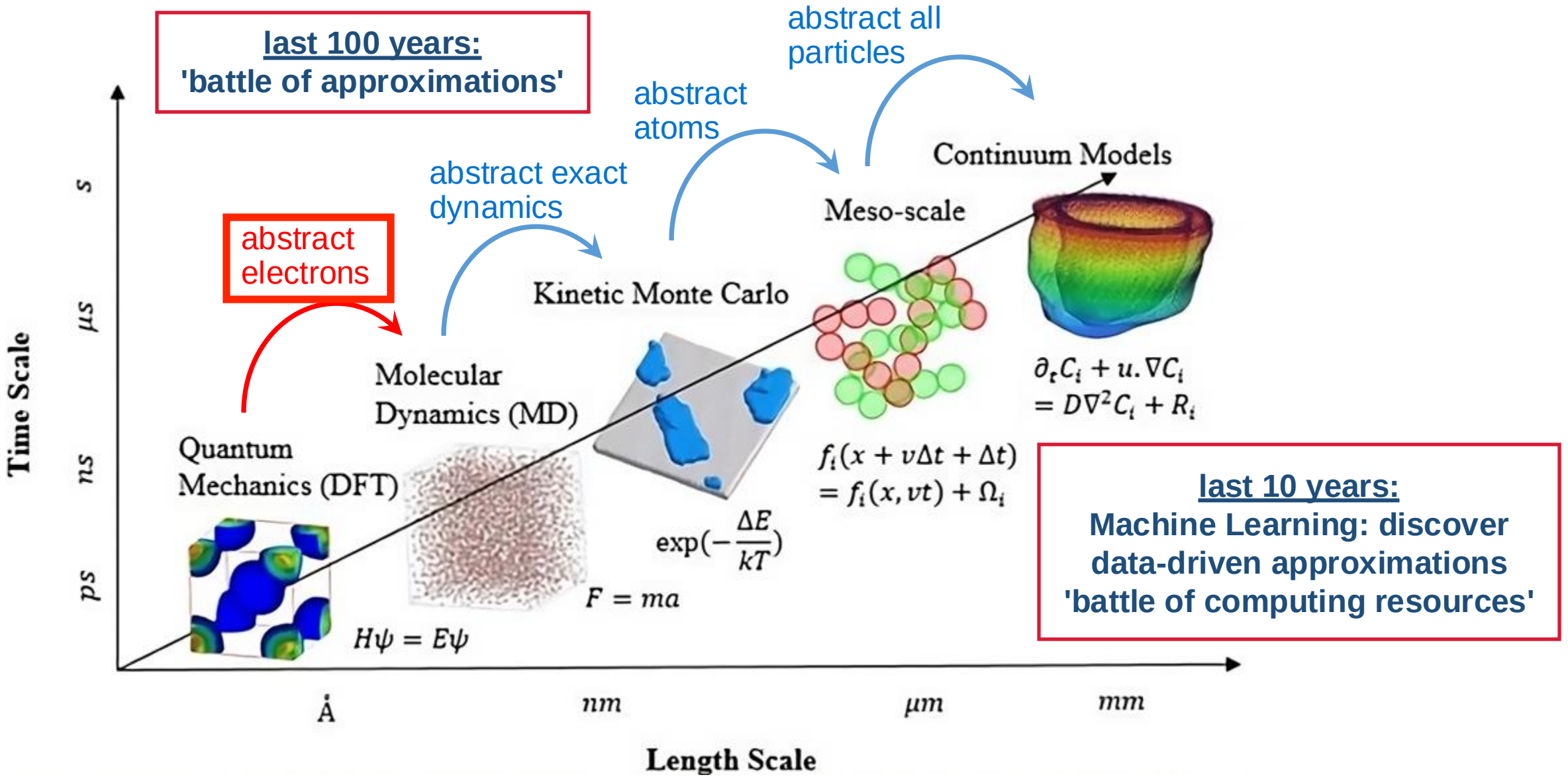


Andrea Gardin

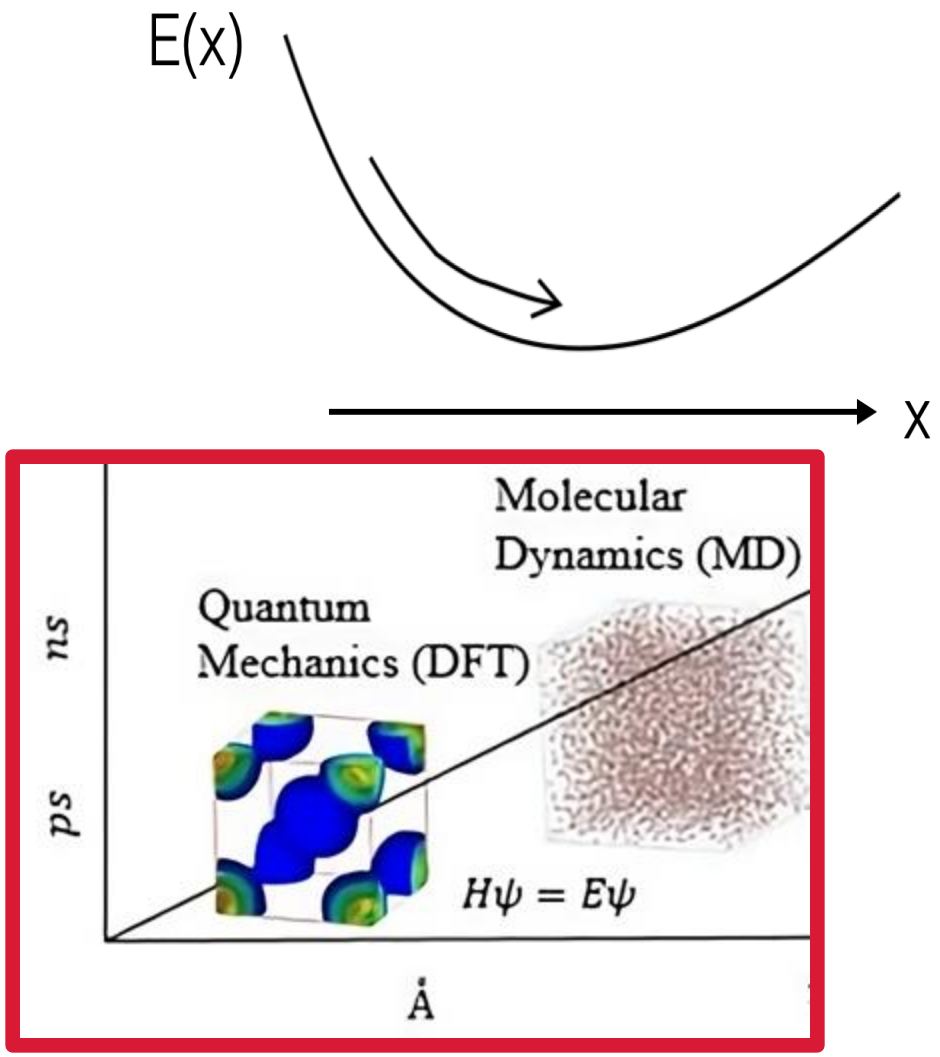
Lecture Outline

- Brief Intro to MLIPs: Context and Overview
- Anatomy of a potential: locality, E0s, forces
- Atomic Descriptors: Symmetry, Smoothness, Completeness
- Models Architectures: Linear, Kernel, MPNNs (MACE)
- Fitting and testing MLIPs

Multiscale Molecular Modelling - the Battle of Approximations



Force-field, Potential Energy, Molecular Dynamics



Molecular Dynamics is the method we use to explore the potential energy surface of the nuclei

We obtain forces as the gradient of the potential energy surface:

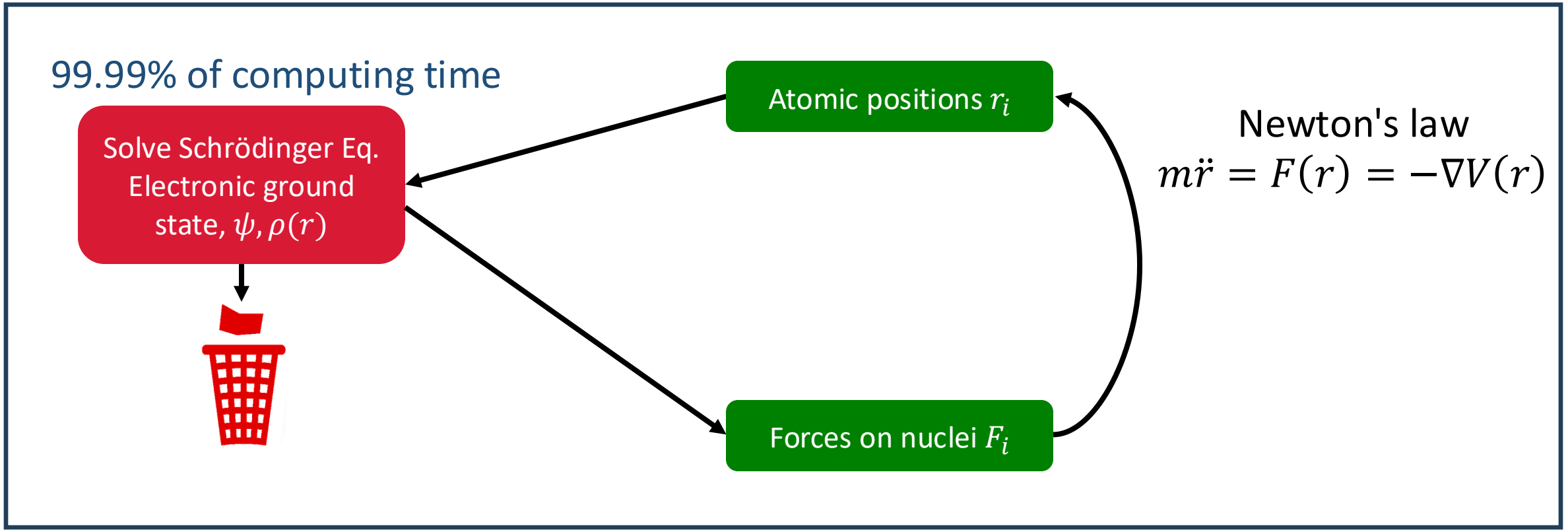
$$F = -\frac{dE(x)}{dx}$$

We integrate Newton's second low of motion to update positions

$$\mathbf{F}_i = m_i \mathbf{a}_i = m \frac{d^2 \mathbf{r}_i}{dt^2}$$

Ab Initio Molecular Dynamics

Solve very **expensive Schrodinger Equation** to obtain energies, forces
Update positions **very slightly** (1.0 femtosecond at a time)

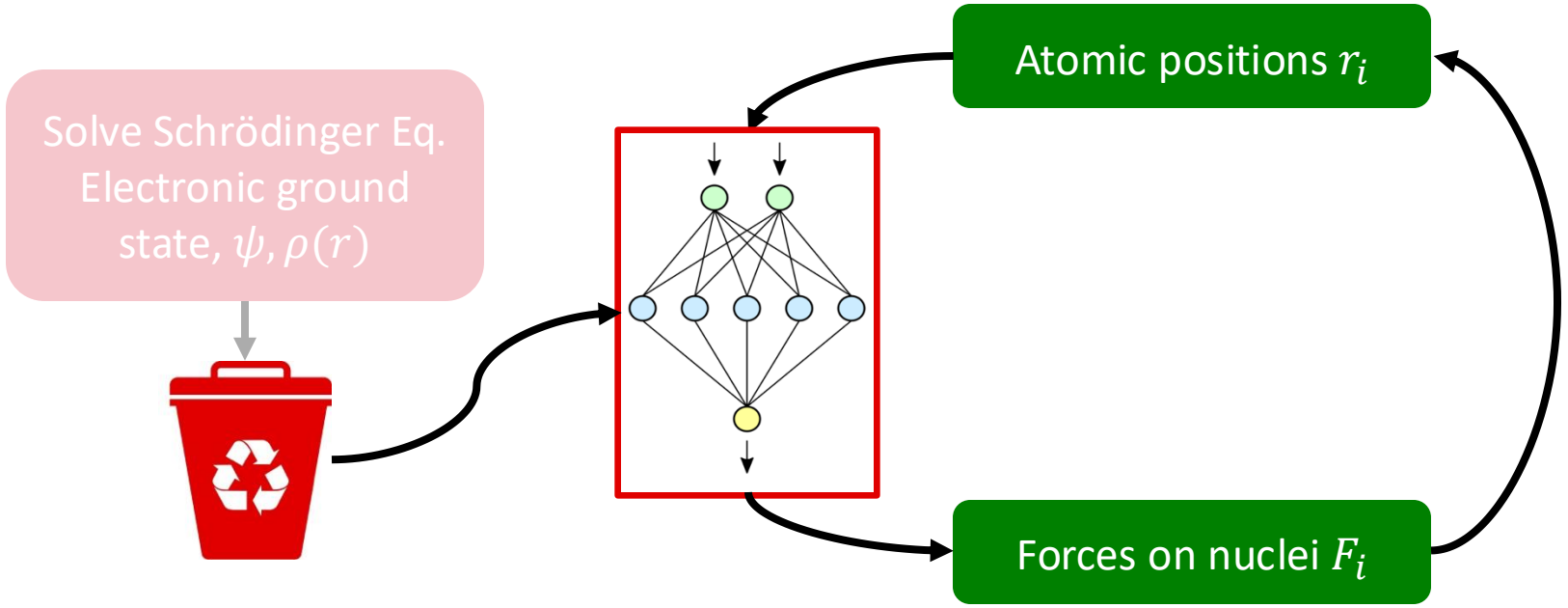


Discard old Schrodinger solution, solve again for very similar geometry.

Machine Learning Interatomic Potentials (MLIPs)

MLIPs learn the mapping from **geometry** to **energy, forces**

Instead of using simple functions (MM), they use **universal approximators**



Newton's law

$$m\ddot{r} = F(r) = -\nabla V(r)$$

Map from atomic positions to total energy

$$E \equiv V(r) : R^{3N} \rightarrow R$$

Can speed up the dynamics significantly! **1000-1M X**

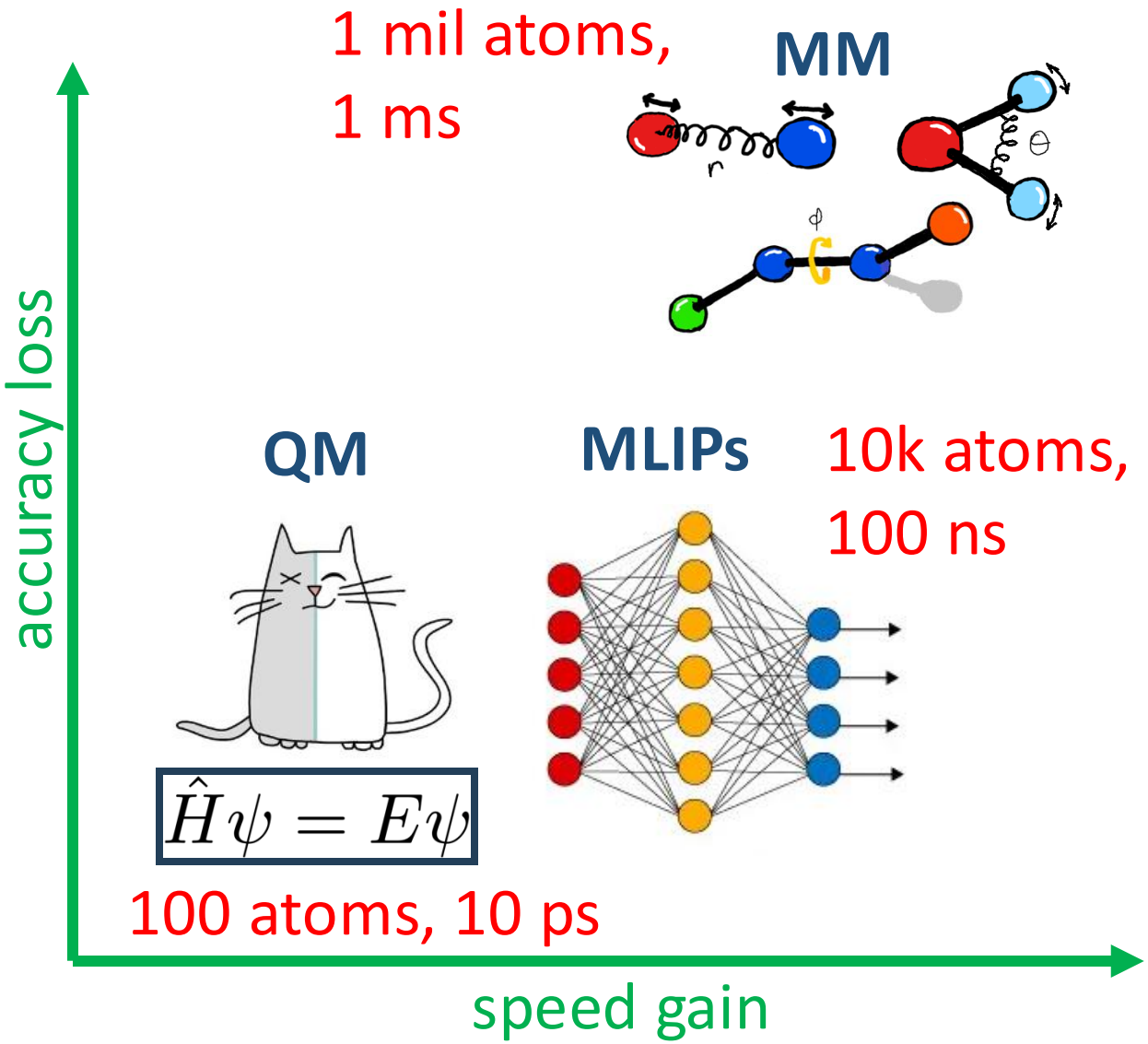
MLIPs generalize Molecular Mechanics (MM) Forcefields

The PES: $E = E(r_1, r_2, \dots)$

In QM, PES from solving the Schrodinger Equation (Born-Oppenheimer approximation)

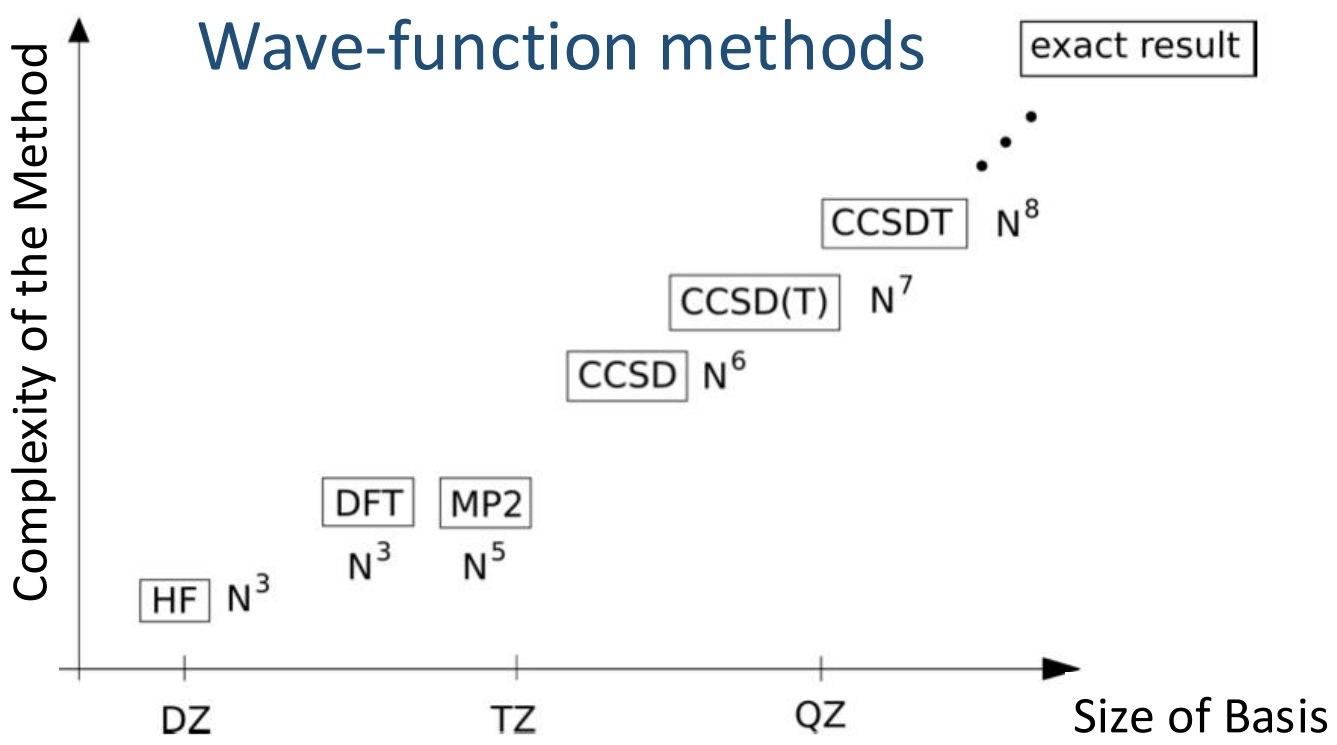
In MM, PES using empirical parameters: bond, angles, charges (**generally non-reactive**)

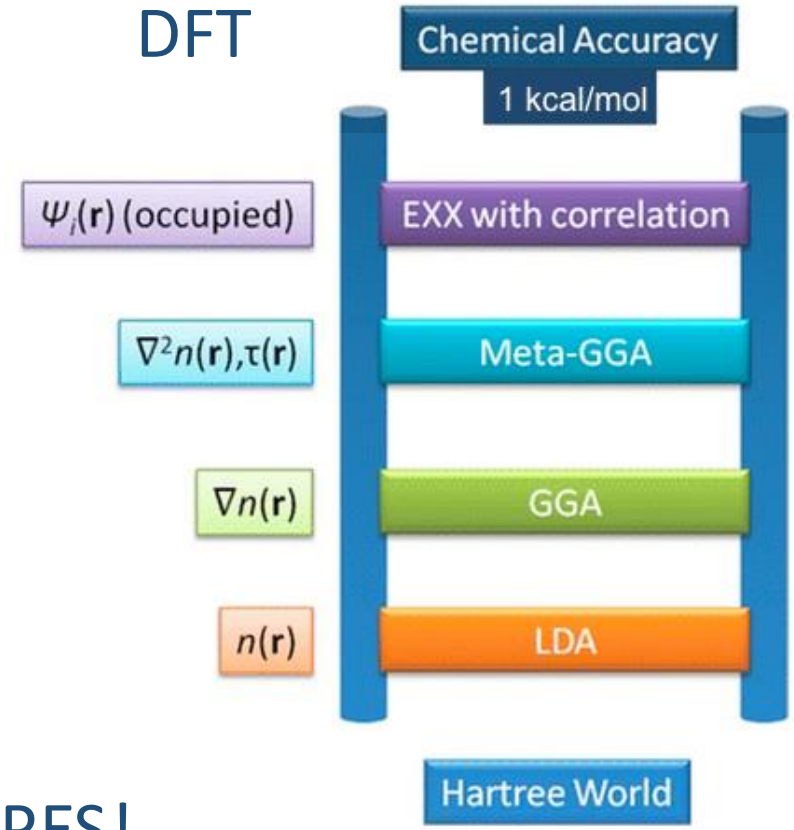
In MLIPS, PES interpolated from know geometries to new geometries (**capture chemical reactions**)



What's the speedup? It depends.

Not all QM methods are created equal:



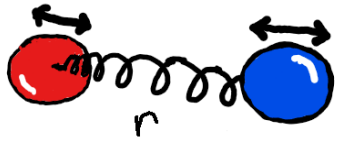


MLIP cost is constant regardless of accuracy of PES!

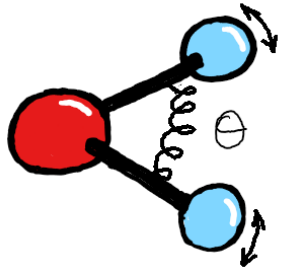
Training is more expensive of course!

Body-order in Molecular Mechanics

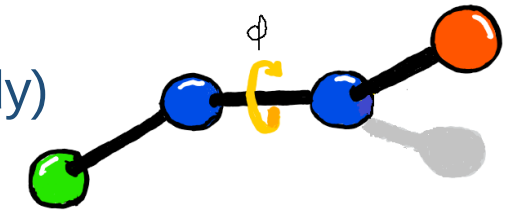
Bonds (2-body)



Angles (3-body)



Dihedrals (4-body)



Pairs (2-body)

$$U = \sum_{bonds} k_r(r - r_0)^2$$

$$+ \sum_{angles} k_\theta(\theta - \theta_0)^2$$

$$+ \sum_{dihedrals} k_\phi(1 + cos(n\phi + \phi_0))$$

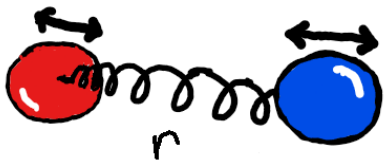
$$+ \sum_{i,j>i} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right]$$

$$+ \sum_{i,j>i} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

Short range covalent bonding

Long range non-bonded interactions

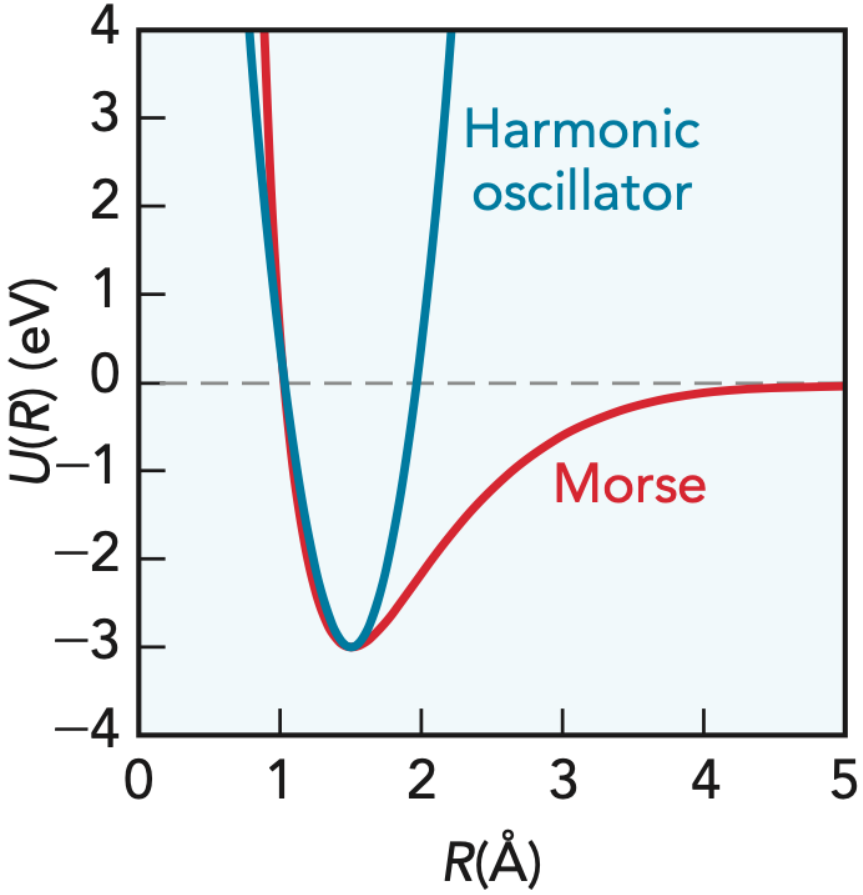
Is There Anything Missing?

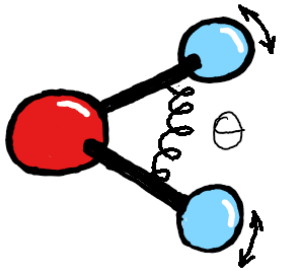


$$E = k_1 r^2 + k_2 r^3 + k_3 r^4 + \cdots$$

harmonic potential vs real potential

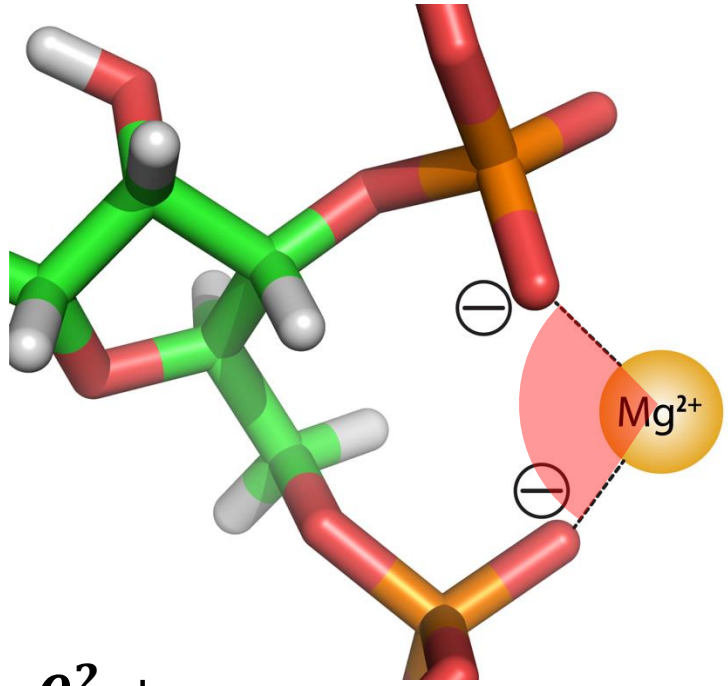
higher-order nonlinear terms to improve the approximation





bonded angles

how about non-bonded angles?

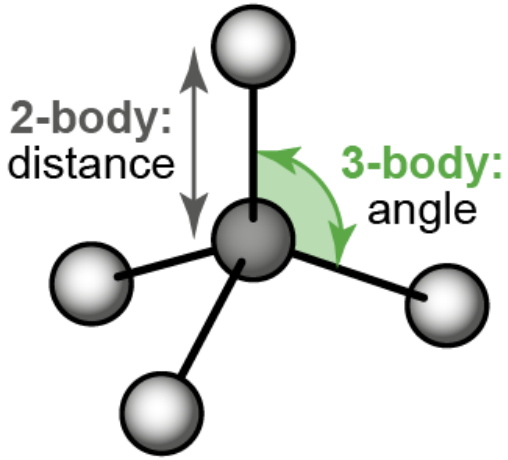


how about bond-angle interactions etc ...?

$$E = k_4 r\,\theta + k_5\, r\,\theta^2 + \cdots$$

MLIPs generalize MM Force Fields

a Conventional *N*-body terms



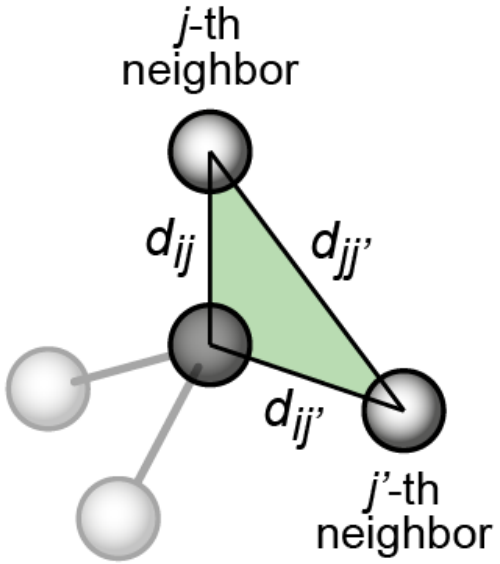
Bonds (2-body)

Angles (3-body)

Dihedrals / Impropers (4-body)

enumerated from the outset

b General 3-body descriptor



all Pairs within cutoff (2-body)

all Triangles (3-body)

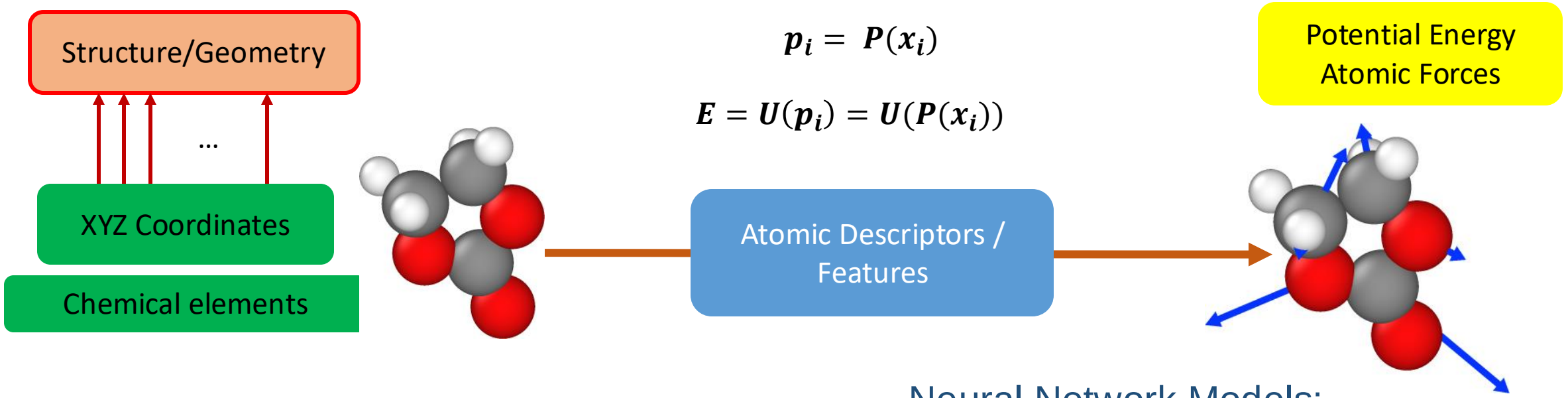
all Tetrahedra (4-body) ...

determined on the fly: inferred from data

Maximum Body-order of MLIP:

$$\frac{\partial^N P}{\partial x_{i_1}, \ldots, \partial x_{i_N}} \neq 0$$

General Anatomy of the MLIP



Linear Models:

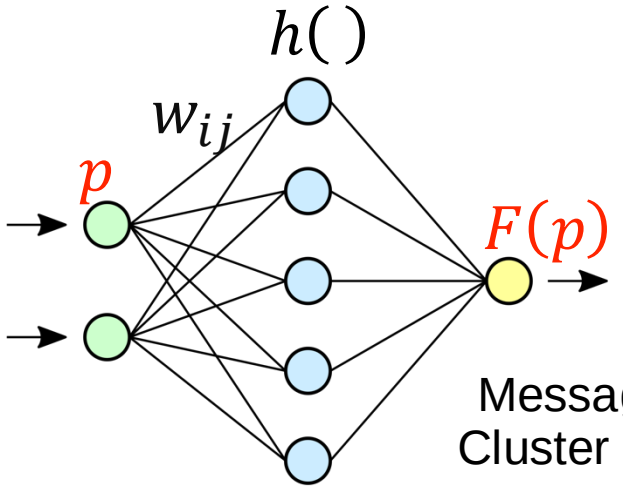
$$F(p) = \sum_i w_i \, p^i$$

Kernel Models:

$$F(p) = \sum_i w_i \, K(p, p_i)$$

Gaussian Approximation Potentials (GAP)

Neural Network Models:



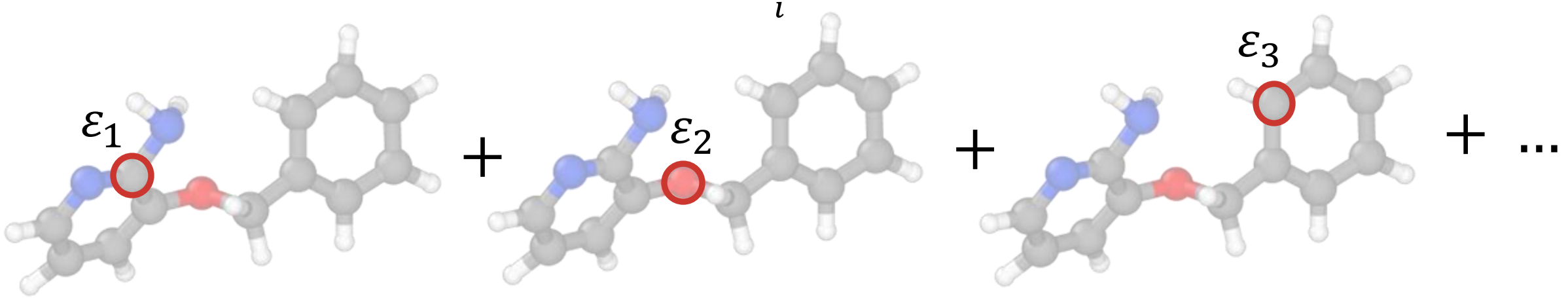
Message-passing Atomic Cluster Expansion (MACE)

Central Concepts in MLIPs: Locality of Quantum Mechanics

Sort-range ML models rely on energy decomposition and locality:

- total energy can be written as a sum of atomic energies:

$$E_{tot} = \sum_i \varepsilon_i$$



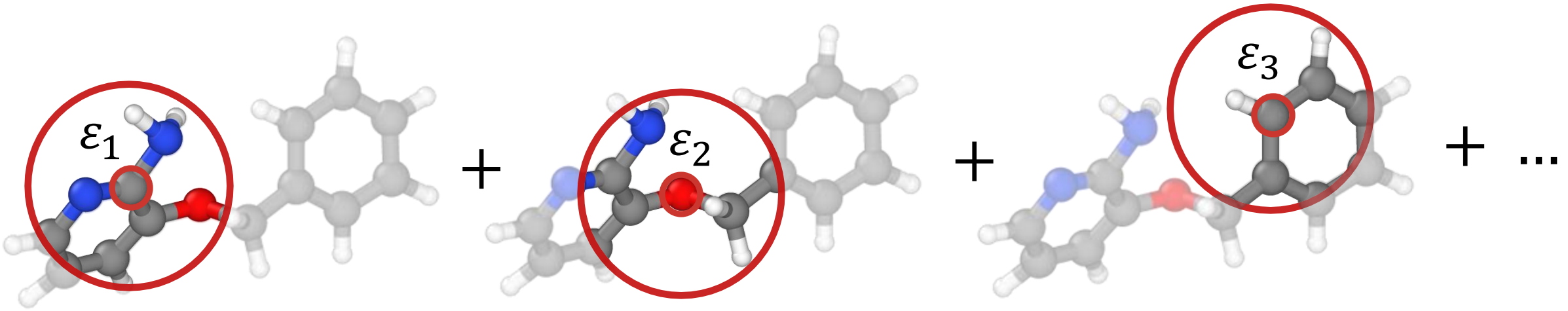
Central Concepts in MLIPs: Locality of Quantum Mechanics

Sort-range ML models rely on energy decomposition and locality:

- total energy can be written as a sum of atomic energies:

$$E_{tot} = \sum_i \varepsilon_i$$

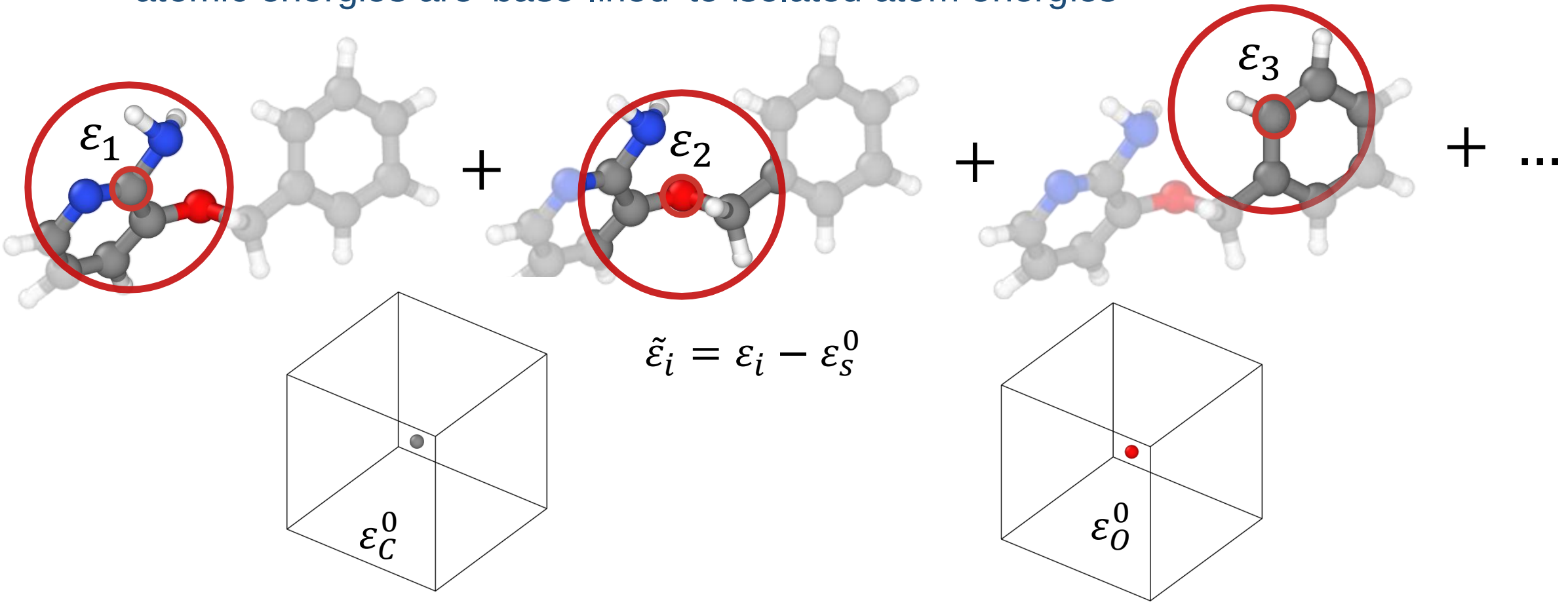
- atomic energies are a function of local environment (within a cutoff)



$$\varepsilon_i = \varepsilon_i(r < R_{cut})$$

Central Concepts in MLIPs: Isolated energies

- atomic energies are 'base-lined' to isolated atom energies



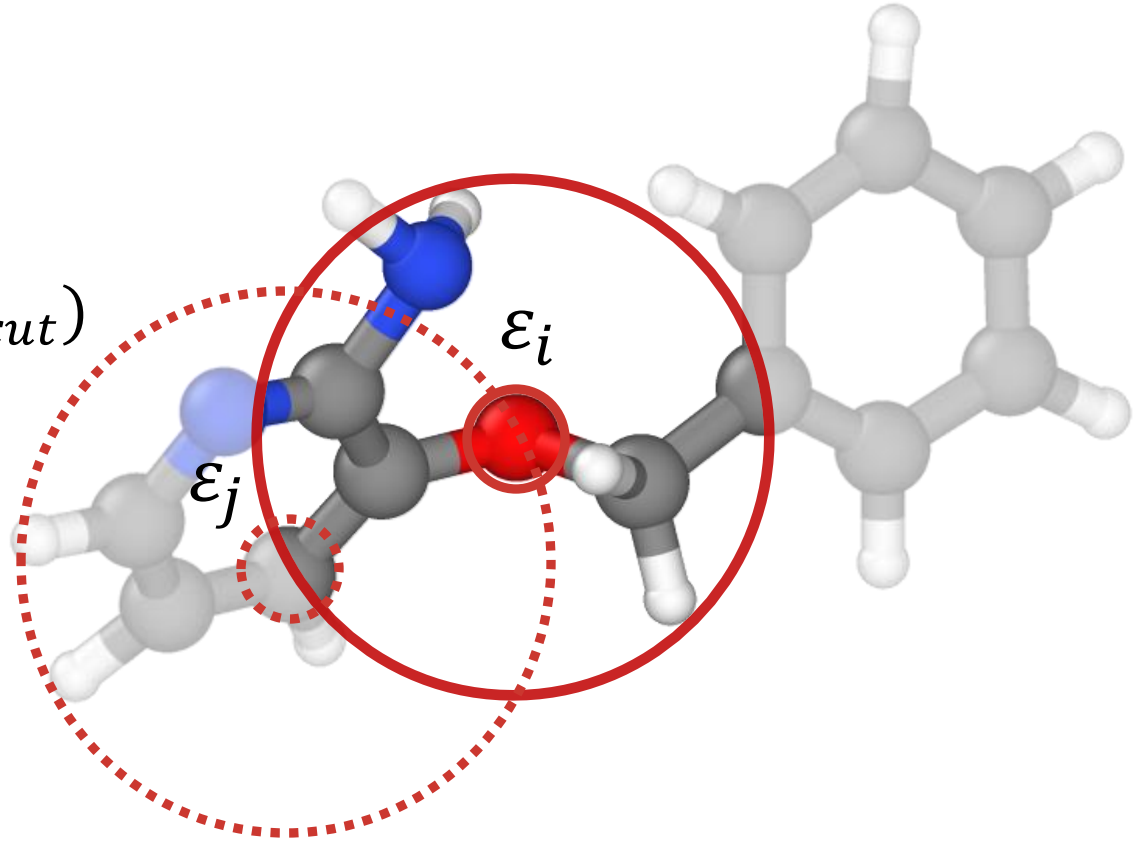
This ensures transferability across chemical compounds and a physically meaningful baseline in the absence of data for chemical reactions.

Central Concept in MLIPs: Calculating forces

- forces are derivatives of **total** energy:

$$F_i = \frac{\partial E_{tot}}{\partial r_i} = \sum_i \frac{\partial \varepsilon_i}{\partial r_i} + \sum_{j \neq i} \frac{\partial \varepsilon_j}{\partial r_i}$$

$$F_i = \sum_i \frac{\partial \varepsilon_i}{\partial r_i} (r < R_{cut}) + \sum_{j \neq i} \frac{\partial \varepsilon_j}{\partial r_i} (r < 2R_{cut})$$

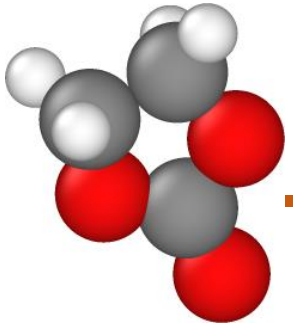


This ensures:

- energy conservation
- force equivariance
- extends effective interaction

Representation and Atomic Descriptors

Use Atomic Descriptors to represent Structure/Geometry



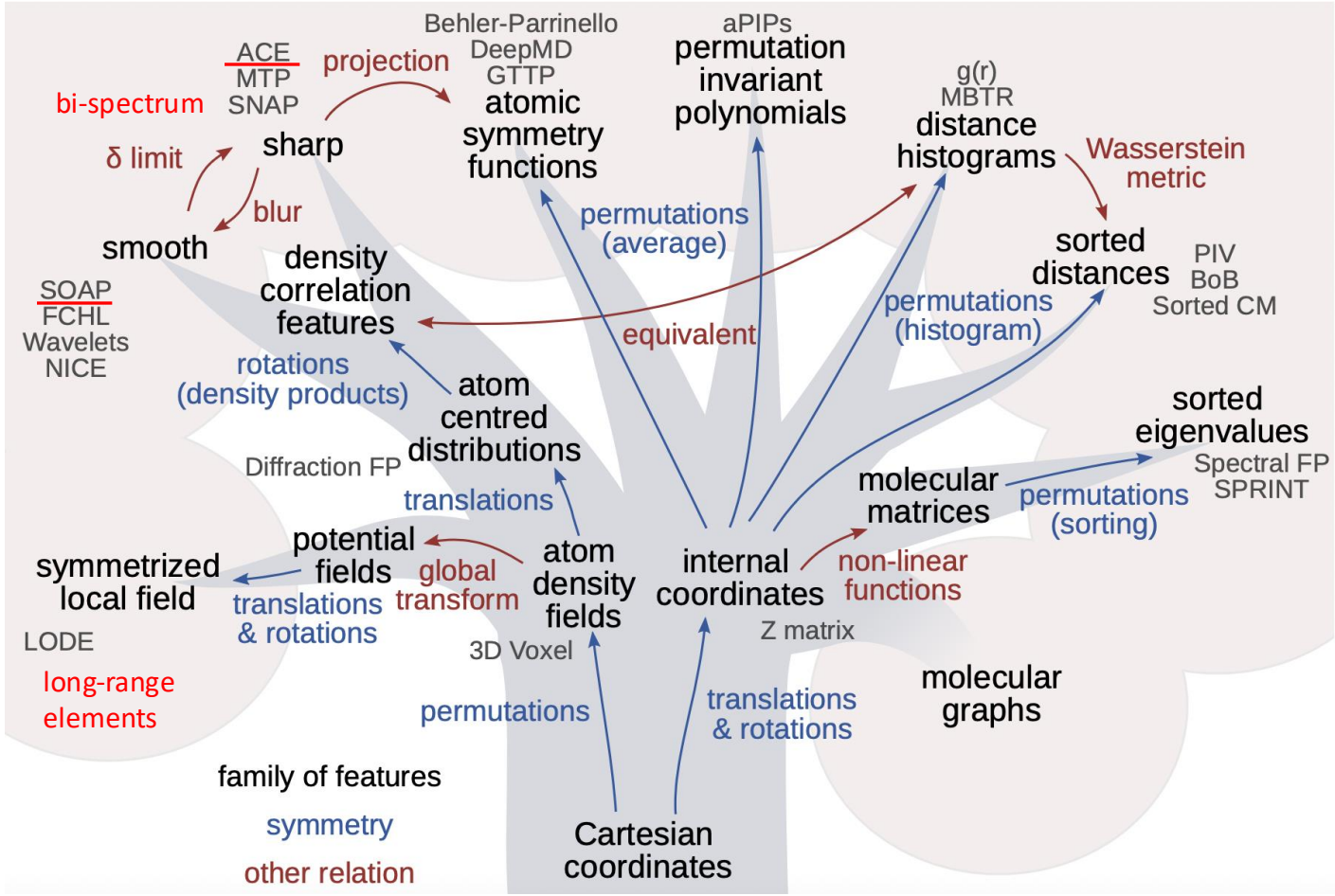
$$\boldsymbol{p_i} = \boldsymbol{P}(\boldsymbol{x_i})$$

Atomic Descriptors

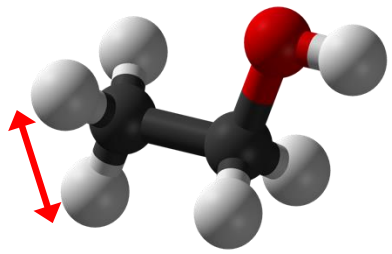
Properties

- symmetry: translational, rotational, permutational
- smoothness
- completeness
- correlation order (2, 3, 4-body)

Large and growing family tree of atomic descriptors!



Descriptor Symmetries: Permutation, Rotation, Translation

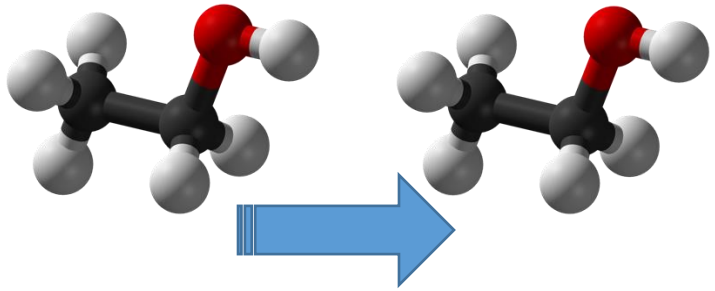


PERMUTATION

Permutation is a discrete symmetry of the energy!

Descriptors should be **Invariant** to **Permutations:**

$$P(x_i) = P(x_{\sigma(i)})$$



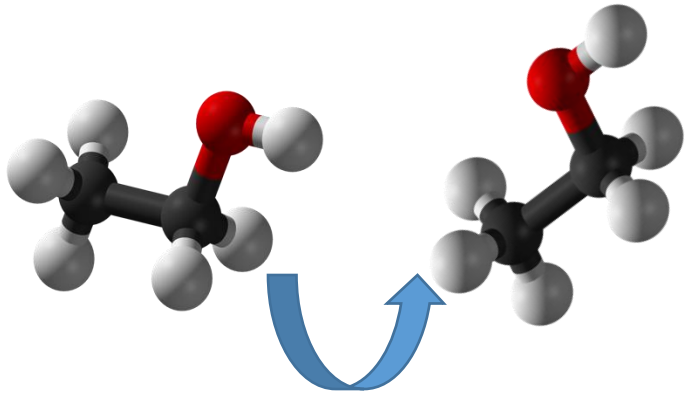
TRANSLATION

Translation is a continuous symmetry of the energy!

Infinitely many environments map to the same energy!

Difficult to `learn`, our descriptors require **Translational Symmetry**:

$$P(x_i) = P(x_i + T)$$



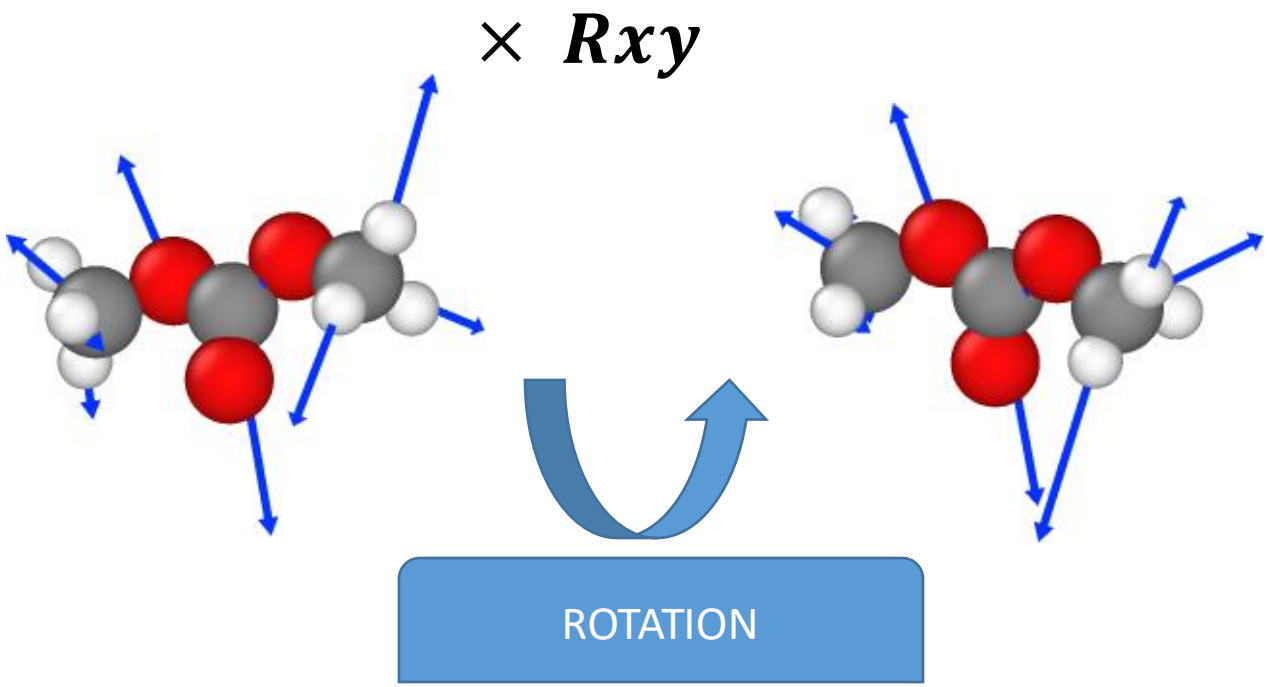
ROTATION

Rotation is also a continuous symmetry of the energy!

Descriptors should be **Invariant** to **Rotations:**

$$P(x_i) = P(x_i \times R)$$

Rotational Equivariance



What about properties that do change with rotations?

E.g. **Forces, Dipole vectors rotate with the coordinate frame!**

We want to avoid learning the transformation, instead we can make atomic descriptors **Equivariant with Rotations**:

$$\boldsymbol{P(x_i \times R) = P(x_i) \times R}$$

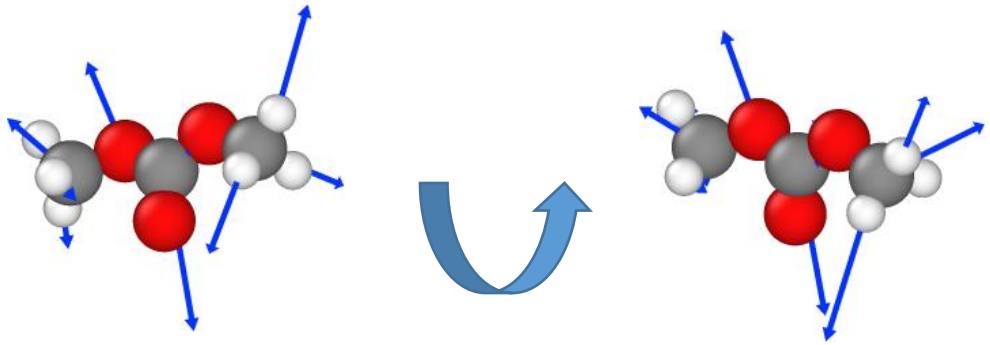
Question: can we fit equivariant forces with an invariant MLIP?

Let's define a rotated coordinate system (i.e. Rotate molecule in Ovito):

Energy is rotationally invariant:

$$x' = Rx$$

$$E(x) = E(x')$$



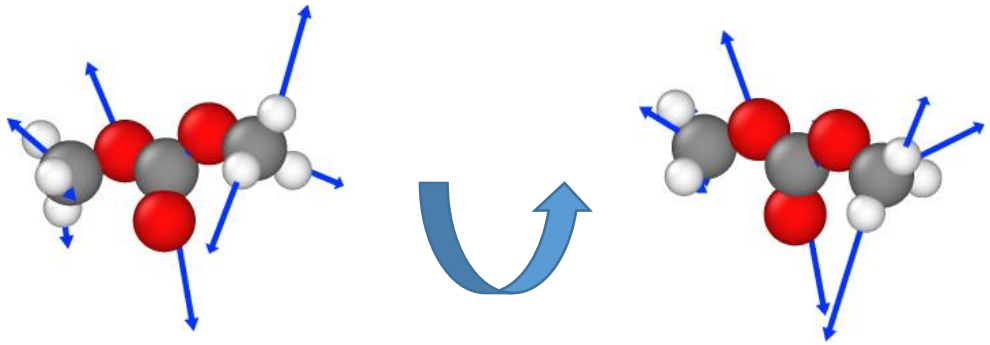
Question: can we fit equivariant forces with an invariant MLIP?

Let's define a rotated coordinate system (i.e. Rotate molecule in Ovito):

$$x' = Rx$$

Energy is rotationally invariant:

$$E(x) = E(x')$$



Forces in the original and rotated frame:

$$F(x) = -\frac{\partial E(x)}{\partial x} \qquad F(x') = -\frac{\partial E(x')}{\partial x'}$$

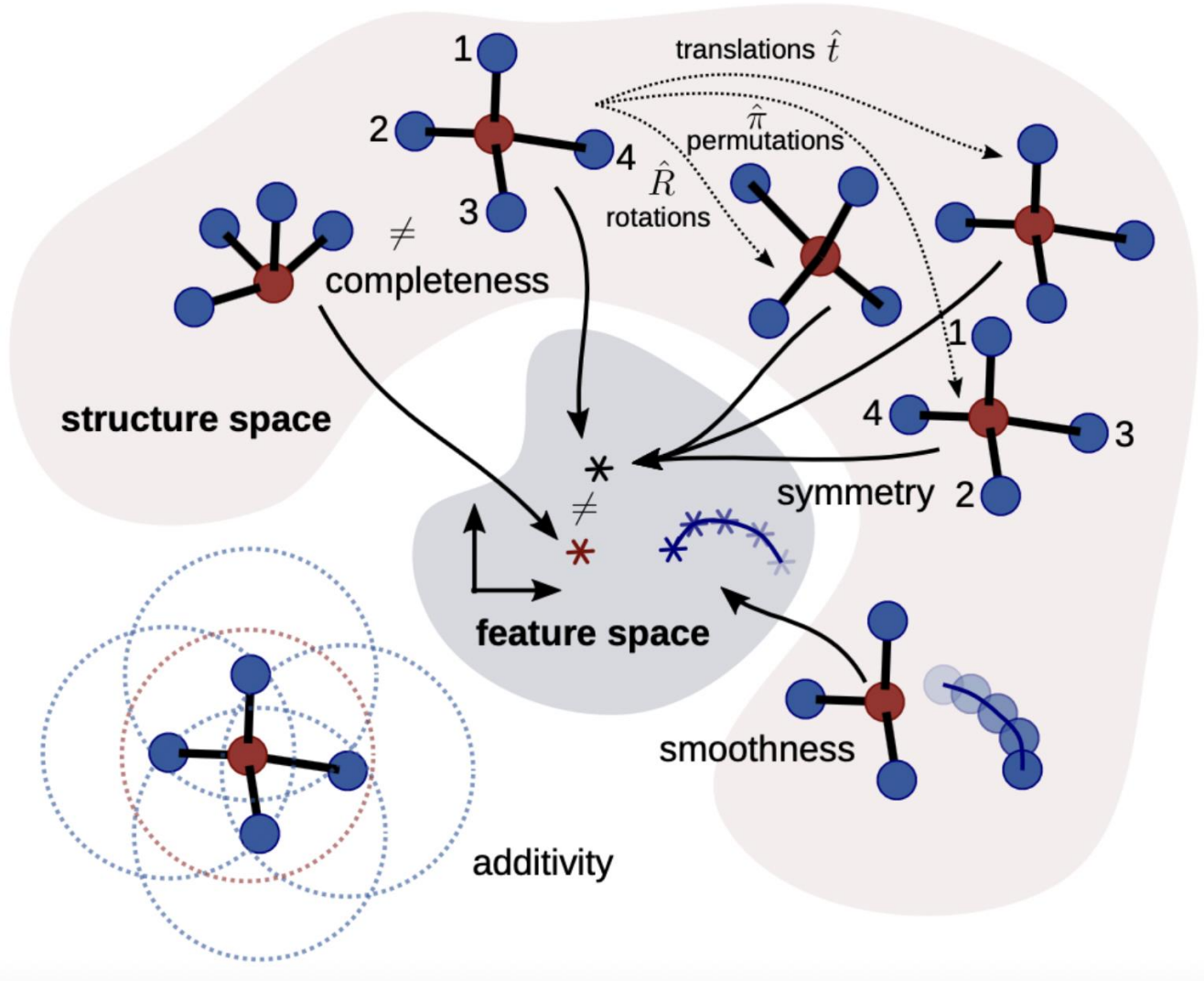
The relation between the forces:

$$\frac{\partial E(x)}{\partial x} = \frac{\partial E(x)}{\partial x'}\frac{\partial x'}{\partial x} = \frac{\partial E(x')}{\partial x'}R$$

$$F(x) = RF(x')$$

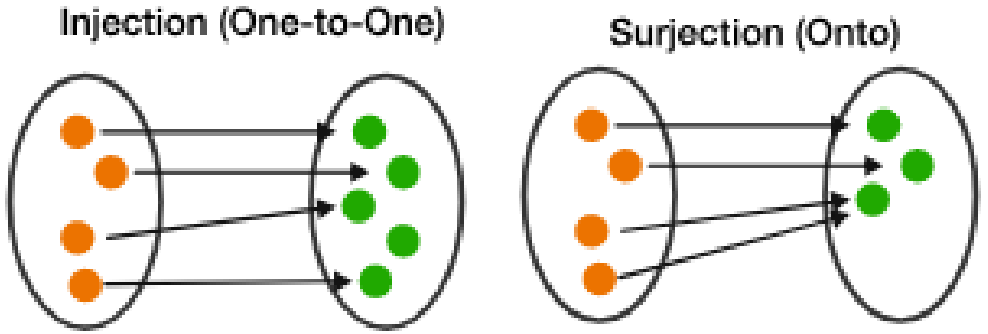
YES! If we obtain forces as the **gradient of an invariant energy**, we ensure they are equivariant! This will work for all vectors that can be expressed as gradients of a scalar field.

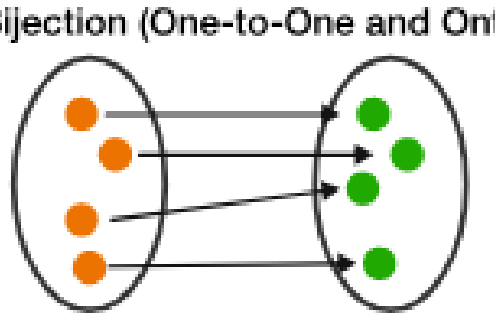
Other Properties of a 'Good' Descriptor: Completeness



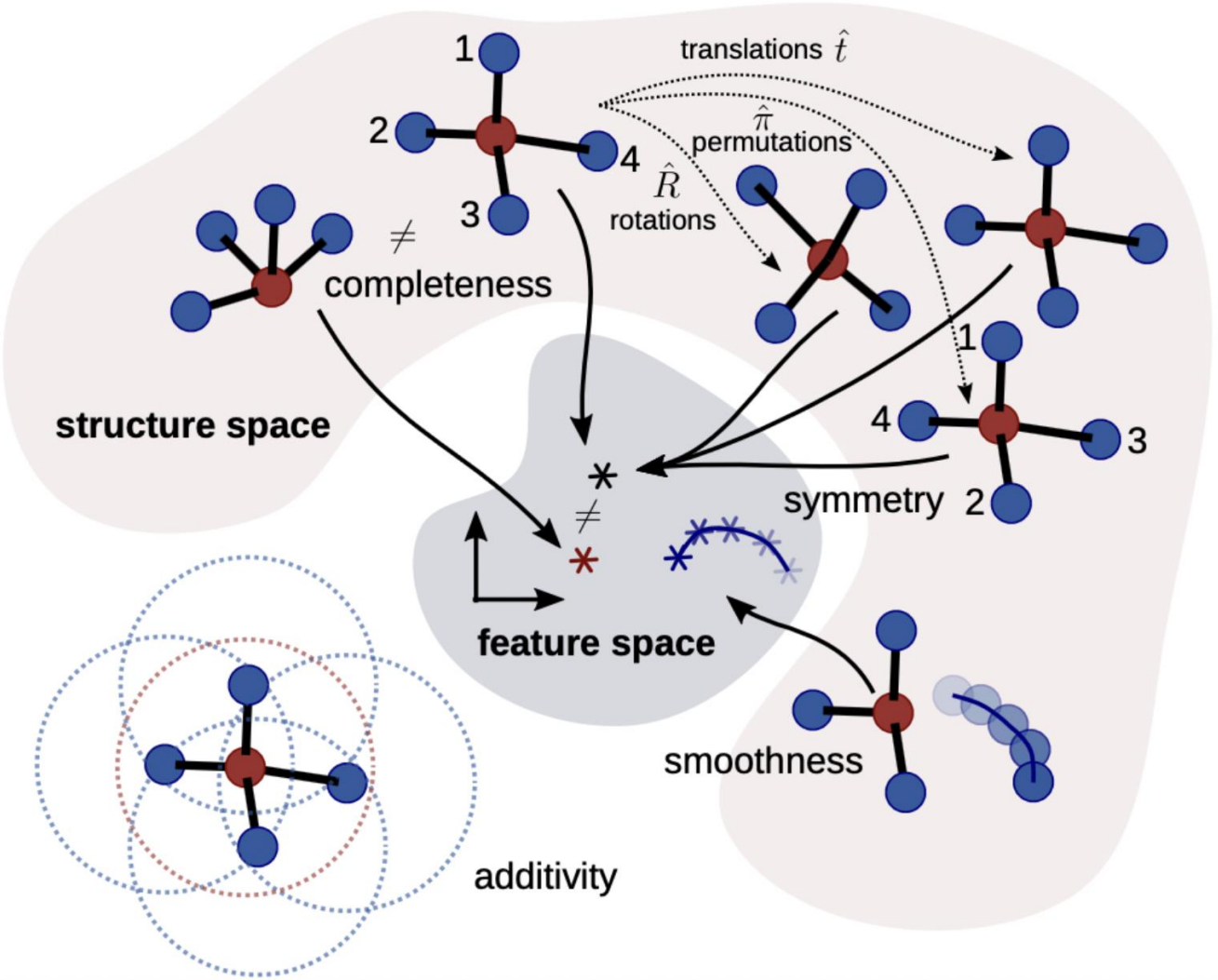
Descriptor Space ≫ Geometry Space

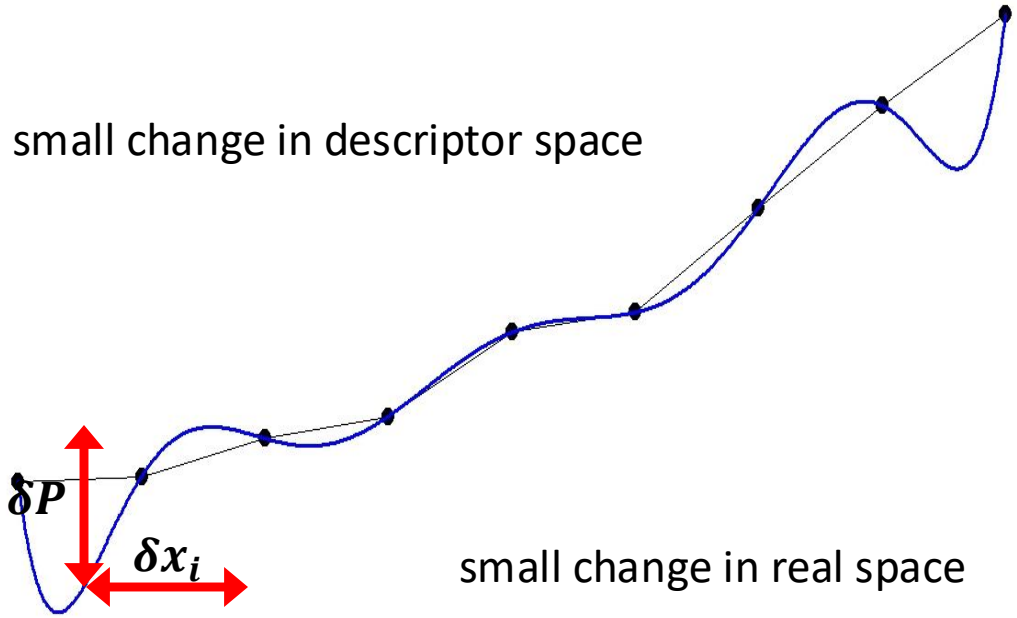
- **many** P **do not** map onto real x_i
- **all** x_i have to map onto a P
- **different** x_i must map onto different P





Other Properties of a 'Good' Descriptor: Smoothness





descriptors need to change smoothly with geometry:

energy is smooth!

so should their arbitrary derivatives:

forces, Hessians are smooth!

Forces and energy conservation

The total energy must be conserved:

$$E = \frac{1}{2}mv^2 + U = const$$

Which means its time derivative must zero:

$$\frac{dE}{dt} = mv \cdot \frac{dv}{dt} + \frac{dU}{dt} = F \cdot v + \frac{\partial U}{\partial x} \cdot \frac{dx}{dt} = F \cdot v + \frac{\partial U}{\partial r} \cdot v = 0$$

It follows that forces must obtained as the gradient of the total energy:

$$F = -\frac{\partial U}{\partial r}$$

Examples: SOAP

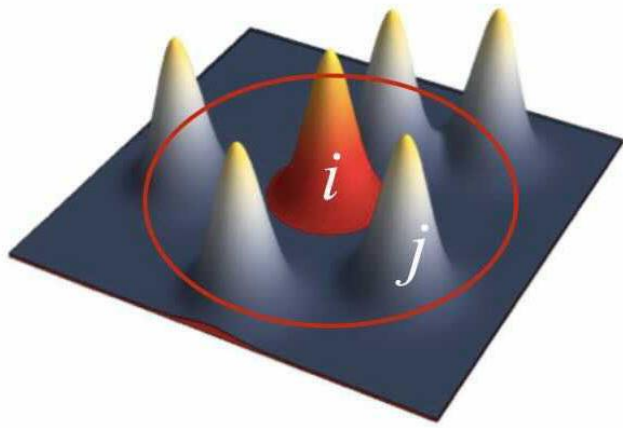
Atomic density **centered** on atom i:

$$\rho^i(\boldsymbol{r}) = \sum_j \exp\left(\frac{-\left|\boldsymbol{r} - \boldsymbol{r}_{ij}\right|^2}{2\sigma_{atom}^2}\right) f_{cut}(\left|r_{ij}\right| < R_{cut})$$

- **permutational invariance**
- **translational invariance**

Expand into **radial / angular** basis:

$$\rho^{i\alpha}(\boldsymbol{r}) = \sum_{nlm} c_{nlm}^{i\alpha}\, R_n(r) Y_{lm}(\hat{r})$$



Power spectrum (the 'feature' vector):

$$P = p_{nn'l}^i = \frac{1}{\sqrt{2l+1}} \sum_m c_{nlm}^i\, c_{n'lm}^i$$

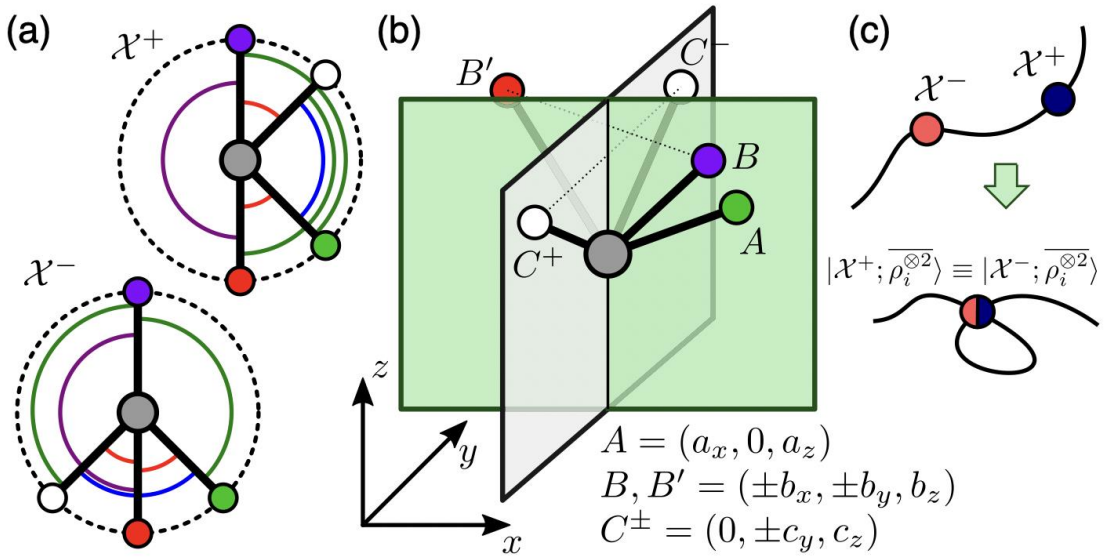
- **rotational invariance**

SOAP is a 3-body Descriptor

3-body Degeneracy:

Two different environments (4 neighbors): same SOAP

SOAP is incomplete!



SN Pozdnyakov, MJ Willatt, AP Bartók, C Ortner, G Csányi, and M Ceriotti PRL **125**, 166001, 2020

body order ↔ minimum neighborhood degeneracy!

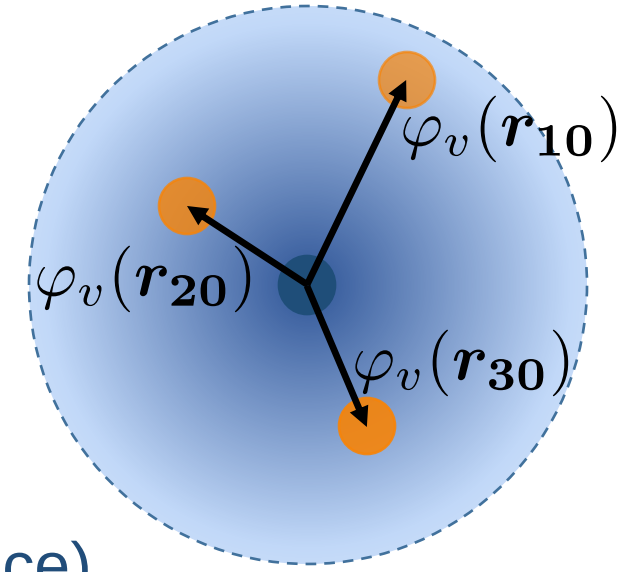
4-body: minimum 7 neighbors proven!

ACE: arbitrary body-order expansion!

Example: ACE - Body-ordered Descriptors

Atomic Cluster Expansion (ACE) as a systematic approach:

One–particle basis, atom-centred (translational invariance)



Radial Angular

$$\varphi_{nlm,z_i z_j}(\boldsymbol{r}) = R_{nl,z_i z_j}(r_{ji}) Y_l^m(\hat{\boldsymbol{r}})$$

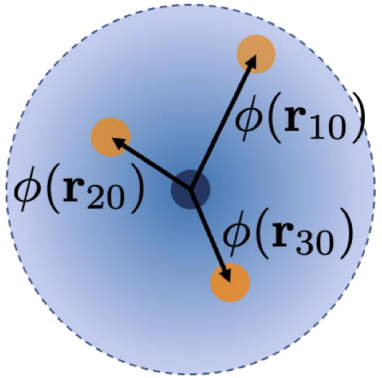
Evaluate for each neighbour and sum (permutation invariance)

$$A_{iv} = \sum_{j \in \mathcal{N}(i)} \varphi_v(\sigma_j, \sigma_i) \quad \text{(2-body)}$$

Phys. Rev. B **100**, 249901 (2019)

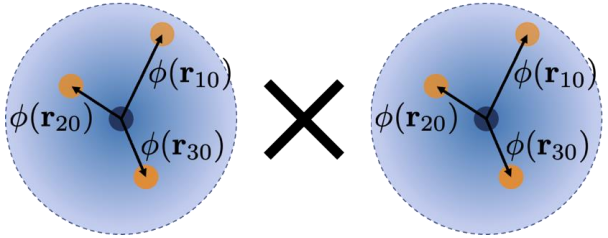
ACE: Systematic Body-ordered Expansion

Construct the product basis ($\nu + 1$ body terms):



$$\mathbf{A}_{i,\mathbf{v}} = \prod_{\xi=1}^{\nu} A_{i,v_\xi} \qquad (\nu + 1 \text{ body})$$

<u>Symmetrize</u> the product atomic-basis using the



Clebsch–Gordan coefficients (rotational invariance)

$$\boldsymbol{B}_{i,\boldsymbol{v}} = \sum_{\boldsymbol{w}} \mathcal{C}_{\boldsymbol{v}\boldsymbol{w}} \boldsymbol{A}_{i,\boldsymbol{v}}$$

<u>Phys. Rev. B **100**, 249901 (2019)</u>

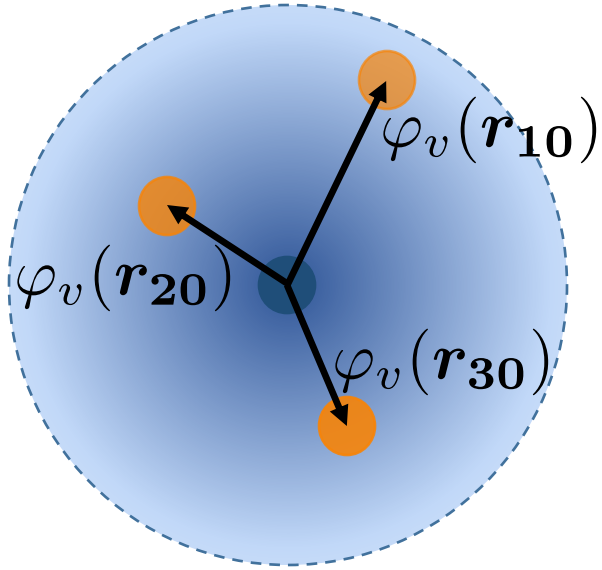
Linear ACE

ACE inspired from cluster-expansions of the energy function:

$$E_i = E_i^{(0)} + \sum_v \tilde{c}_v^{(1)} A_{iv} + \sum_{v_1 v_2}^{v_1 \geq v_2} \tilde{c}_{v_1 v_2}^{(2)} A_{iv_1} A_{iv_2} + \sum_{v_1 v_2 v_3}^{v_1 \geq v_2 \geq v_3} \tilde{c}_{v_1 v_2 v_3}^{(3)} A_{iv_1} A_{iv_2} A_{iv_3} + \ldots$$

The problem becomes linear:

$$E = \sum_{i,\boldsymbol{v}} c_{i,\boldsymbol{v}} \boldsymbol{B}_{i,\boldsymbol{v}}$$



... and we know how to solve it:

$$L = \sum_j \| E(X_j) - \mathbf{B}(X_j)^T \mathbf{w} \|^2$$

ACE is a precursor to MACE

Dealing with chemical species: embedding

From categorical space to continuous space

Embedding vs one-hot encoding representations

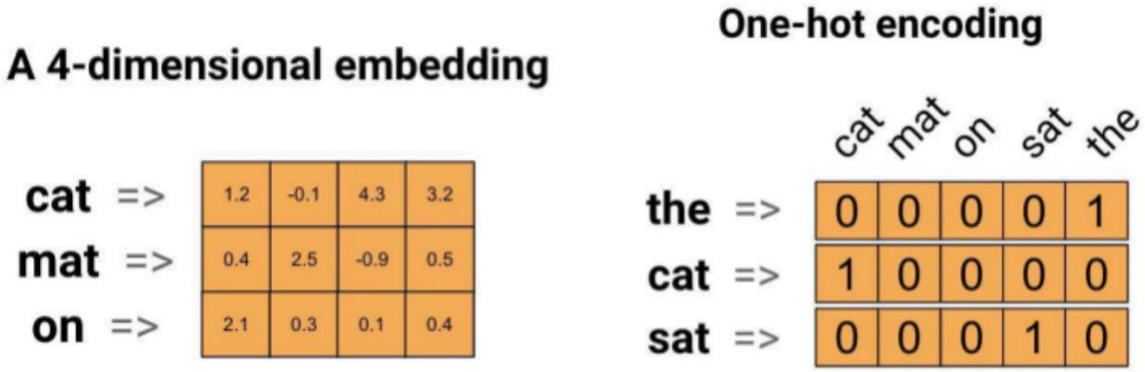
4-Dimensional Embedding
- dense, low-dimensional representations of data
- each word (e.g., "cat," "mat," "on") is represented by a 4-dimensional vector with learned values

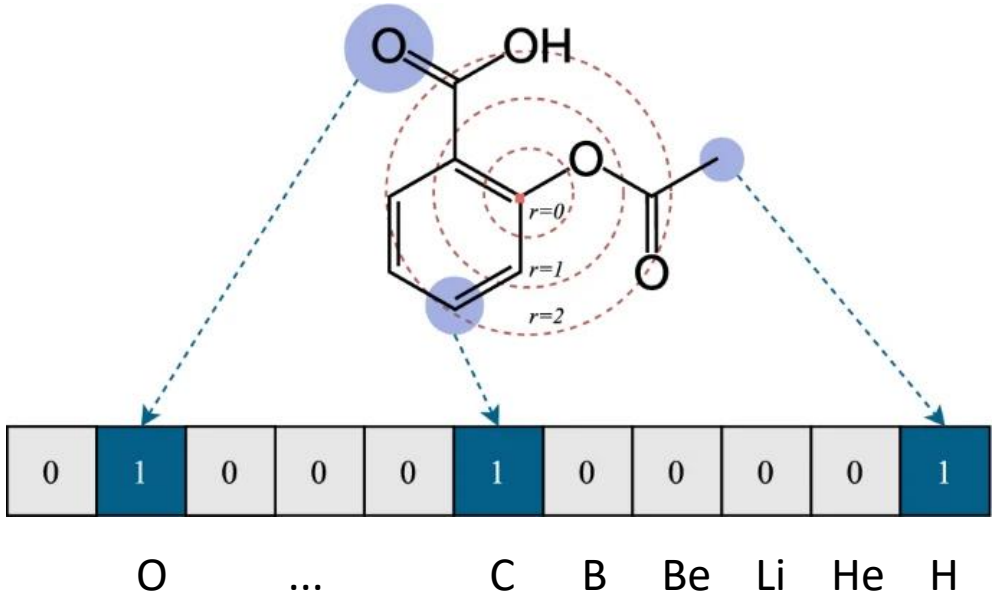
One-Hot Encoding
- categorical data as binary vector
- each word is represented by a vector where:
  - One position is "1" (indicating the word).
  - All others are "0."

  "the" → [0, 0, 0, 0, 1]

  "cat" → [1, 0, 0, 0, 0]
- high dimensionality: one vector per word
- no notion of similarity between "cat" and "dog."

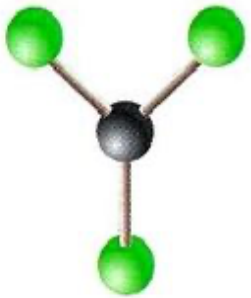
MACE starts with **one-hot encoding** and learns an **embedding**



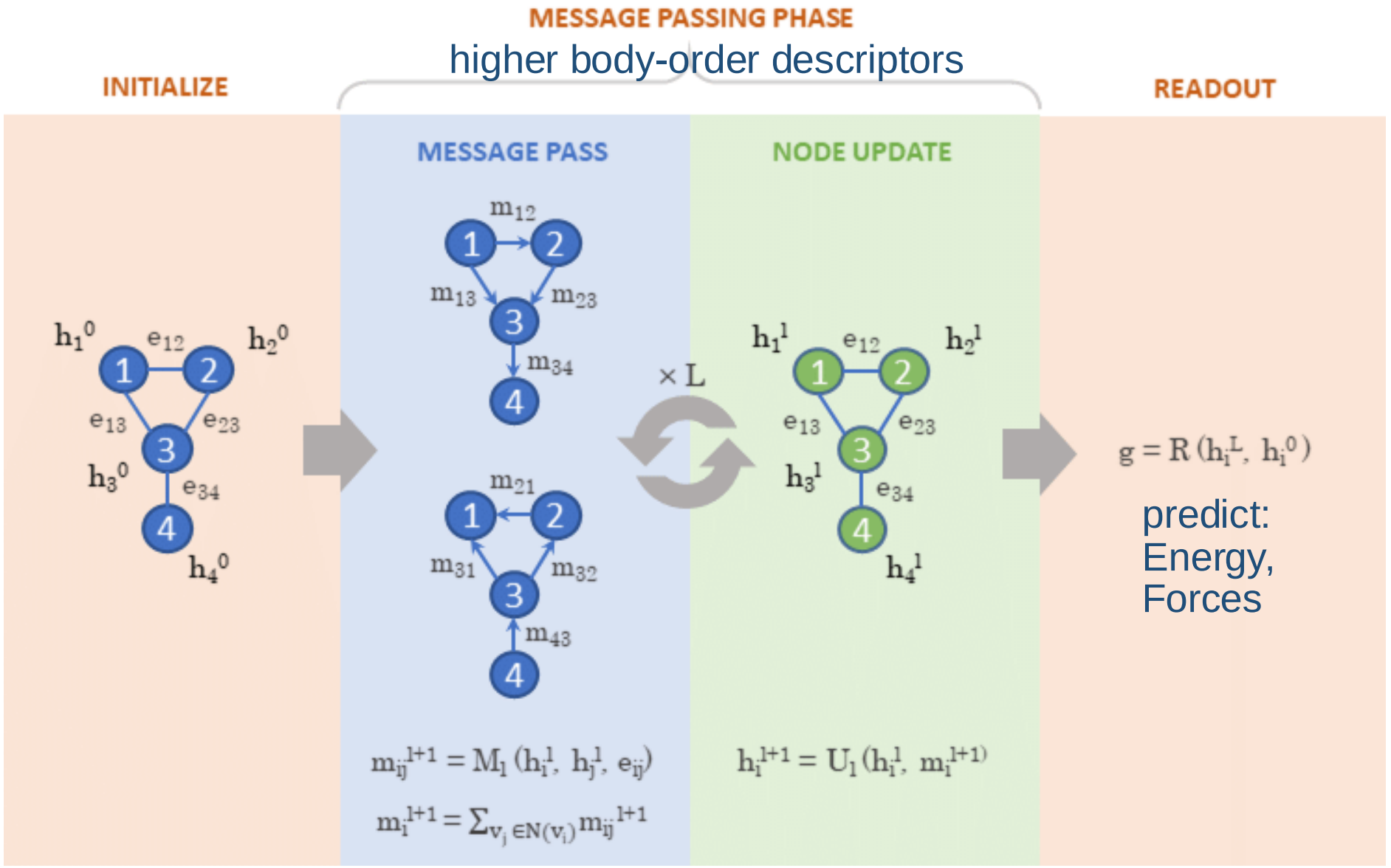


MPNN+ACE: MACE

Molecules are graphs



- each node is an atom
- messages are interactions between neighbor atoms
- **MACE** uses **ACE** basis as initial features



‘Embedding’
How the model sees the data

‘Interaction’
Pooling information from neighbours

‘Product’
Form powers of features, for higher body order

Atom’s chemical species

graph ‘edge’

Species one-hot

Node embedding

$$h^{(0)}_{i,k00} = \sum_z W_{kz}\delta_{zzi}$$

radial embedding

$$j^n_0(r_{ij}) = \sqrt{\frac{2}{r_{\text{cut}}}}\frac{\sin\left(\pi n \frac{r_{ij}}{r_{\text{cut}}}\right)}{r_{ij}} f_{\text{cut}}(r_{ij})$$

Ylm’s (angular embedding)

$$Y^{m_1}_{l_1}(\hat{r}_{ij})$$

Node features 0 (n_node, k)

edge features (n_edge, 8)

edge attributes (n_edge, (ell+1)^2)

linear

$$\bar{h}^{(s)}_{i,kl_2m_2} = \sum_{\tilde{k}} W^{(s)}_{k\tilde{k}l_2} h^{(s)}_{i,\tilde{k}l_2m_2}$$

radial MLP

$$R^{(s)}_{k\eta_1 l_1 l_2 l_3}(r_{ij}) = \text{MLP}\left(\{j^n_0(r_{ij})\}_n\right)$$

One-particle basis ‘conv_tp’

$$\phi^{(s)}_{ij,k\eta_1 l_3 m_3} = \sum_{l_1 l_2 m_1 m_2} c^{\eta_1 m_1 l_1 m_2}_{l_3 m_3} R^{(s)}_{k\eta_1 l_1 l_2 l_3}(r_{ij}) \times Y^{m_1}_{l_1}(\hat{r}_{ij}) \tilde{h}^{(s)}_{j,k_2 l_2 m_2}$$

Neighbour sum + linear

$$A^{(s)}_{i,kl_3m_3} = \sum_{\tilde{k},\eta_1} W^{(s)}_{k\tilde{k}\eta_1 l_3} \sum_{j\in\mathcal{N}(i)} \phi^{(s)}_{ij,\tilde{k}\eta_1 l_3 m_3}.$$

Product

$$B^{(t)}_{i,\eta_\nu kLM} = \sum_{lm} \mathcal{C}^{LM}_{\eta_\nu,lm} \prod_{\xi=1}^{\nu} A^{(t)}_{i,kl_\xi m_\xi}, \qquad m^{(t)}_{i,kLM} = \sum_\nu \sum_{\eta_\nu} W^{(t)}_{z_i kL,\eta_\nu} B^{(t)}_{i,\eta_\nu kLM}$$

update

$$h^{(t+1)}_{i,kLM} = \sum_{\tilde{k}} W^{(t)}_{kL,\tilde{k}} m^{(t)}_{i,\tilde{k}LM} + \sum_{\tilde{k}} W^{(t)}_{kz_iL,\tilde{k}} h^{(t)}_{i,\tilde{k}LM}$$

readout

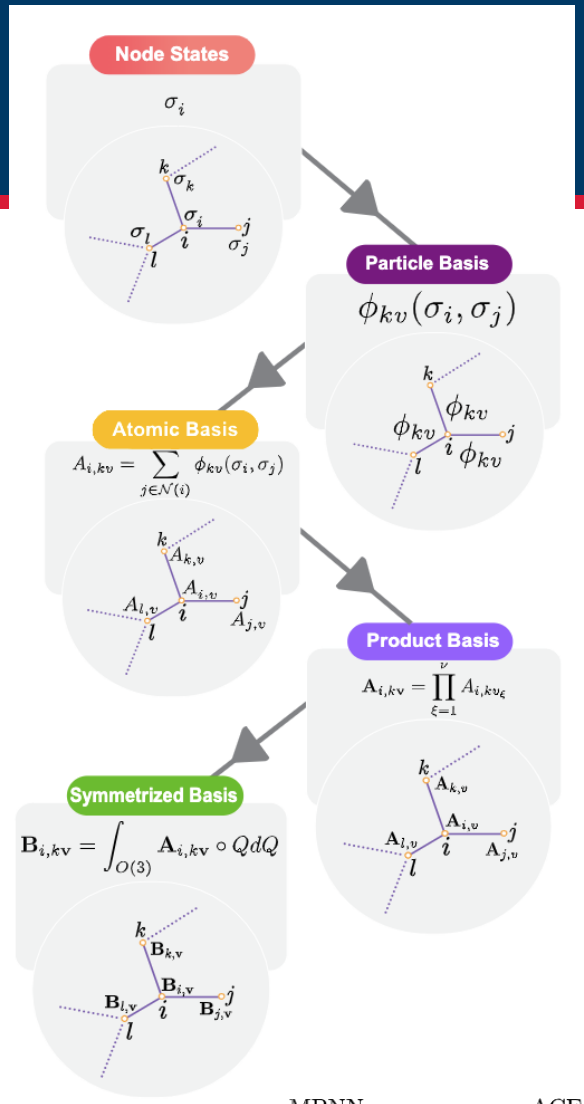
Node energies

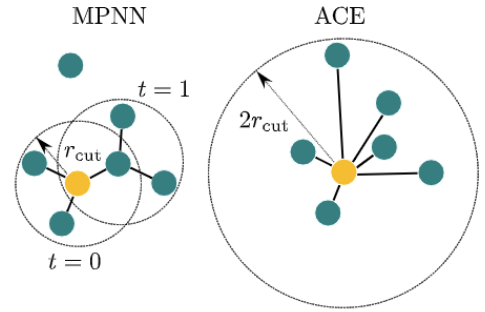
Node features 1 (n_node, k)

edge features (n_edge, 8)

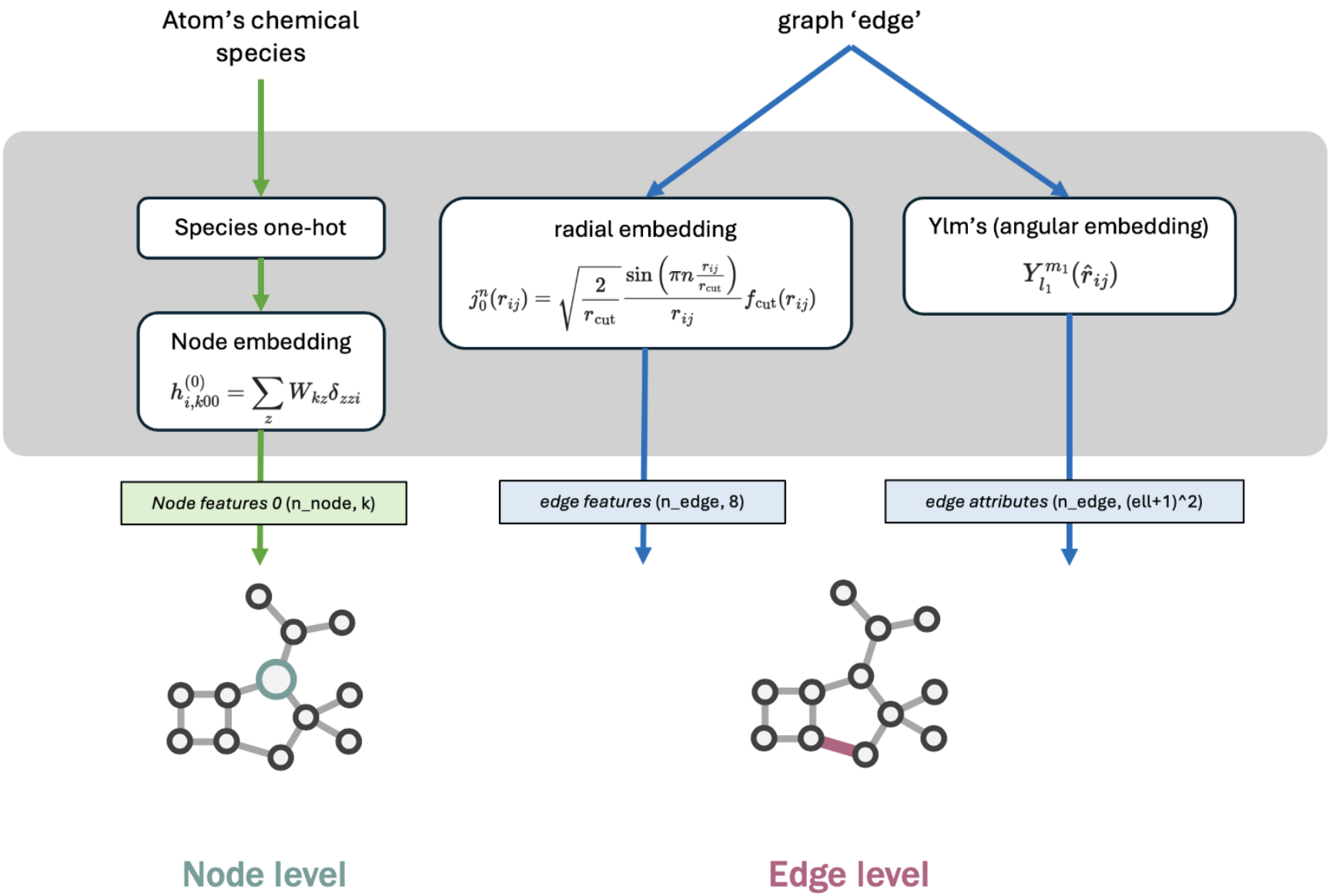
edge attributes (n_edge, (ell+1)^2)

repeat Interaction + product





1. Embeddings: initializing the graph



Computed once in the beginning.

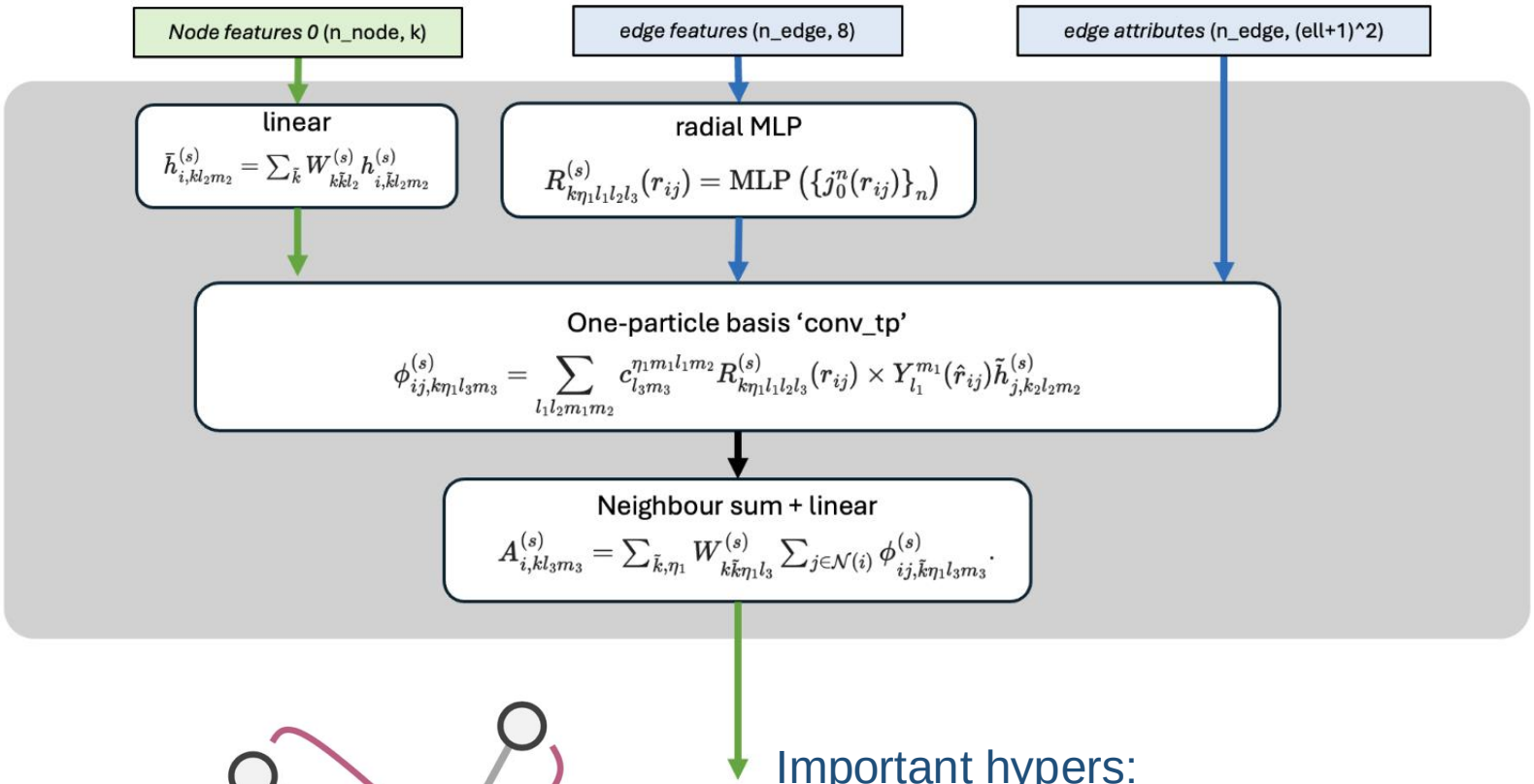
For each atom/node expand distances/ angles into radial/angular basis.

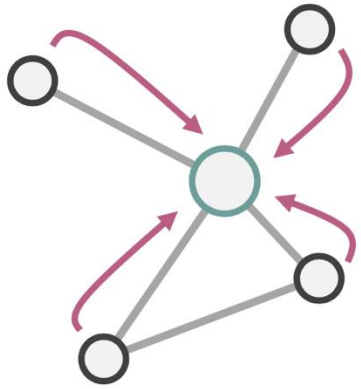
Encode species one-hot and project into a learnable embedding.

Important hypers:

- R_{cut} (r_max)
- L_{max} (max_ell)
- N_{max} (num_radial_basis)

2. Interaction: pooling across neighbours





Important hypers:

- number of interactions S
- k (number of channels)

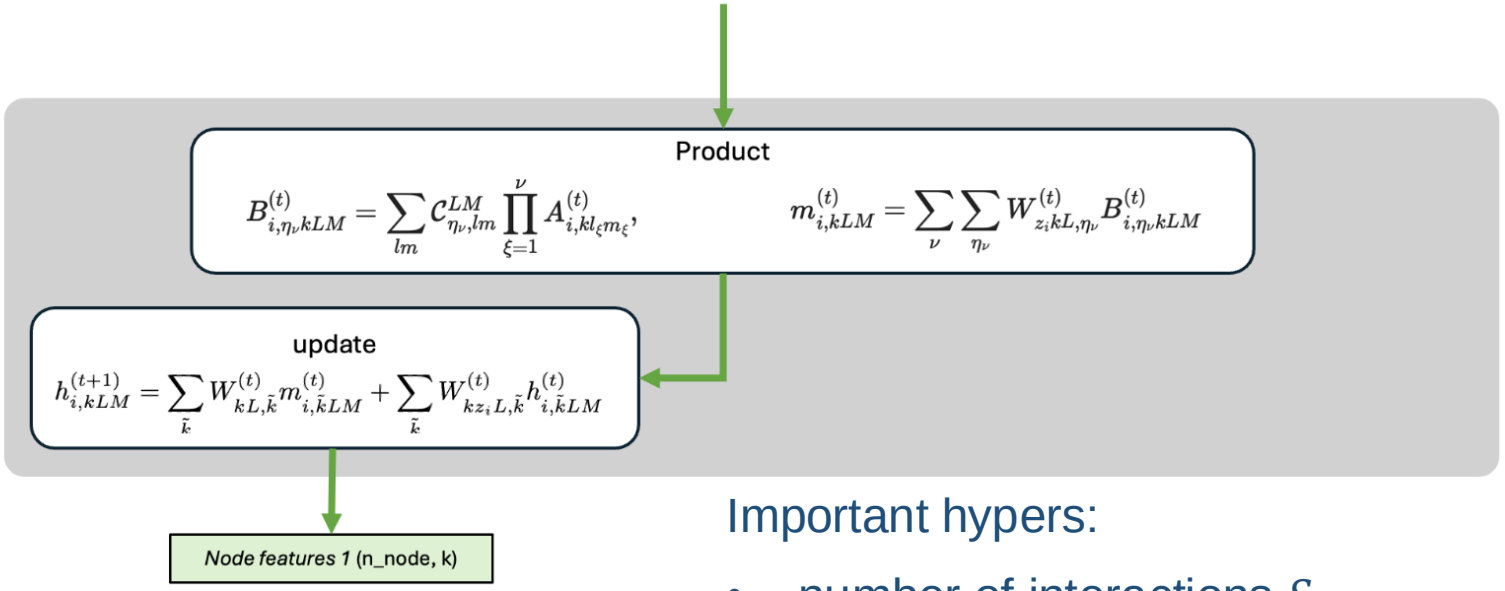
Updated every time the model evaluates the interaction layer (S times)

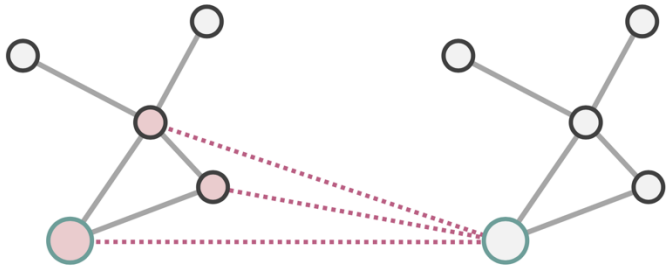
The one-particle basis (ϕ) is constructed using **e3nn**, and unlike in ACE, it is equivariant (there are η_1 ways to construct equivariances)

The atomic basis (A) is obtained by pooling over neighbors (permutational invariance)

Everything is multiplied by additional learnable weights

3. Product: updating the node features





Important hypers:

- number of interactions S
- k (number of channels)
- L_{max} (maximum equivariance):
  32x1e+32x1o:
  32 invariant and
  32 equivariant channels
  $k \times (Y_{00}, Y_{1,-1}, Y_{1,0}, Y_{1,+1})$
- ν correlation order

Updated every time the model evaluates the product layer (S times)

The atomic basis is tensor-multiplied with itself to form the product basis (main difference to NequIP: short-cut to obtaining higher body orders with fewer layers)

The fully symmetric basis (B) is constructed by contracting the atomic basis (A) using **e3nn**. There are η_ν ways to achieve this.

Finally, the message is obtained by multiplying with learnable weights and the node feature is updated.

4. Readout from each layer

In an S-layer MACE, the readout from the features at layer s is:

$$\mathcal{R}^{(s)}\left(\boldsymbol{h}_i^{(s)}\right) = \begin{cases} \sum_k W_k^{(s)} h_{i,k00}^{(s)} & \text{if } 1 < s < S \\ \mathrm{MLP}\left(\left\{h_{i,k00}^{(s)}\right\}_k\right) & \text{if } s = S \end{cases}$$

Forces obtained via autodiff. Loss in each batch:

$$\mathcal{L} = \frac{\lambda_E}{B}\sum_{b=1}^{B}\left(\frac{E_b - \hat{E}_b}{N_b}\right)^2 + \frac{\lambda_F}{3B}\sum_{b=1}^{B}\sum_{i_b,\alpha=1}^{N_b,3}\left(-\frac{\partial E_b}{\partial r_{i_b,\alpha}} - \hat{F}_{i_b,\alpha}\right)^2$$

Total field-of view:

$$R_{cut} \times S$$

Total body-order (within the first)

$$(\nu + 1) \times S + 1$$

Training the Model: Minimizing the Loss

Input: atomic environments p

Output: observables $F(p)$

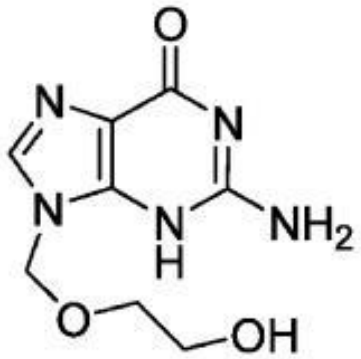
Learning:

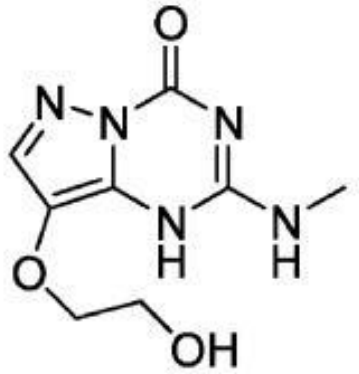
minimize loss on train set $\{p, F(p)\}$

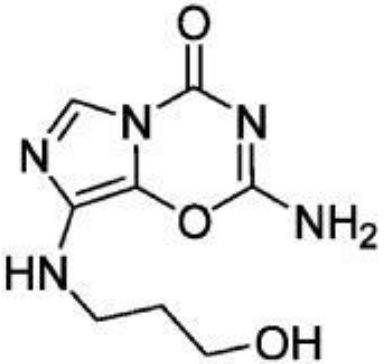
$$\mathcal{L} = \sum_{n=1}^{N_{data}} [F_n - F(p_n)]^2$$

to determine w_{ij} etc

Data can be different molecules, or different geometries or both ...



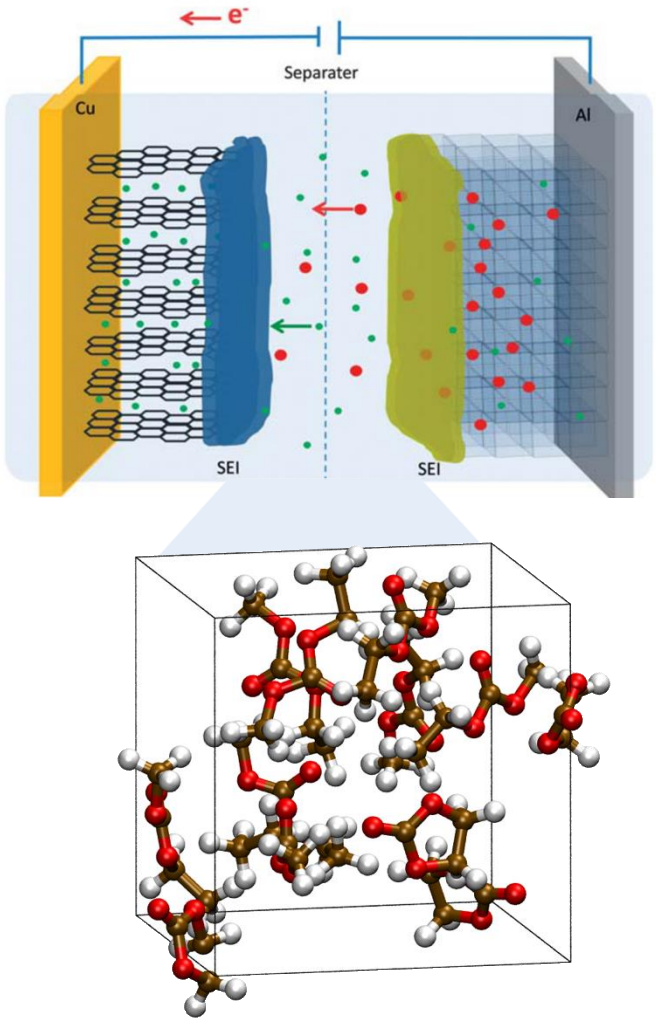




p_1	p_2	p_3
F_1	F_2	F_3
$F(p_1)$	$F(p_2)$	$F(p_3)$
F_1-$F(p_1)$	F_2-$F(p_2)$	F_3-$F(p_3)$

Case study: MLIP for an organic solvent

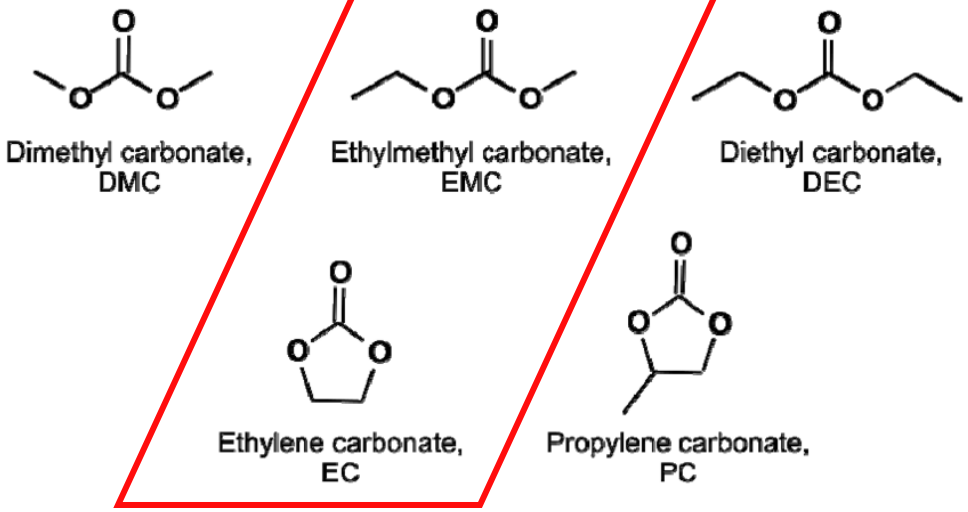
Li-ion battery

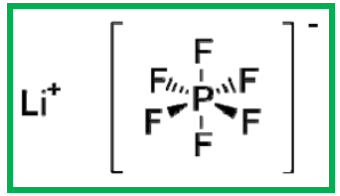


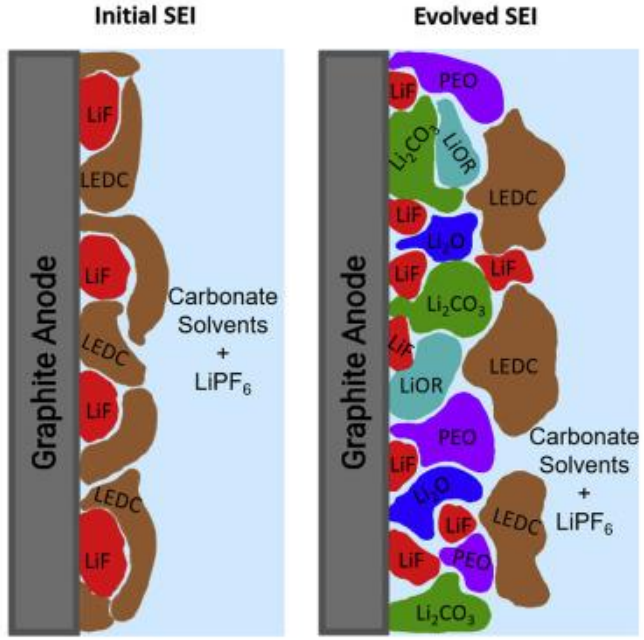
LP57 solvent: 33% EC : 67% EMC

- simple GAP model on total energies

- 3 atomic species: H, C, O
- neutral molecules $(\sim 1/r^3)$







Electrolyte-Interface

+Salt:

- 6 atomic species: add Li, P, F (quadratic scaling)
- charged long-range effects $(\sim 1/r)$

Case study: MLIP for an organic solvent

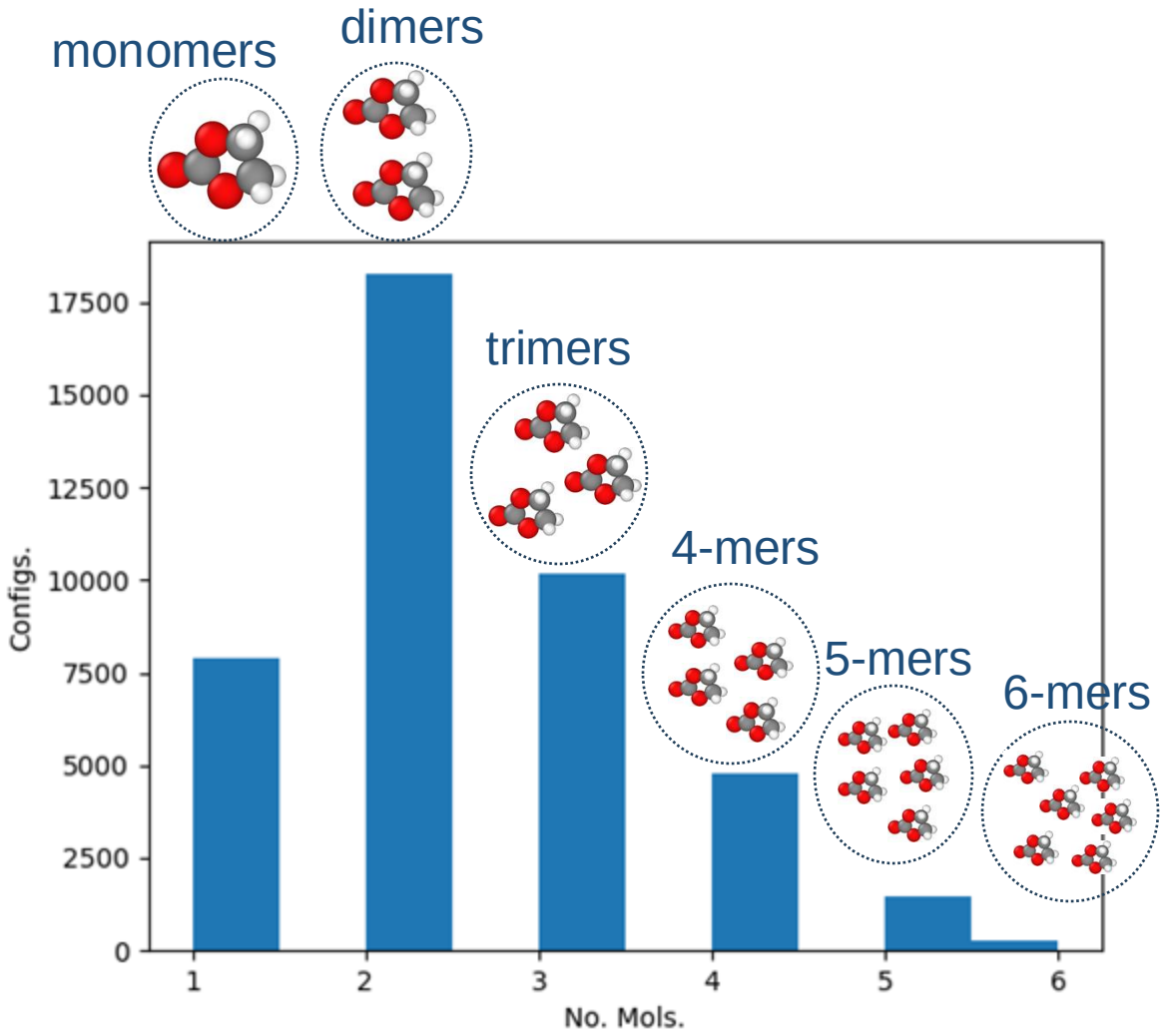
THE JOURNAL OF PHYSICAL CHEMISTRY B
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

Schrödinger

High-Dimensional Neural Network Potential for Liquid Electrolyte Simulations

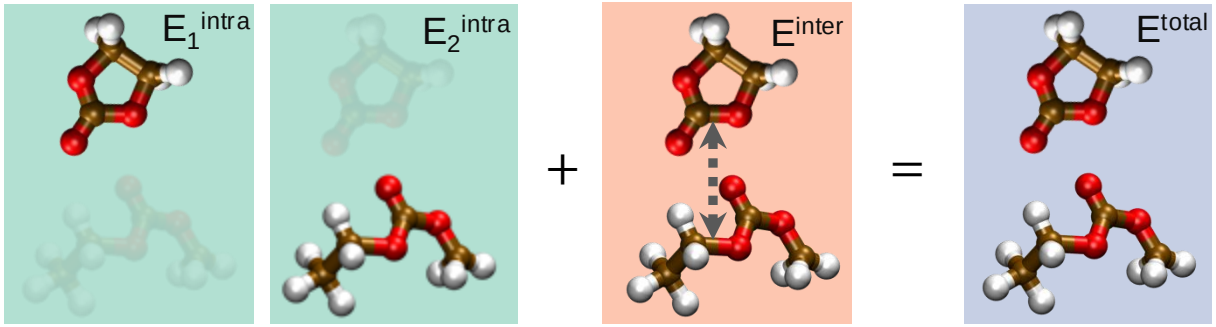
Steven Dajnowicz*, Garvit Agarwal*, James M. Stevenson, Leif D. Jacobson, Farhad Ramezanghorbani, Karl Leswing, Richard A. Friesner, Mathew D. Halls, and Robert Abel

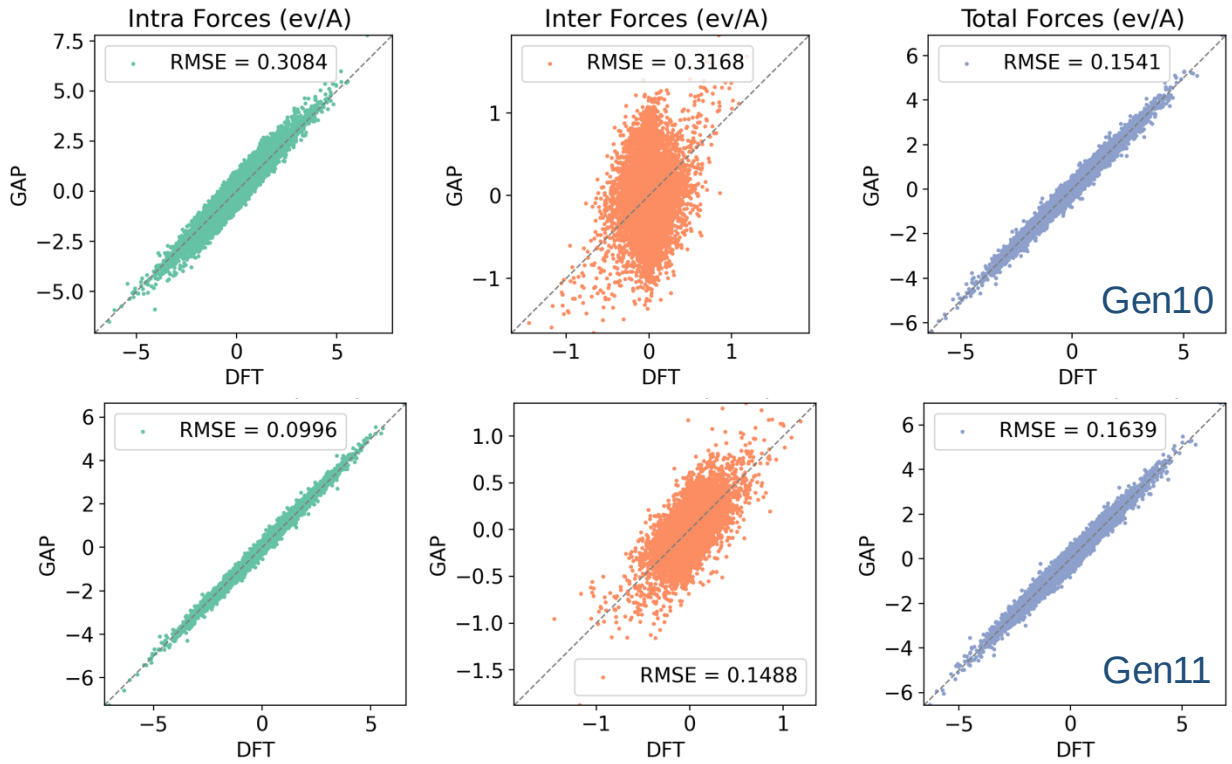
*



- training data type: molecular clusters
- Neural Network + long-range
- we will fit a MACE model on a subset of this data

Intra-/Inter- Decomposition as a Test





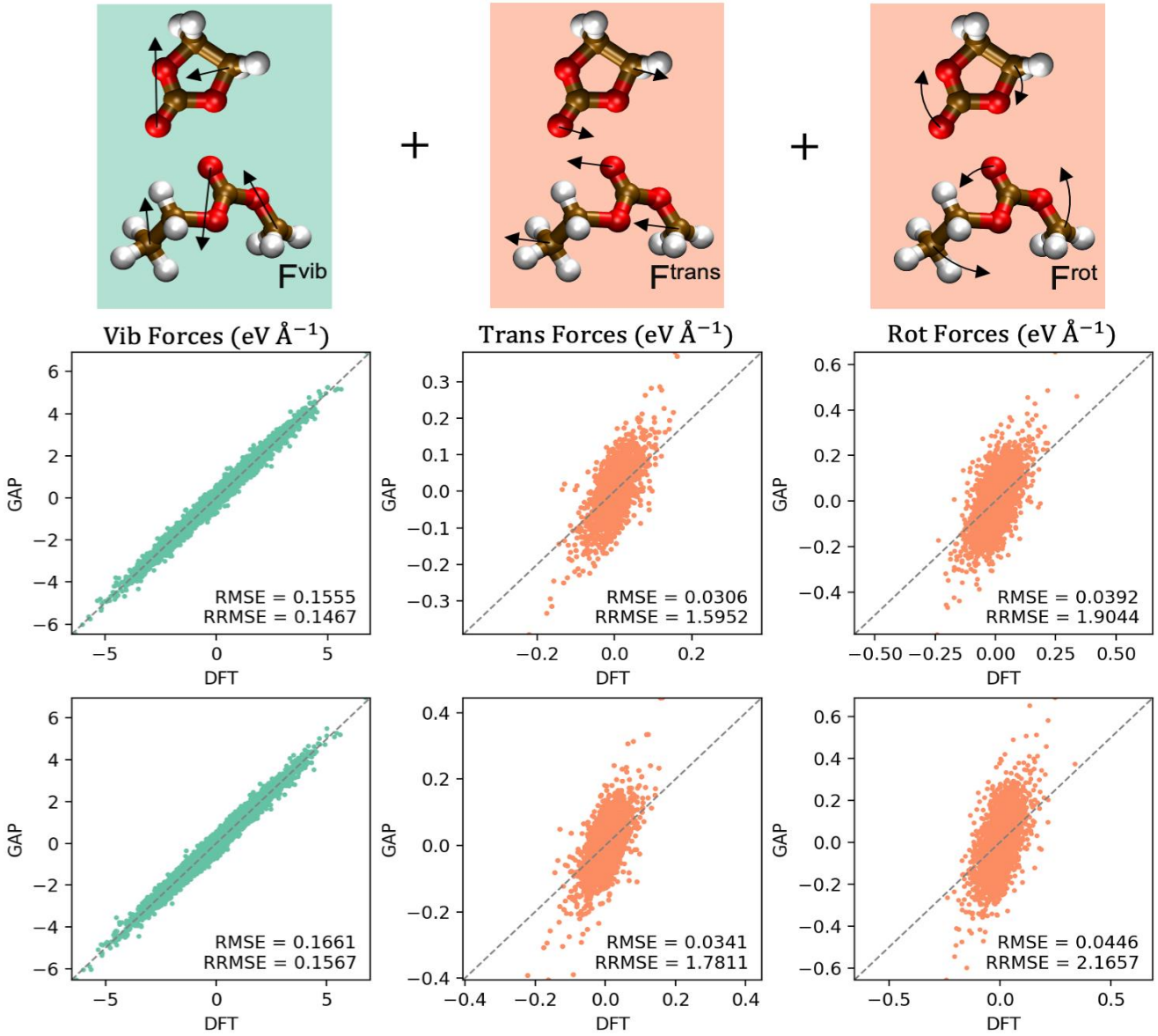
$$L = \left\| F_{total}^{DFT} - F_{total}^{ML} \right\|^2$$

$$L = \left\| \underbrace{\left(F_{intra}^{DFT} - F_{intra}^{ML}\right)}_{\text{dominates loss}} + \underbrace{\left(F_{inter}^{DFT} - F_{inter}^{ML}\right)}_{\text{drives dynamics}} \right\|^2$$

Intra/Inter test:

- **split liquid** configs into molecular
- recompute **molecules** (same geometry) **in vacuum**: intra contributions
- compute inter- as **difference between total and intra-**

Trans-/Rot-/Vib- Decomposition of Forces



Translational forces:

$$F_j^{\text{trans}} = \sum_{k \in j} f_k \qquad \text{(per molecule)}$$

$$f_i^{\text{trans}} = \frac{m_i}{M_j} F_j^{\text{trans}} \qquad \text{(per atom)}$$

Rotational forces:

$$T_j = \sum_{k \in j} f_k \times r_k \qquad \text{(per molecule)}$$

$$f_i^{\text{rot}} = m_i r_i \times (I_j^{\alpha\beta})^{-1} T_j \qquad \text{(per atom)}$$

Vibrational forces:

$$f_i^{\text{vib}} = f_i - f_i^{\text{trans}} - f_i^{\text{rot}} \qquad \text{(per atom)}$$

CAMMLS 2025
Chemicals and Materials Machine Learning School

Newcastle University

Machine Learning Interatomic Potentials II

Ioan-Bogdan Magdău
ioan.magdau@ncl.ac.uk

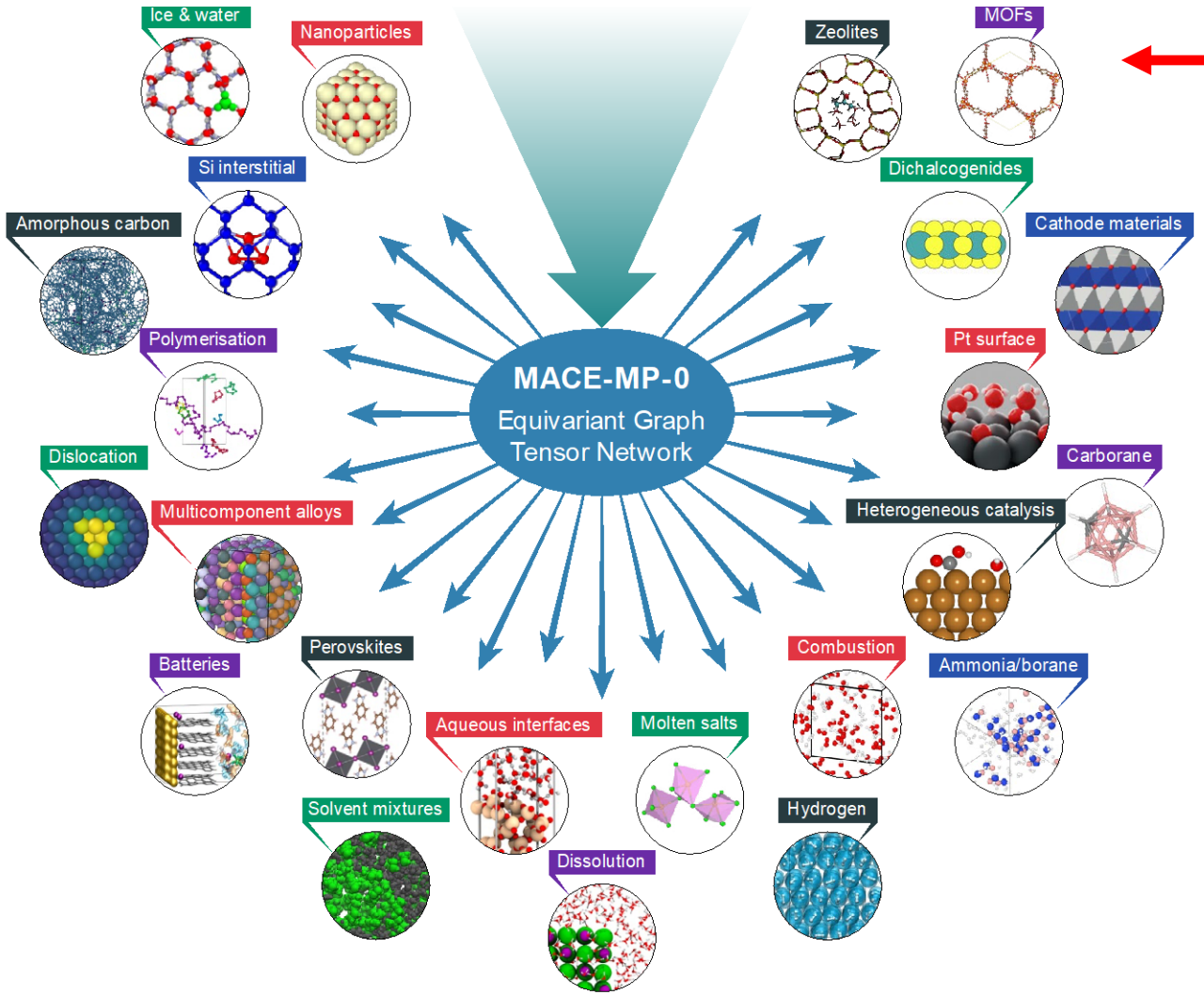
CAMMLS Spring School 2025

Lecture Outline

- Iterative Training: improving stability and accuracy
- Error estimation: committee models
- Active learning: unsupervised iterative training
- Foundational models: out-of-the-box MLIPs
- Fine-tuning on new data and new labels

MACE-MP-0: a First Generation Foundational Model

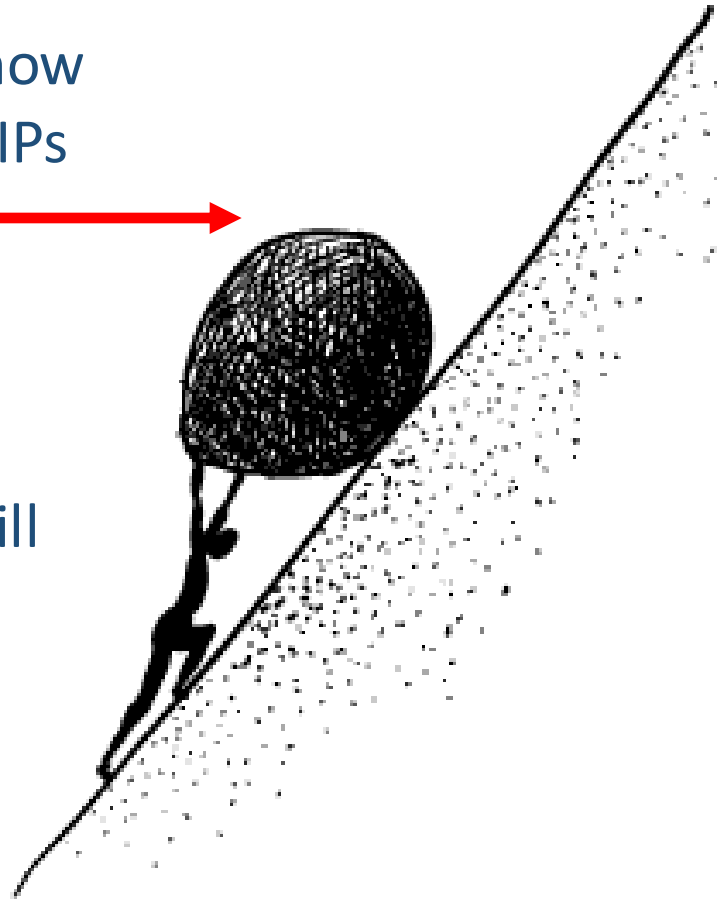
Batatia, I et al, arXiv preprint, arXiv:2401.00096 (2023)



to fully appreciate foundational models

let's first look at how we used to fit MLIPs a few years ago

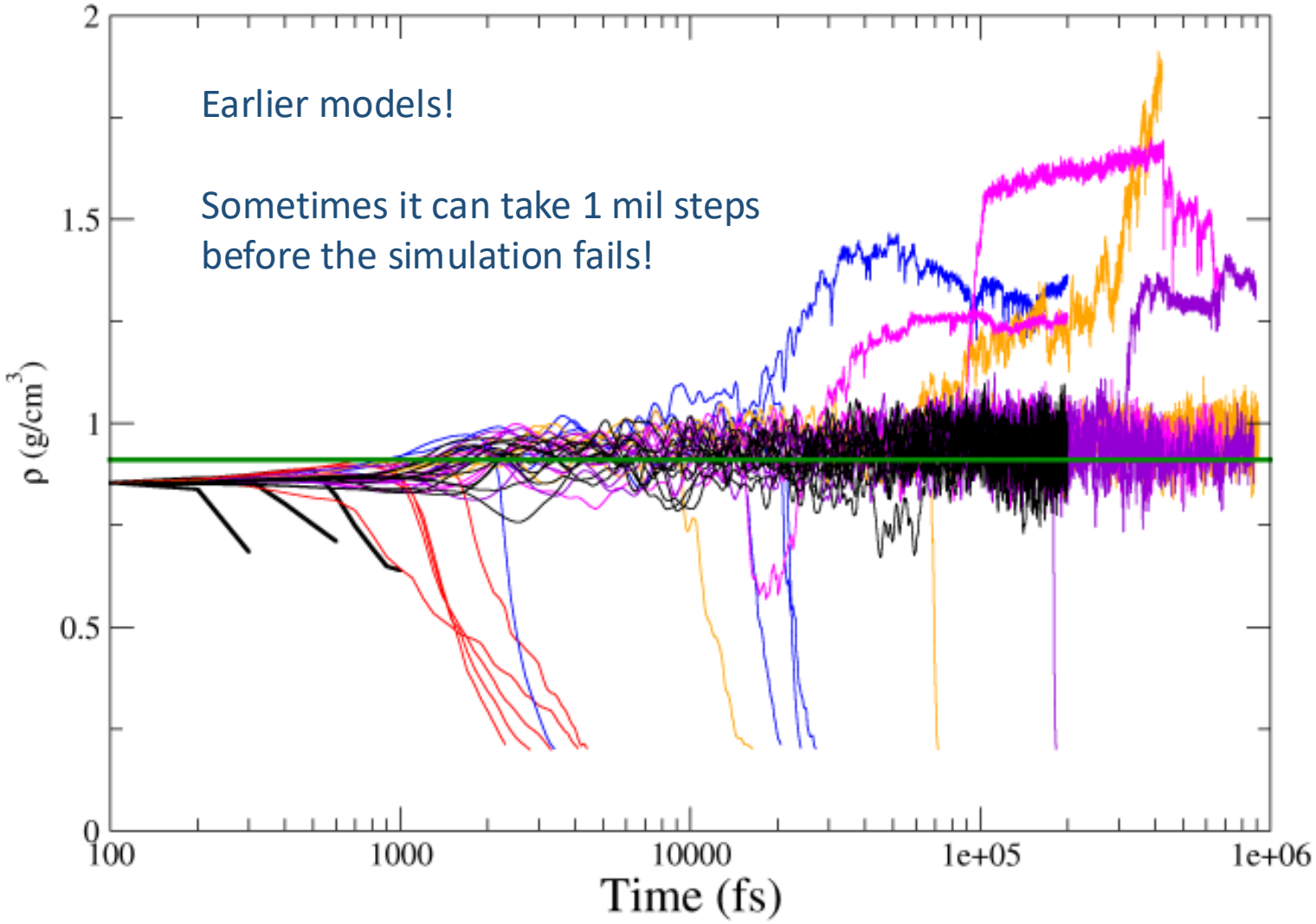
these concepts: **iterative training**, **active learning** will remain relevant, hopefully needed less often



Stability and Accuracy in Molecular Dynamics

Stability isn't everything, but without it we can't do anything

Most of us are interested in applications that involve MD beyond AIMD.



Test Error, MD Stability and MD Accuracy

1. Root Mean Square Errors (RMSE)

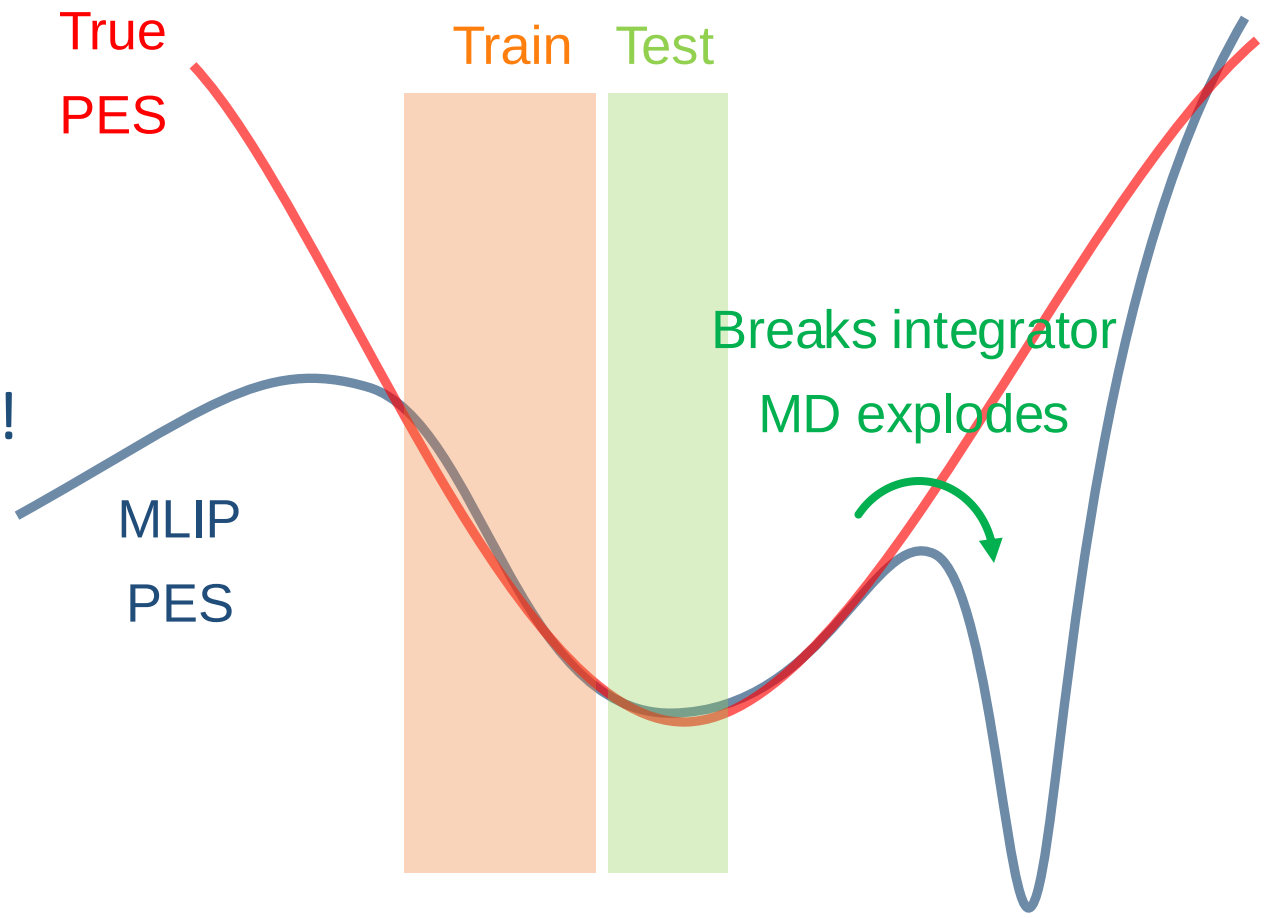
low test errors do no guarantee stability!

2. MD Stable Potentials

stability does not guarantee MD accuracy!

3. Prediction accuracy: thermodynamics and kinetics

Goal: correct probability distribution!
difficult to check



Test Error, MD Stability and MD Accuracy

1. Root Mean Square Errors (RMSE)

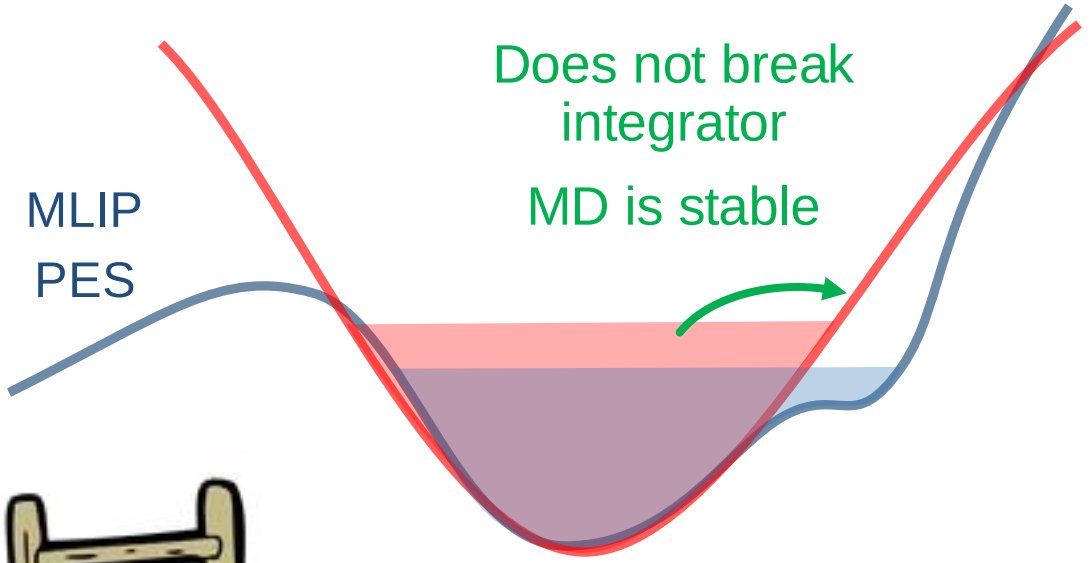
low test errors do no guarantee stability!

2. MD Stable Potentials

stability does not guarantee MD accuracy!

3. Prediction accuracy: thermodynamics and kinetics

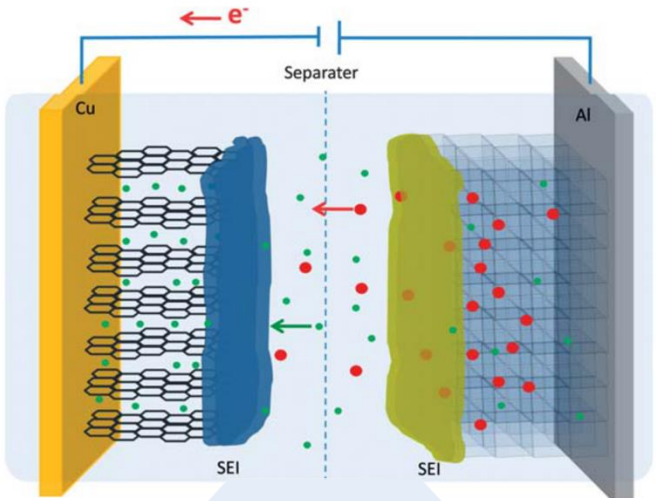
Goal: correct probability distribution!
difficult to check

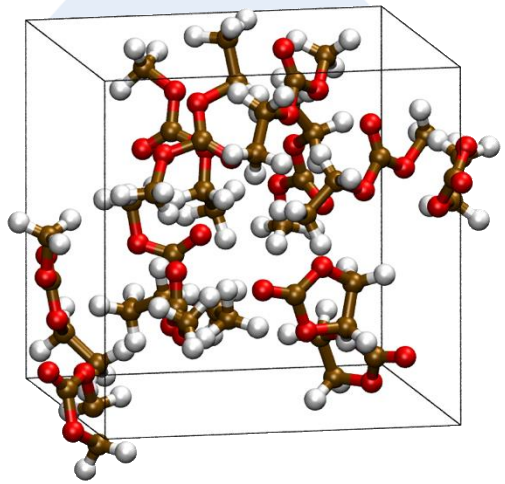


- Accurate Kinetics $\langle A(t) \cdot A(0) \rangle$
- Accurate Thermodynamics $\langle A \rangle$
- Stable MD $\tau_{stab} \rightarrow \infty$
- Low Test-Errors: Predetermined Tests

GAP Potential for the Organic ~~Electrolyte~~ Solvent

Li-ion battery

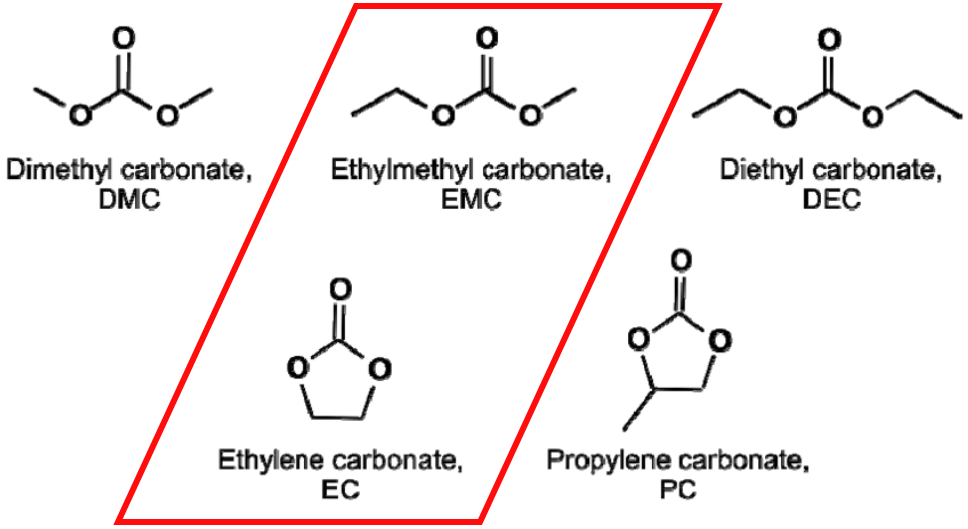


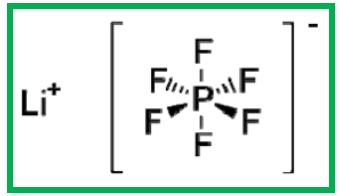


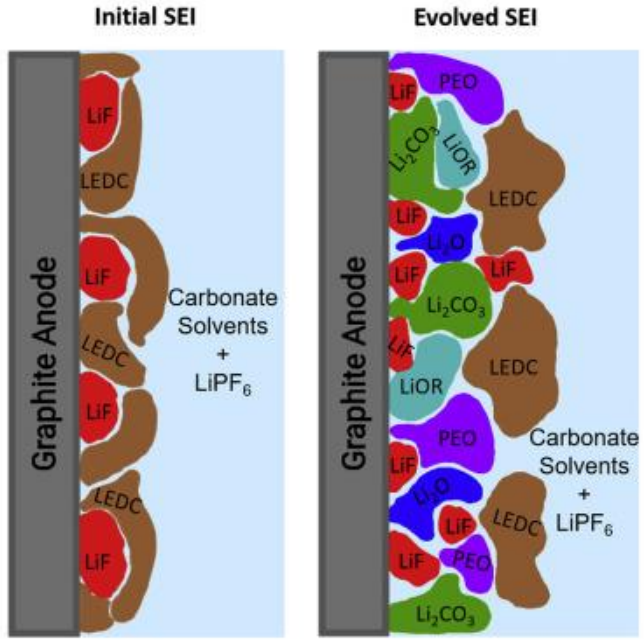
LP57 solvent: 33% EC : 67% EMC

- simple GAP model on total energies

- 3 atomic species: H, C, O
- neutral molecules $(\sim 1/r^3)$





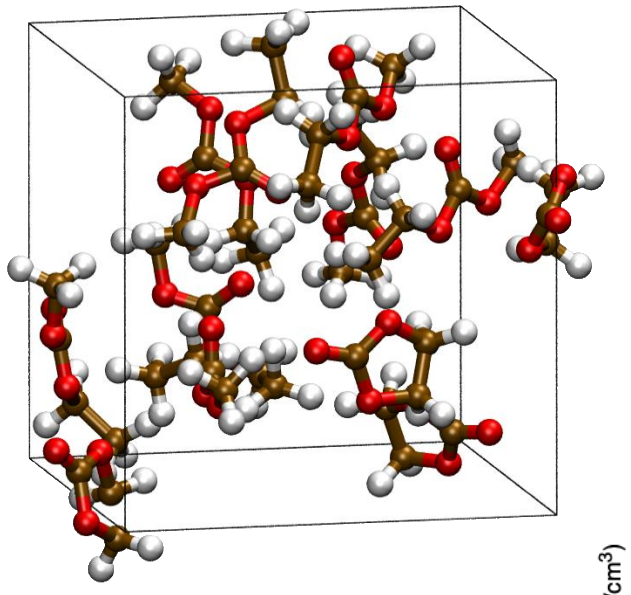


Electrolyte-Interface

+Salt:

- 6 atomic species: add Li, P, F (quadratic scaling)
- charged long-range effects $(\sim 1/r)$

Training on Fixed / Pre-generated Data



OPLS sampling: decorrelated samples, low computational cost

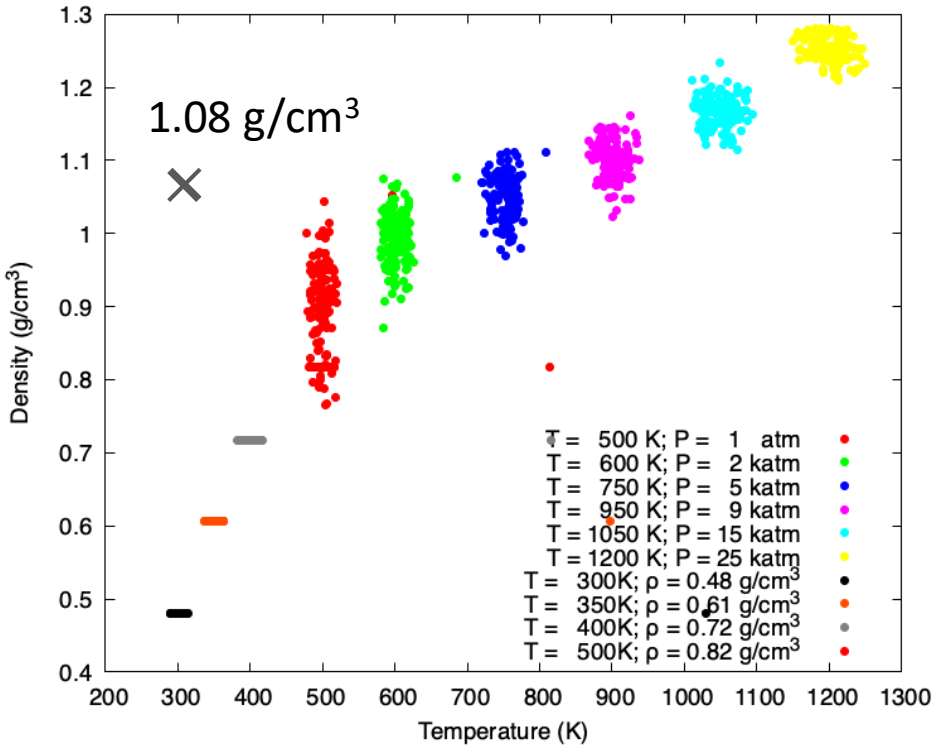
DFT calculations **PBE+G06** 12-molecule configurations

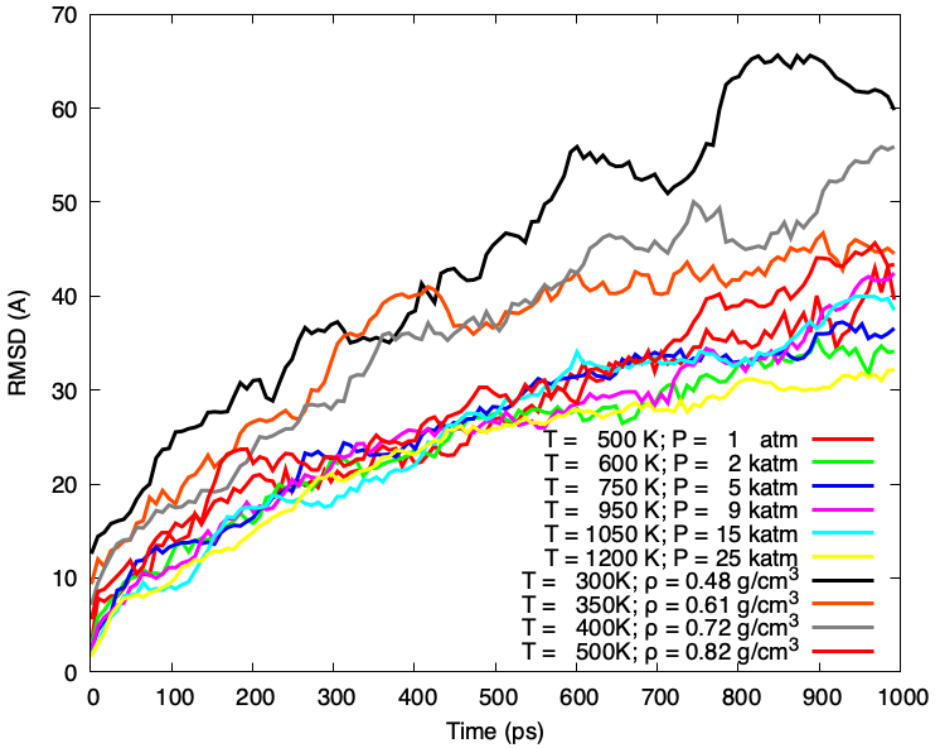
Wide range of **densities and temperatures**, maintaining **diffusive** behavior

Target Conditions:

33% EC: 66% EMC

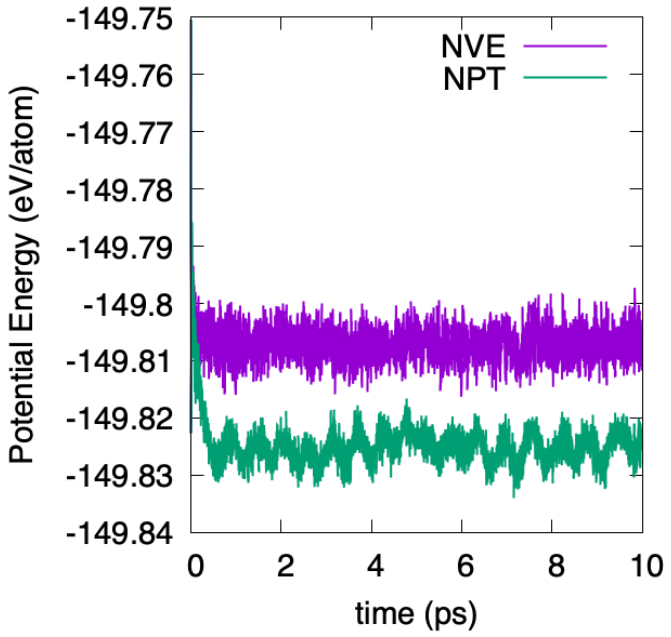
~1.08 g/cm^3 at 300K

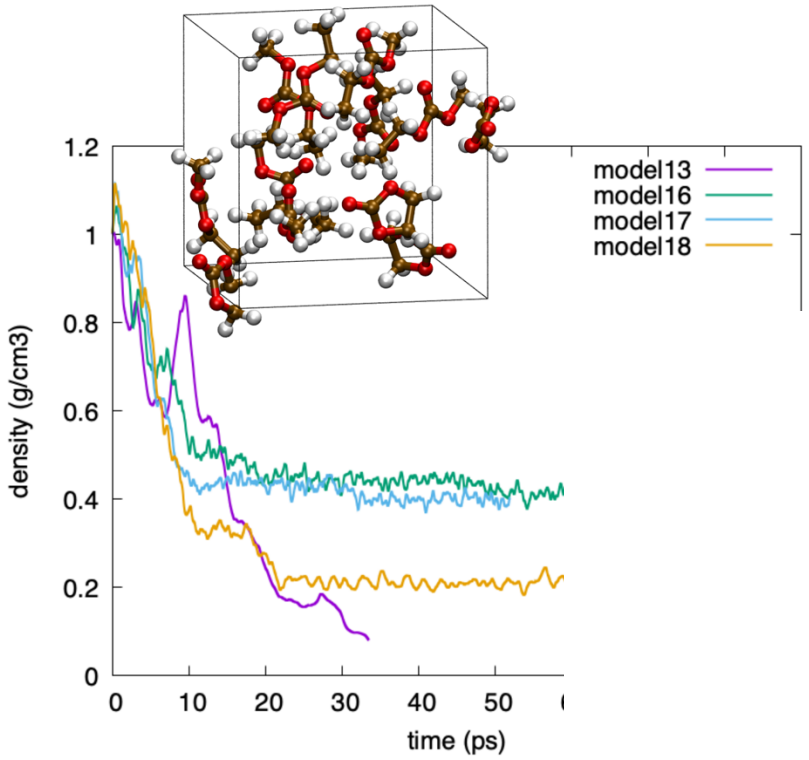




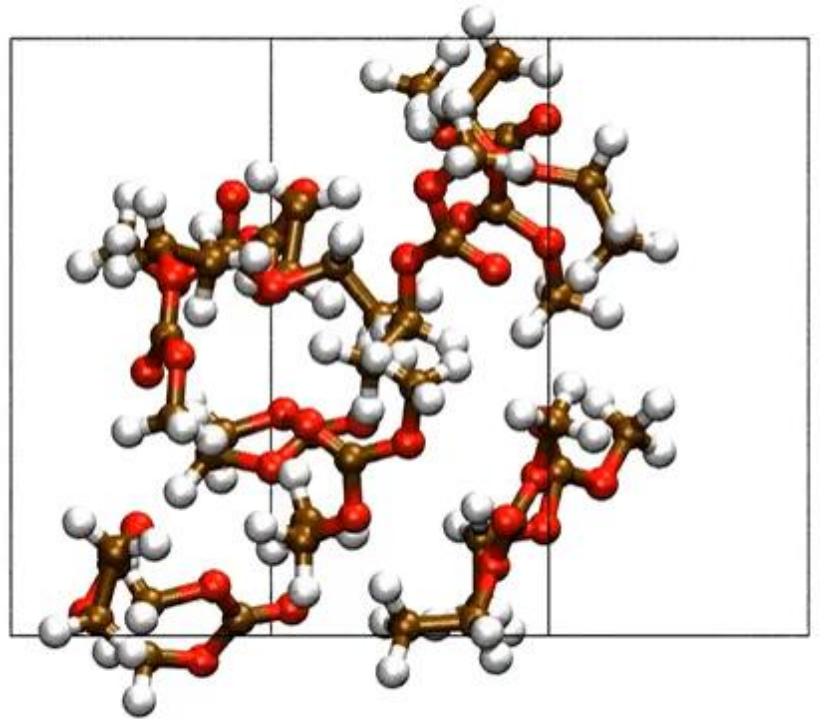
The Density Problem: Bubble Formation

stable NVE/NVT



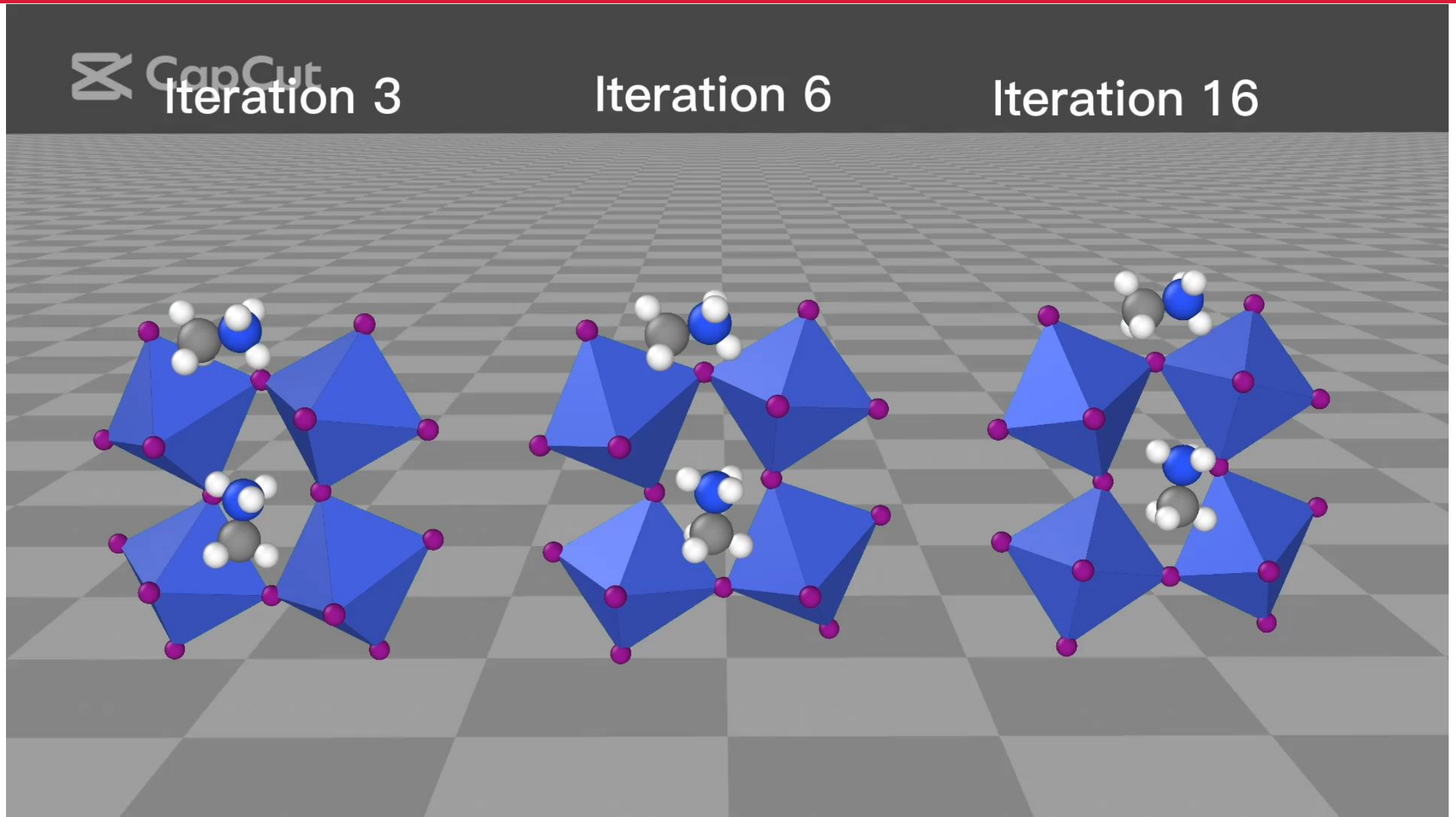


- well-behaved molecules
- liquid density collapse
- bubbles formation

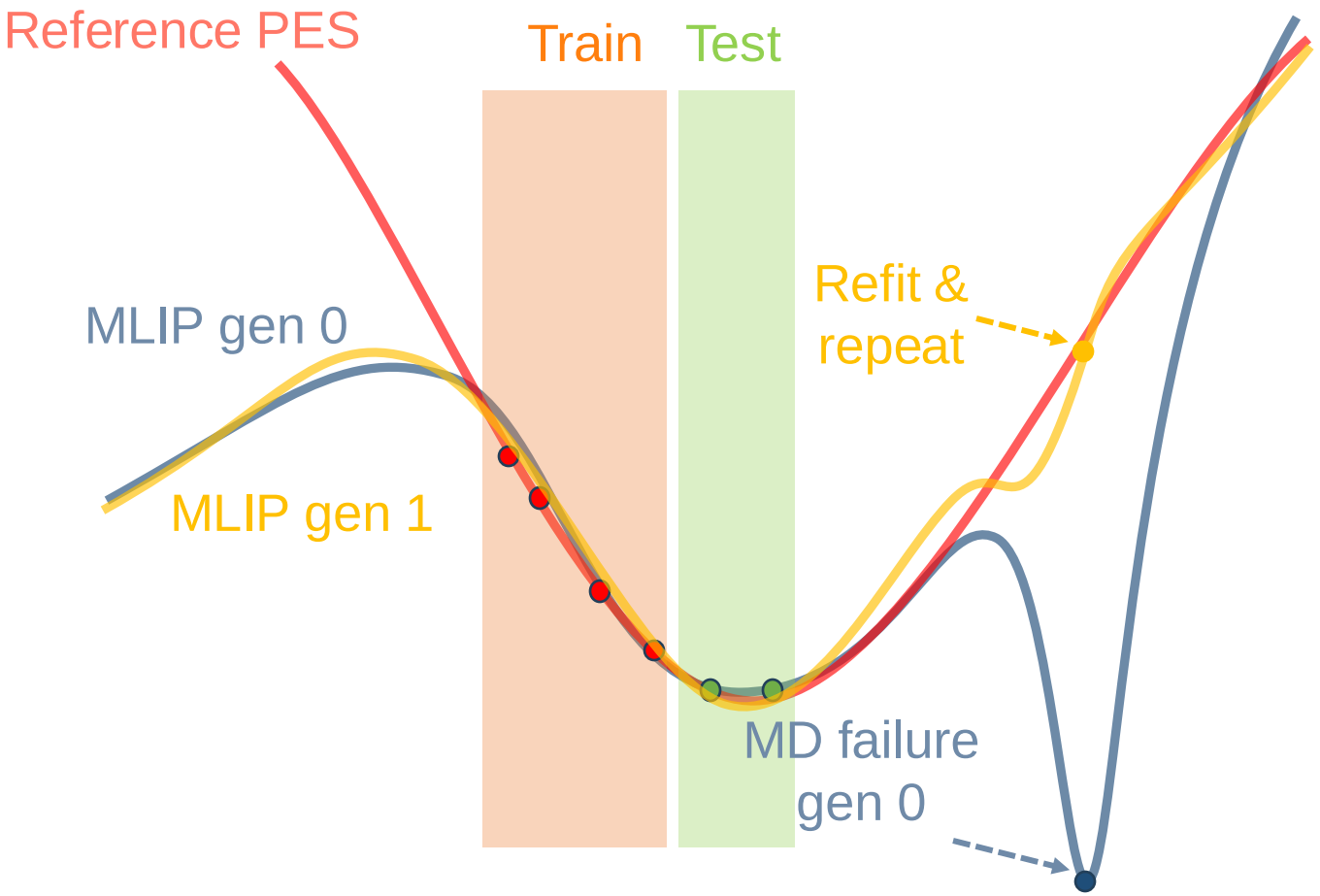


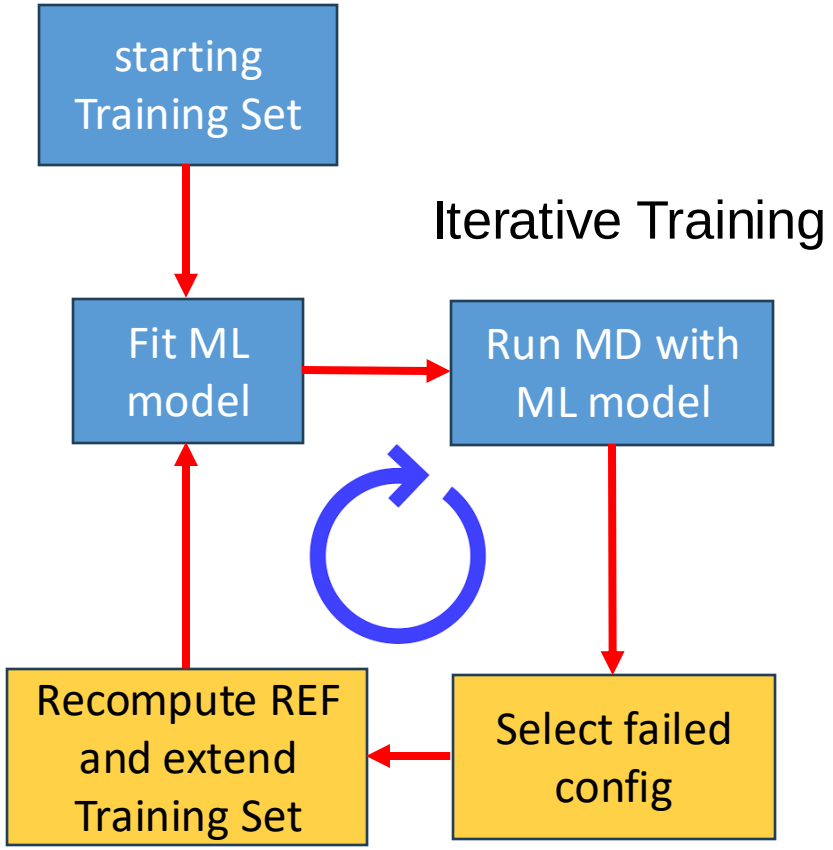
- Coverage of phase space?
- Transferability across **molecular compositions**?
- Difference in **scale** and **dimensionality** of **Intra- / Inter-**?

Iterative Training: model learns from mistakes



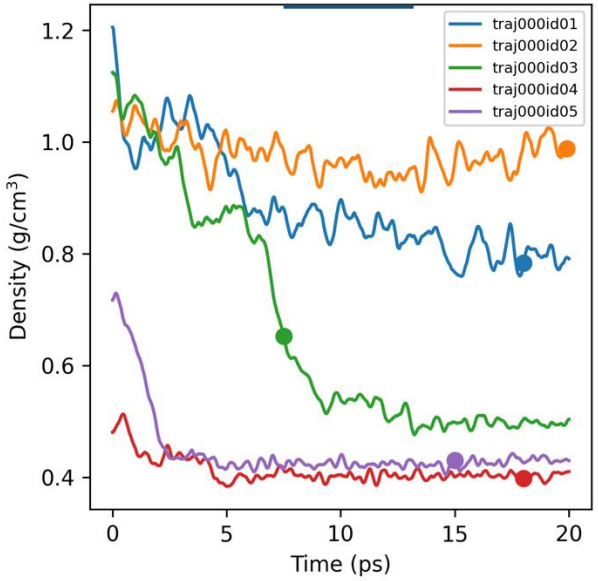
Improving the models – Iterative Training



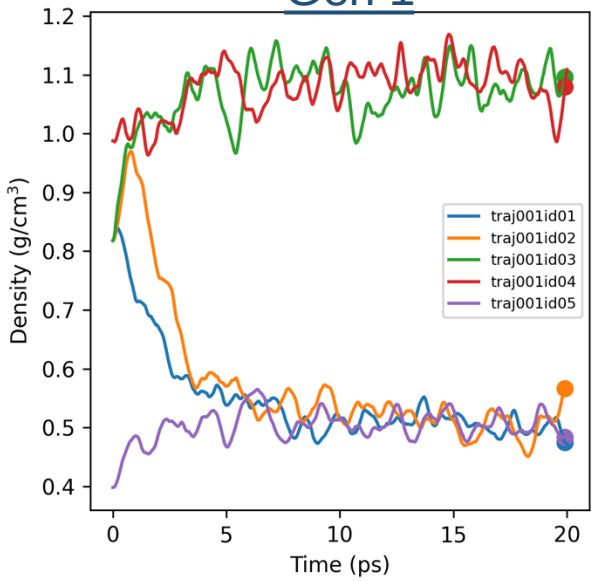


Iterative Training

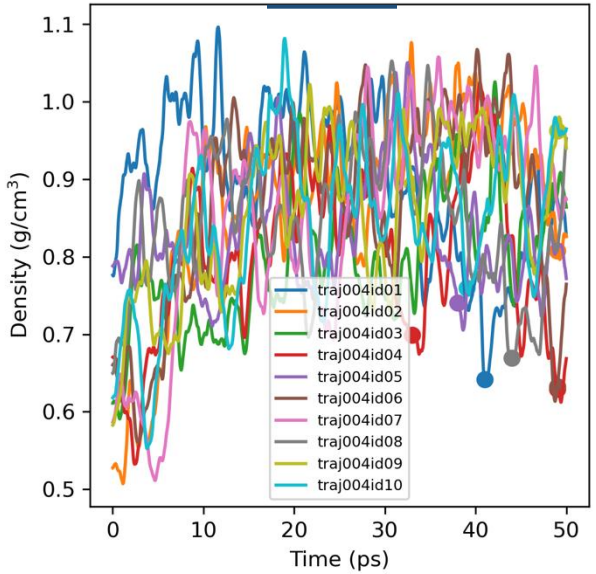
Gen 0



Gen 1



Gen 4



50 OPLS configs

Iterations with **12-molecules** (accessible to ab initio)

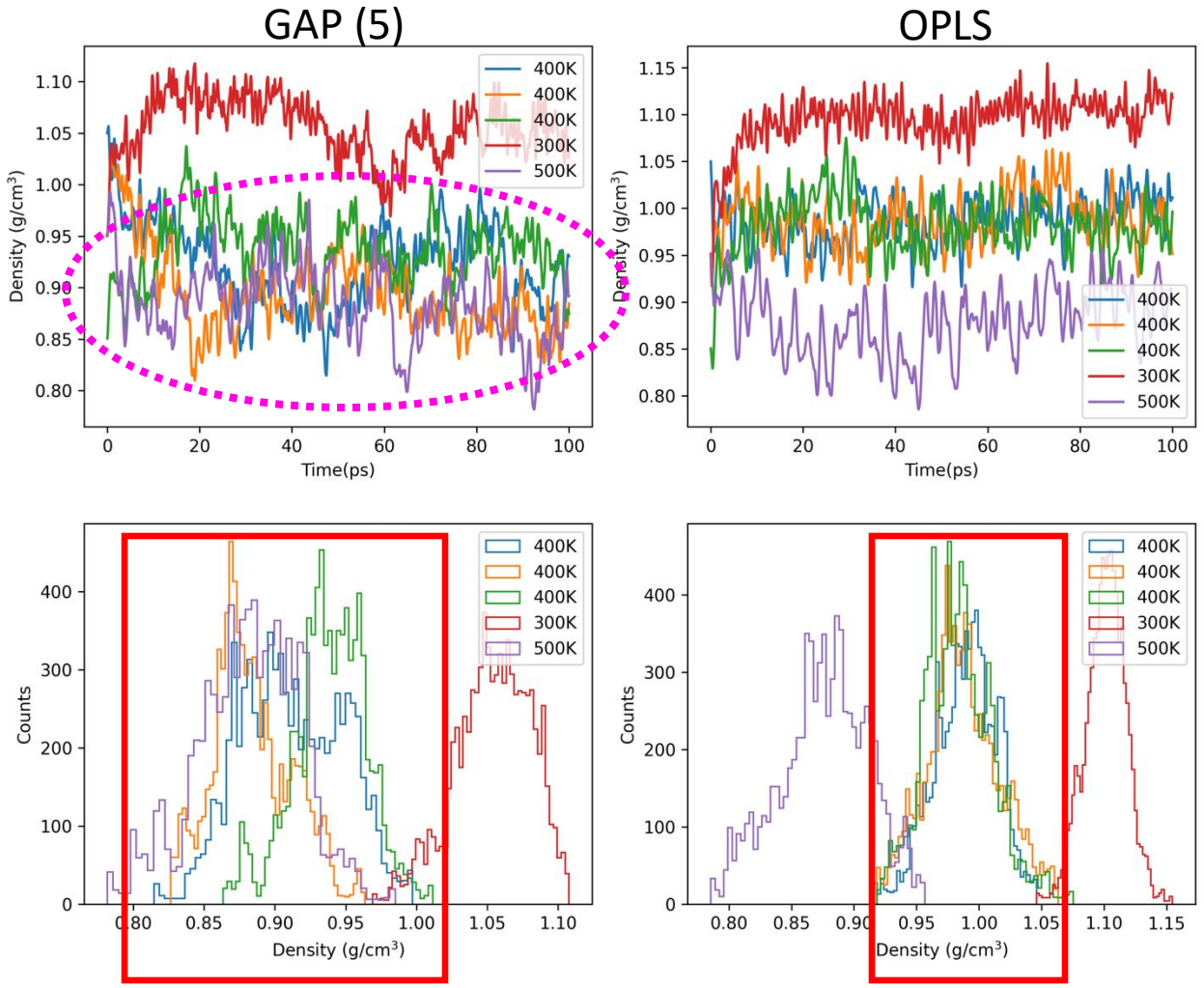
Multiple Temperatures: 500K ➔ 1200K (extrapolation)

Multiple Pressure: 1bar ➔ 25kbar

Proxy for folding back: **density instabilities**

Gen 4: 50 OPLS + 35 GAP-MD configs: **stable densities!**

Testing on Larger System (48 molecules)



Target Composition (33:67)

- **GAP densities stable at all temperatures!**
- Densities not well reproduced at 400K
- Same density at 400K and 500K

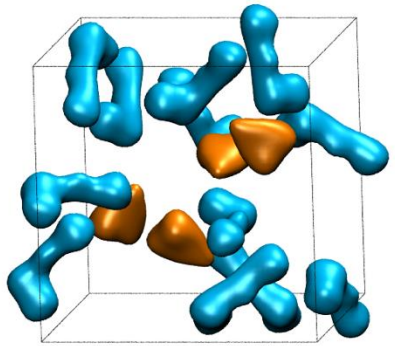
Transferability to other Molecular Compositions

Target Composition

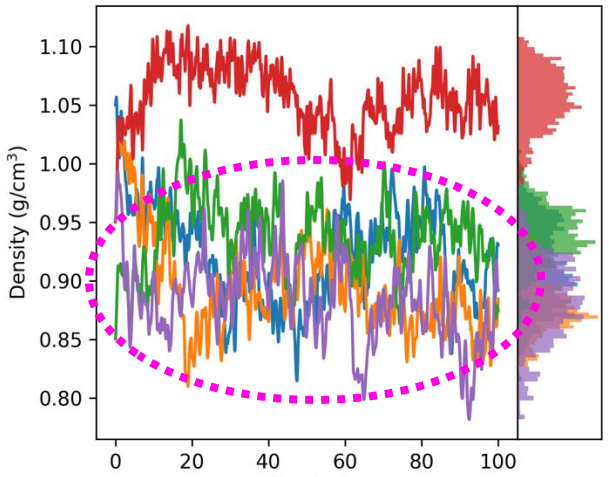
EC

EMC

EC:EMC 33:66



Gen 5

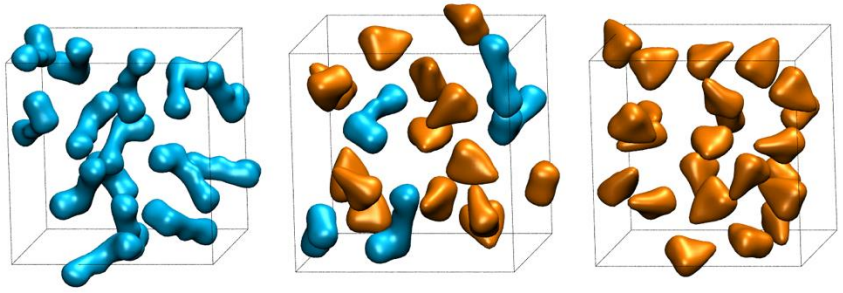


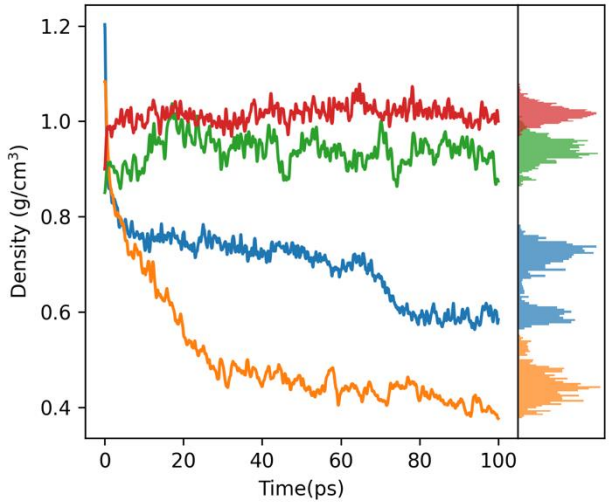
Other Compositions

0:100

66:33

100:0





poor transferability between compositions!

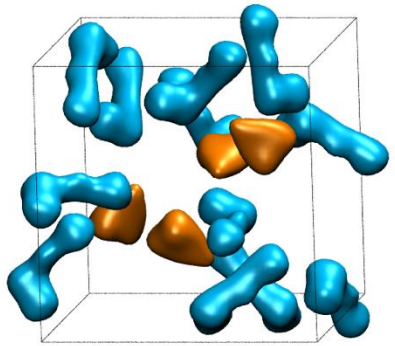
Transferability to other Molecular Compositions

Target Composition

EC

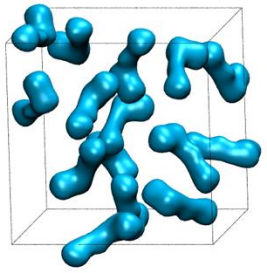
EMC

EC:EMC 33:66

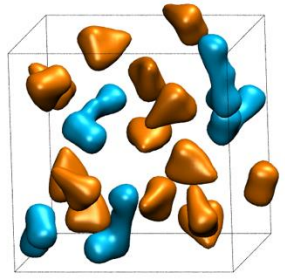


Other Compositions

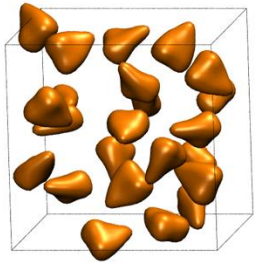
0:100

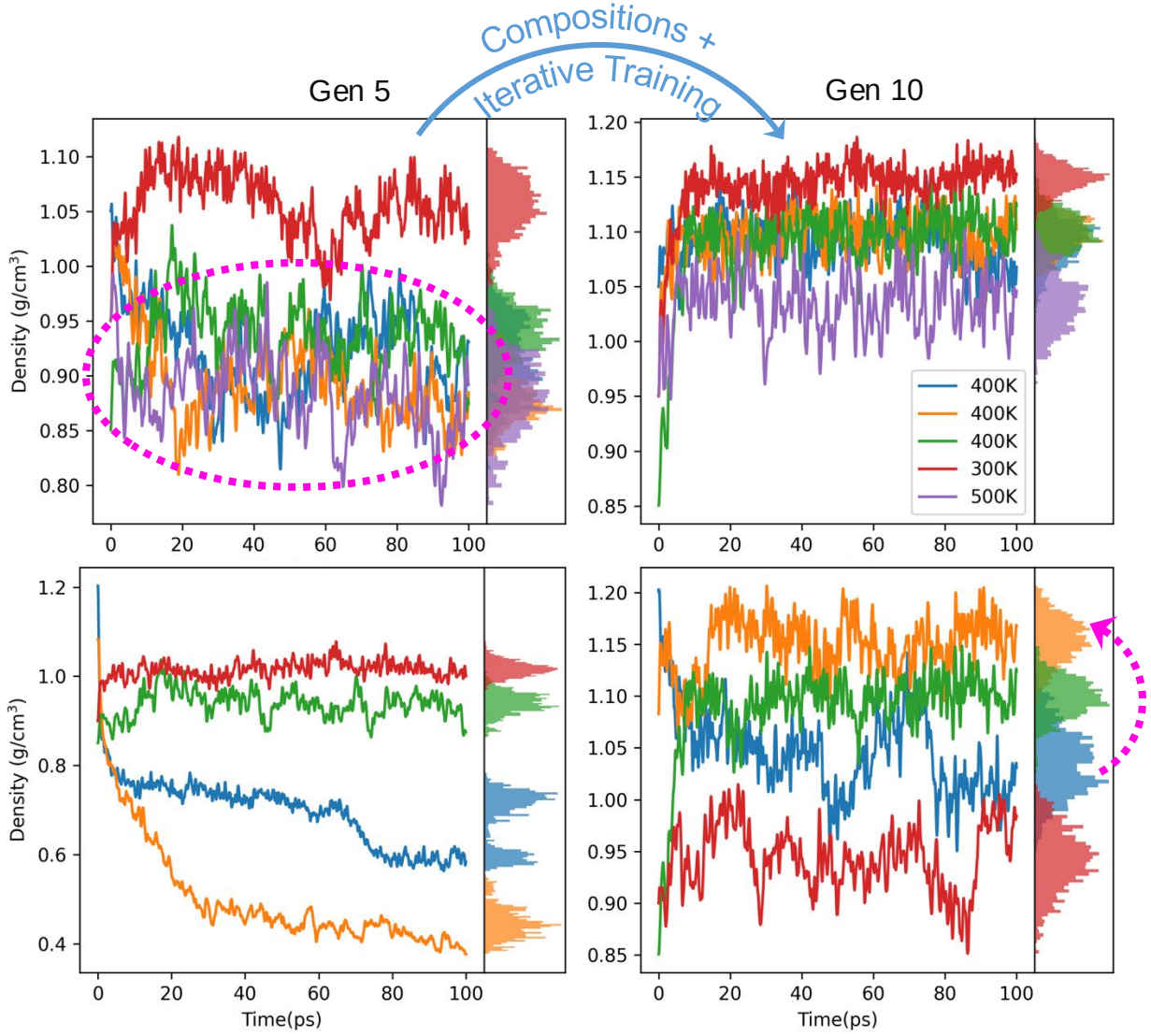


66:33



100:0



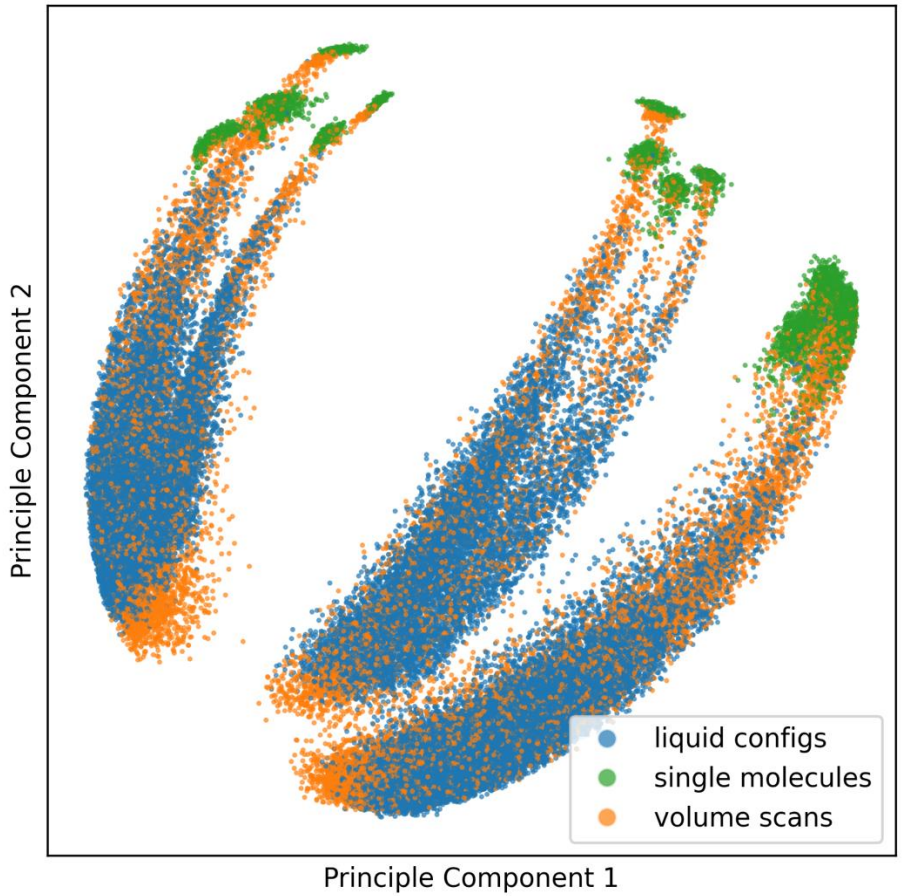


better convergence at the target composition!

stable densities all compositions!

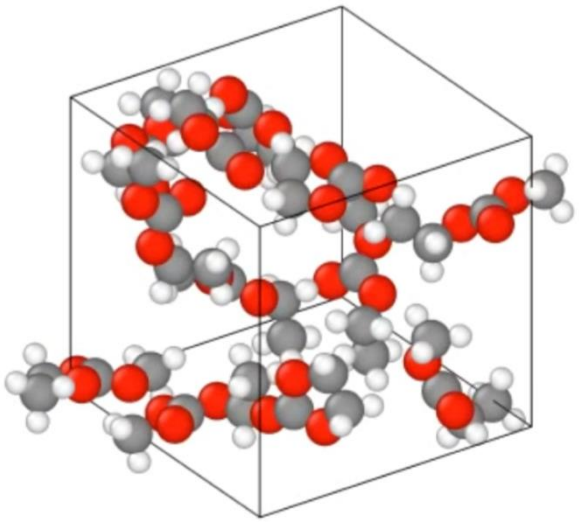
persisting unphysical density order

Volume Scan Test/Train: Pure Inter-molecular Contributions



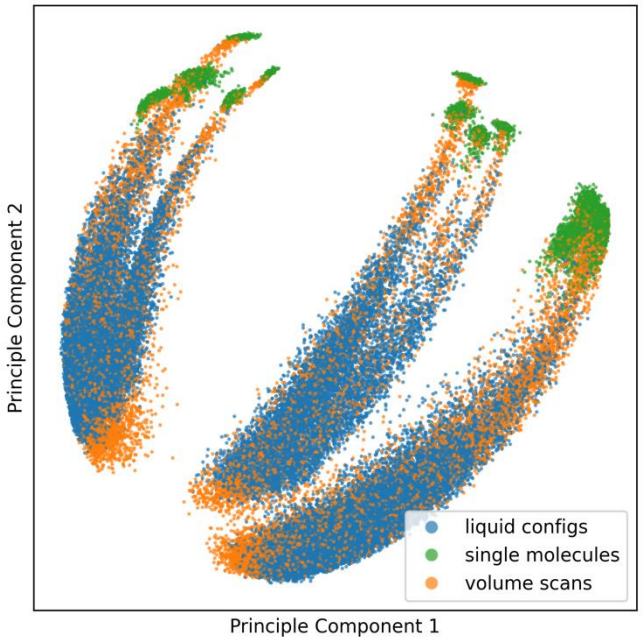
Scan inter-molecular PES:

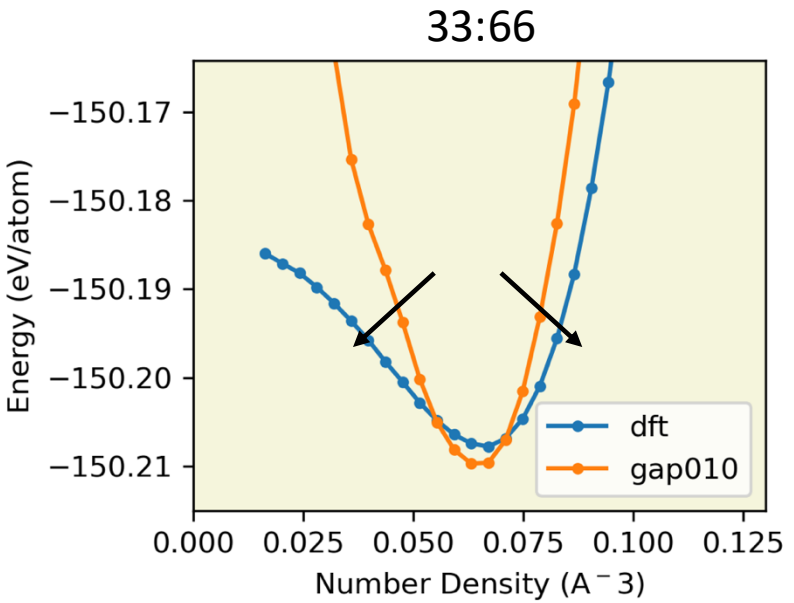
- thermalized configurations from GAP-MD
- energy binding curve with frozen molecules

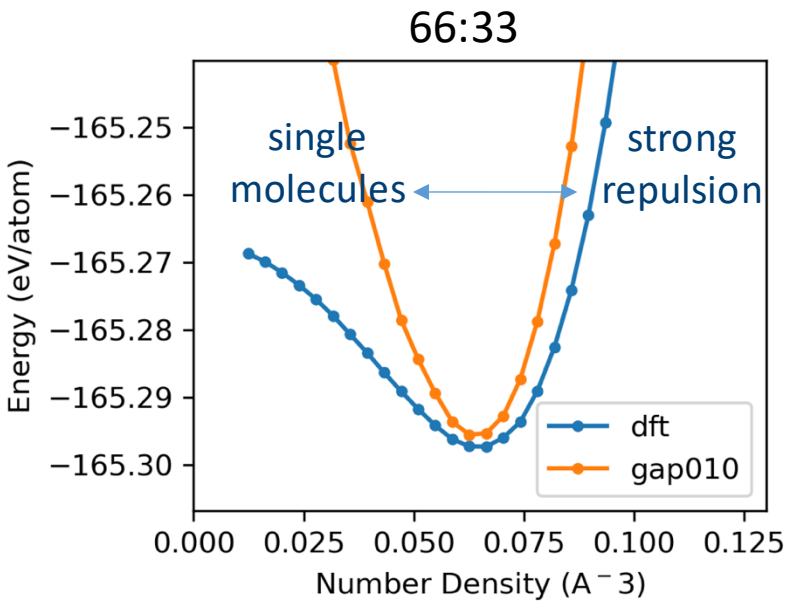


Repulsion ← Volume Scan → Molecules

Volume Scan Test/Train: Pure Inter-molecular Contributions



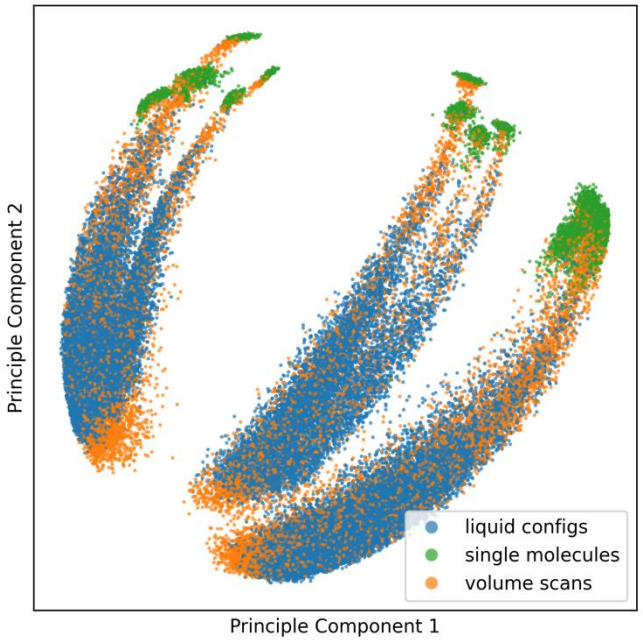


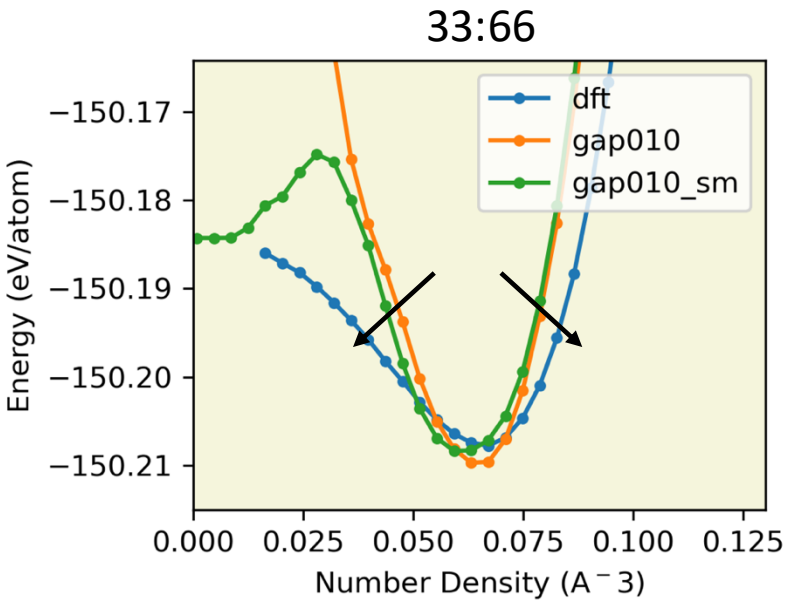


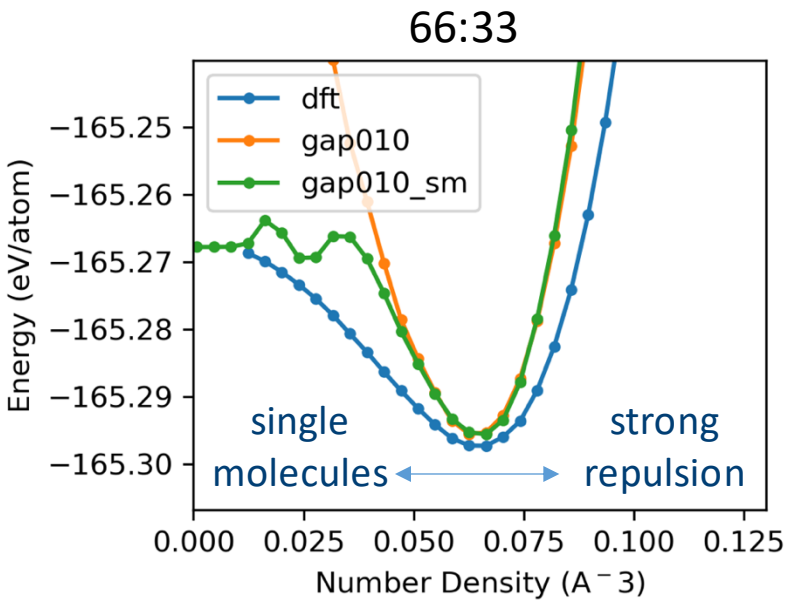
GAP

- describes bottom of the well
- stable densities, but values too high

Volume Scan Test/Train: Pure Inter-molecular Contributions







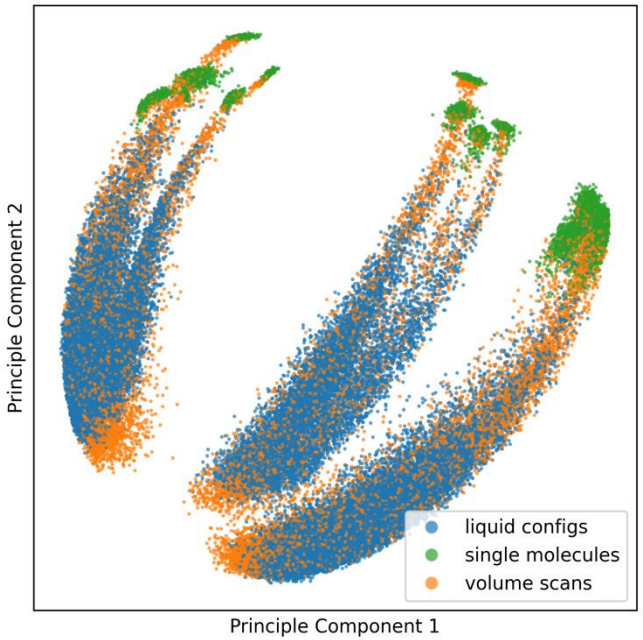
GAP

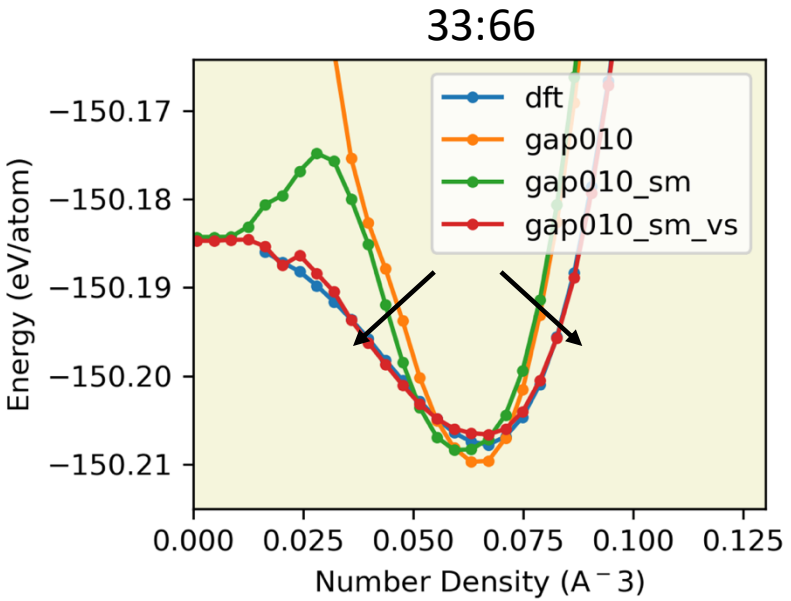
- describes bottom of the well
- stable densities, but values too high

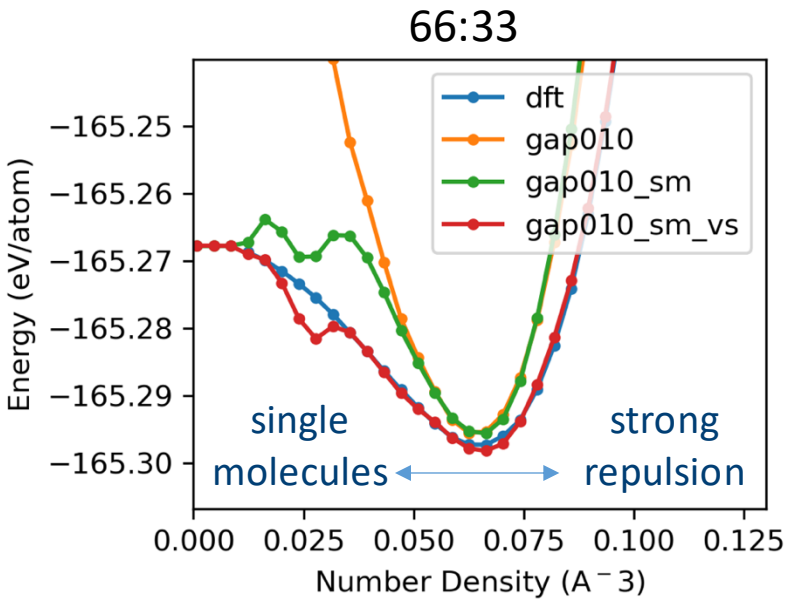
GAP + Single Molecules (SM)

- reproduces the free molecule limit
- density values still too high

Volume Scan Test/Train: Pure Inter-molecular Contributions







GAP

- describes bottom of the well
- stable densities, but values too high

GAP + Single Molecules (SM)

- reproduces the free molecule limit
- density values still too high

GAP + SM + Volume Scans (VS)

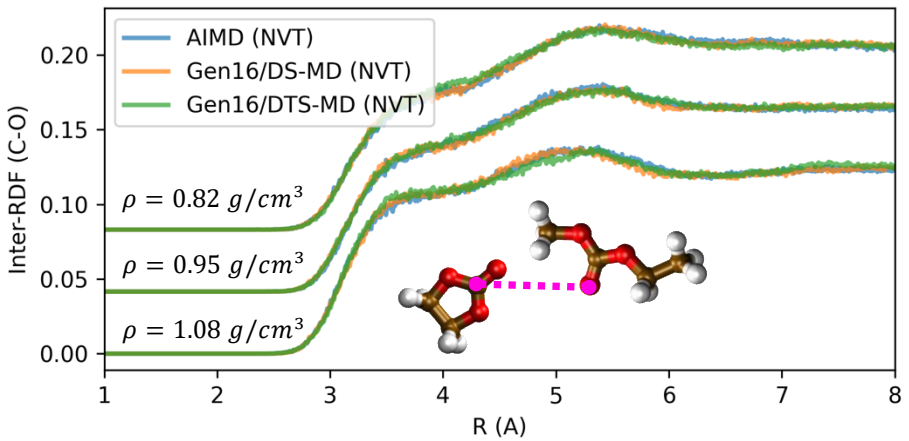
- reproduces the entire E(V) curve
- captures the strong repulsion limit
- still some **non-smooth behavior**

2 Postdoc Years: GAP-MD vs AIMD on PBE-D3

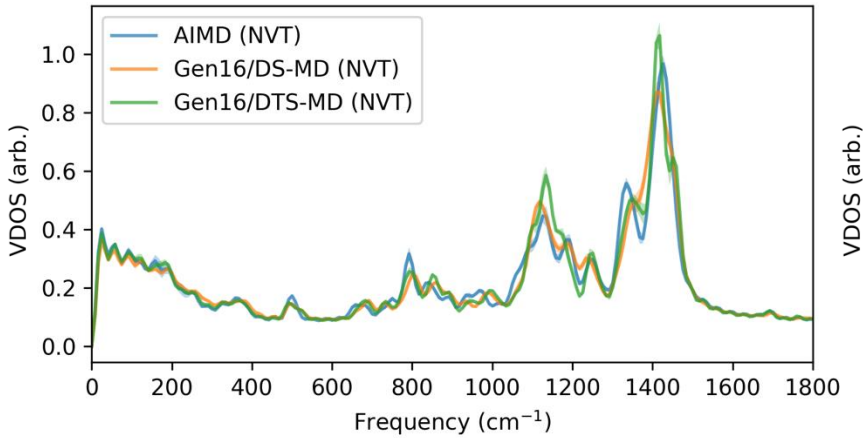
- 3x NVT 20ps simulations, 500 K, 640 atoms, EC:EMC (3:7 M)
- each AIMD **0.5 M core-hours**
- each ML-MD **250 core-hours: 2000x**

Magdău et al, NPJ 2023

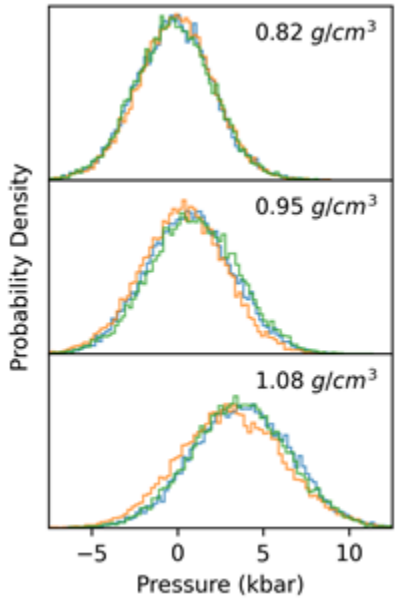
Inter-molecular RDFs



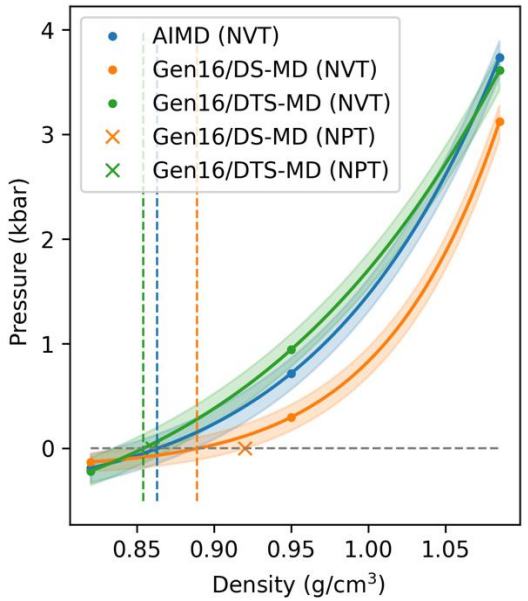
Vibrational Density of States (VDoS)



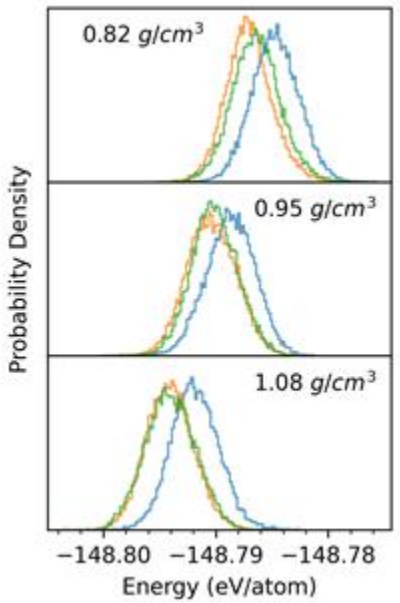
Pressure Distribution



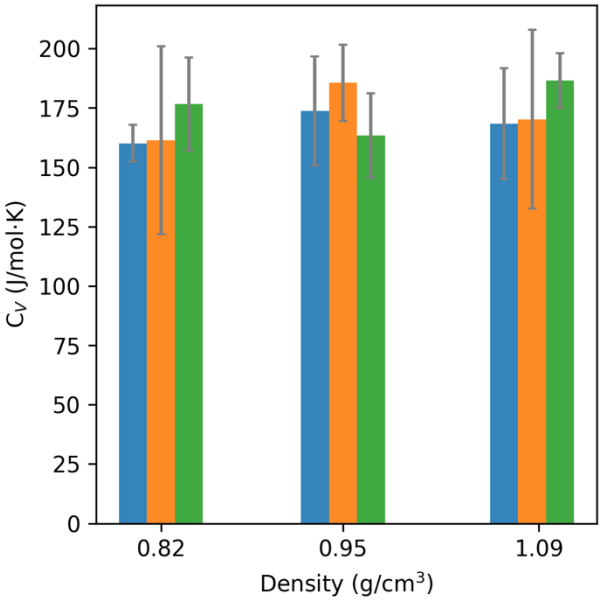
Equation of State (EoS)



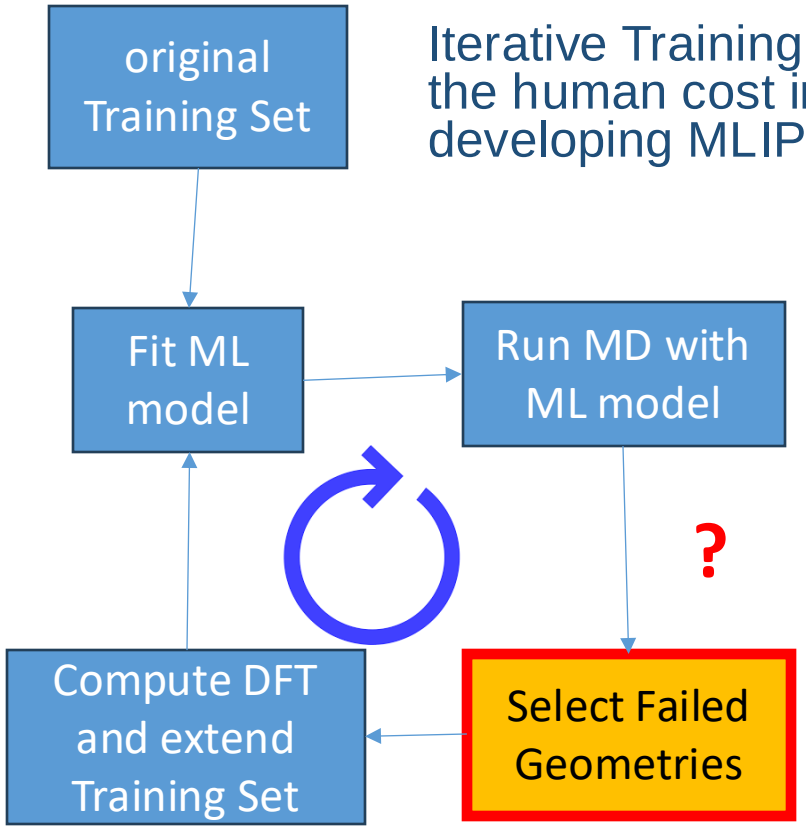
Energy Distribution



Specific Heat



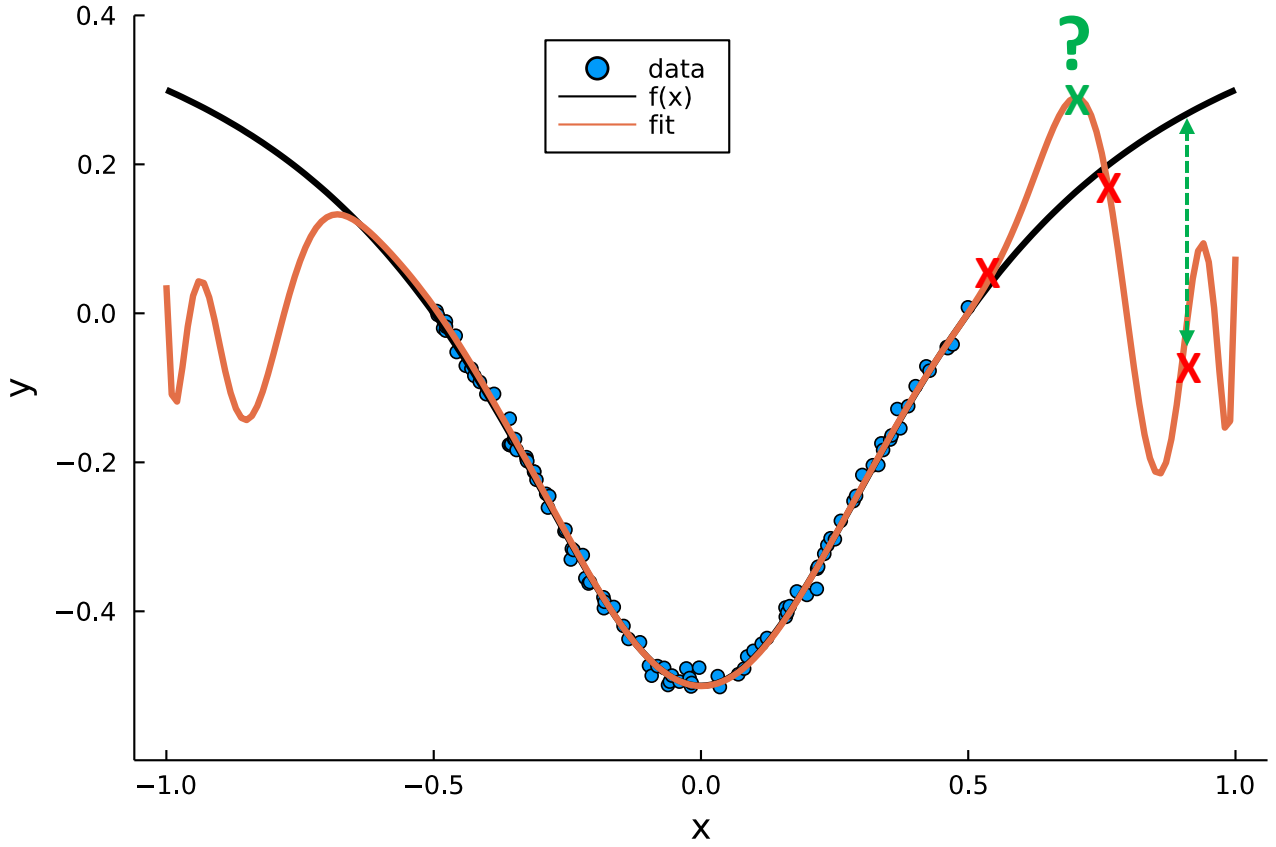
Iterative Training: How to best choose 'failed' configs?



Iterative Training: most of the human cost in developing MLIPs

Active Learning is a way to automate the Iterative Training by choosing new configs automatically, for example based on **true error**.

Which configs to add back to training?



In practice, hard to know **true error**, because it can be very expensive to compute!

Uncertainty Prediction – Bayesian View

Linear/Kernel models:

$$\mathbf{y} = \boldsymbol{\Psi}\mathbf{c} + \boldsymbol{\epsilon}$$

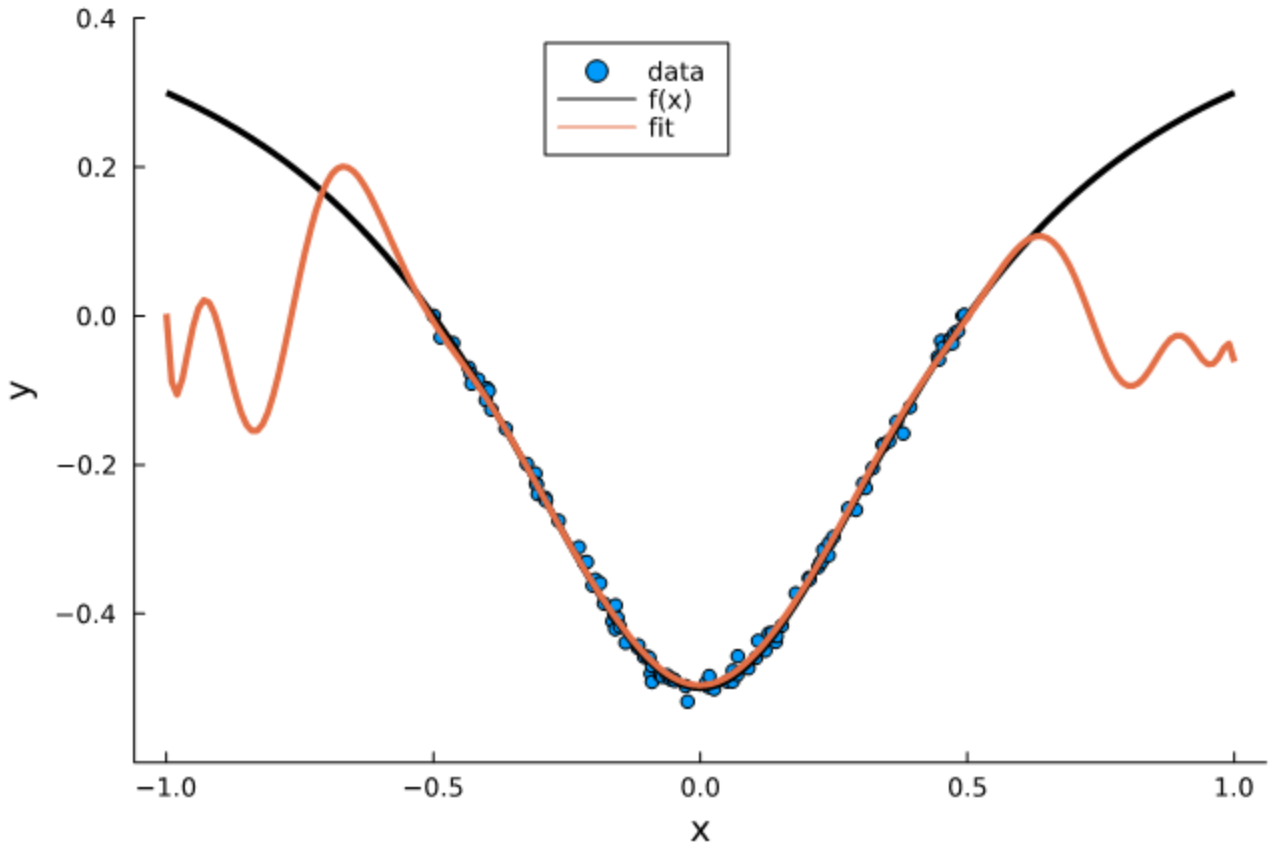
$$Posterior = \frac{Likelihood * Prior}{Normalization}$$

We can use Bayesian Regression to compute **both mean and variance:**

$$\bar{\mathbf{c}} = \lambda \boldsymbol{\Sigma} \boldsymbol{\Psi}^T \mathbf{y}$$

$$\sigma = \sqrt{\frac{1}{\lambda} + \boldsymbol{B}^T \boldsymbol{\Sigma} \boldsymbol{B}}$$

Which configs to add back to training?



A good read: https://towardsdatascience.com/introduction-to-bayesian-linear-regression-e66e60791ea7

Uncertainty Prediction – Committees

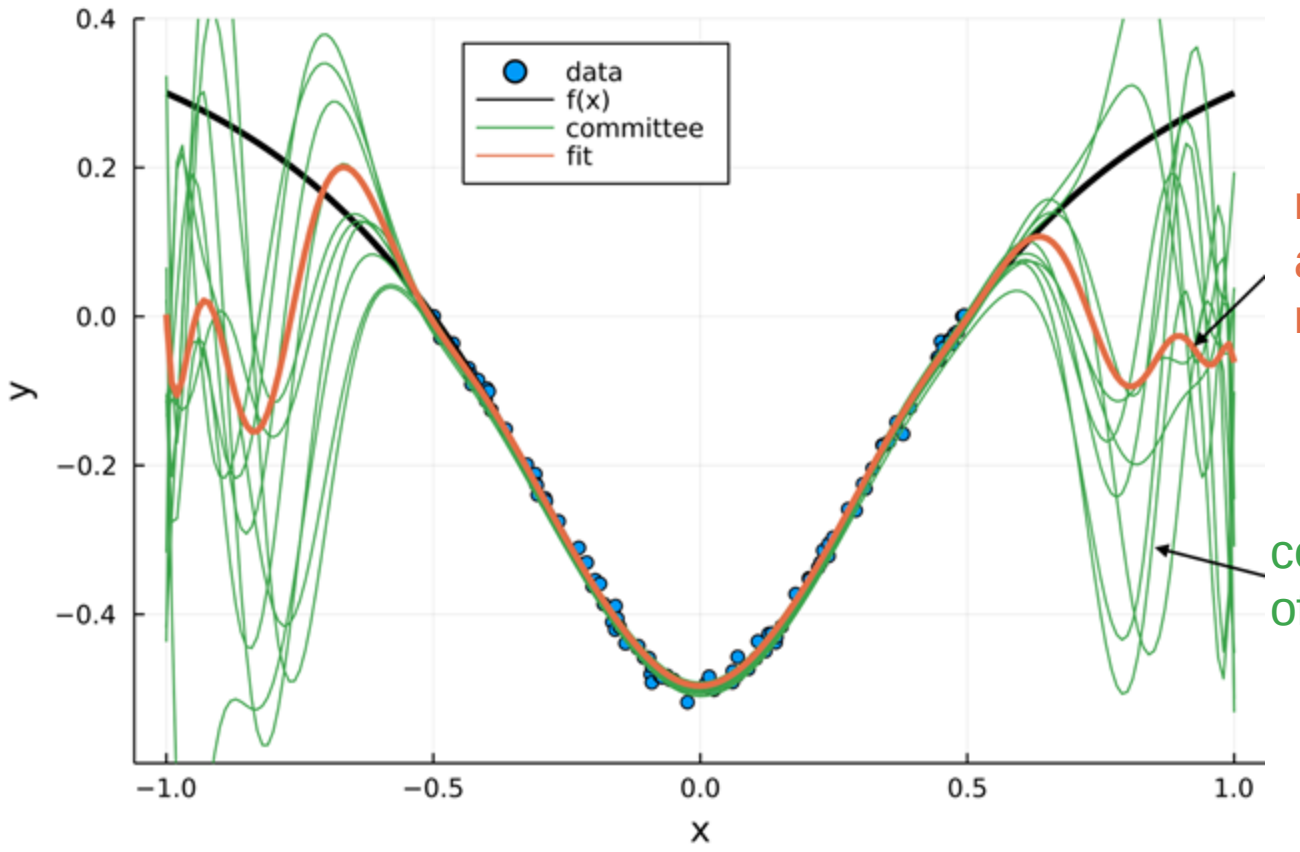
In practice, analytical error estimation can be very expensive

$$O(N_{\text{basis}}^2)$$

Use committee models: 'basket' of models with different solutions

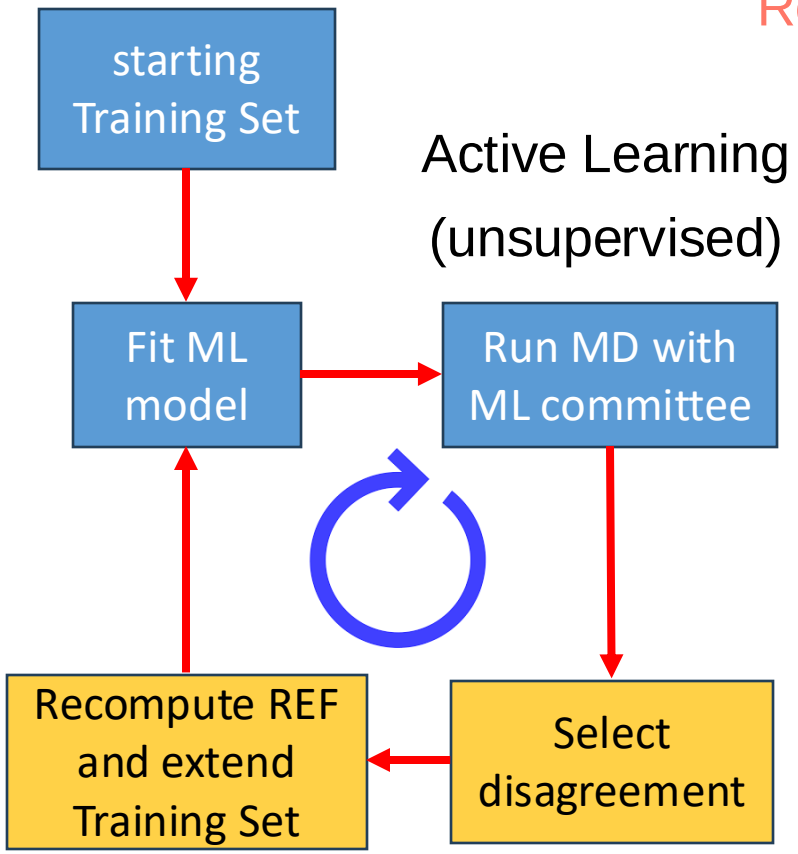
$$\tilde{\sigma} = \sqrt{\frac{1}{\lambda} + \frac{1}{K}\sum_{k=1}^{K}(E^k - \bar{E})^2}$$

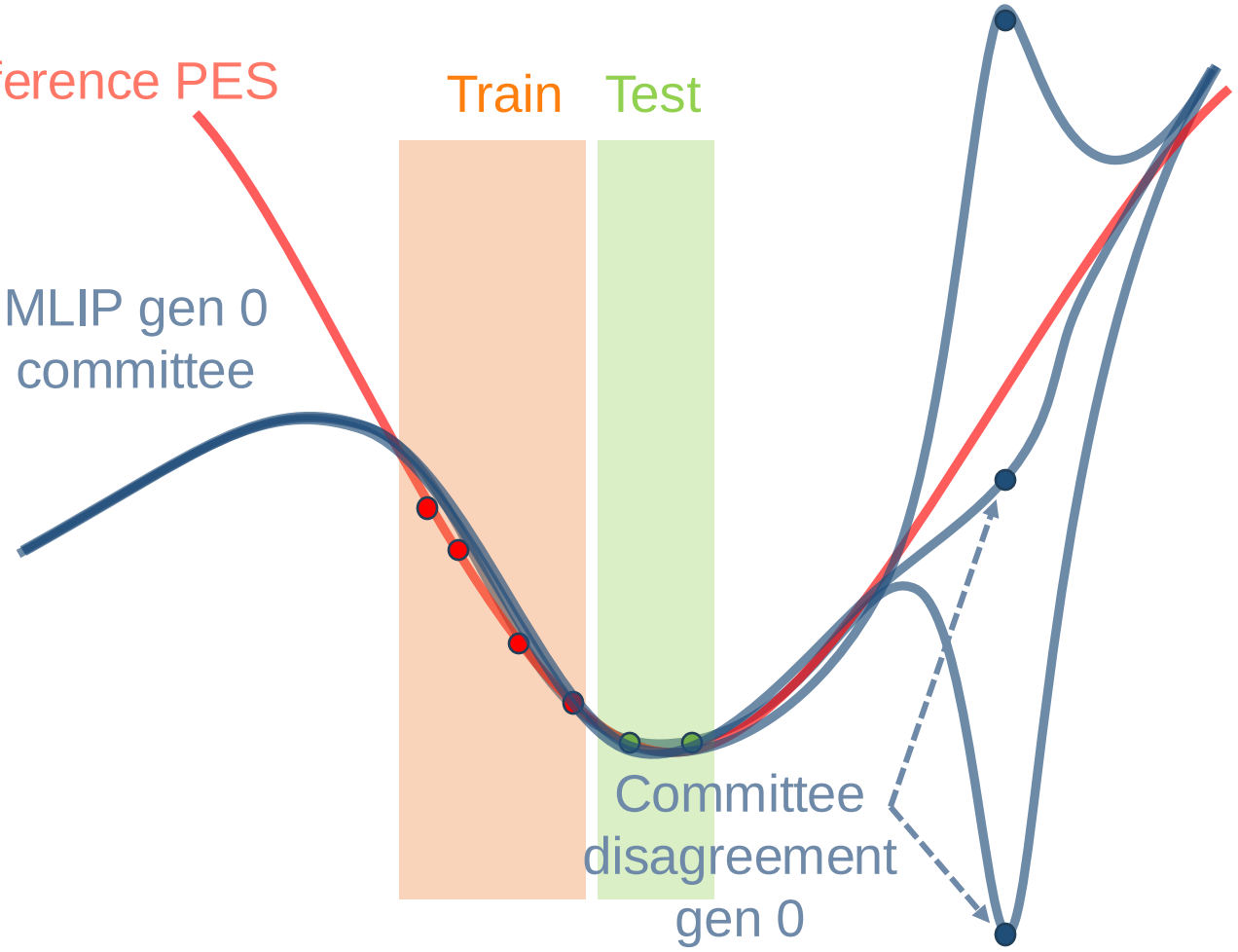
Advantage: committees work for highly-nonlinear Neural Networks (NN) as well: MACE



Uncertainty Prediction guides Active Learning

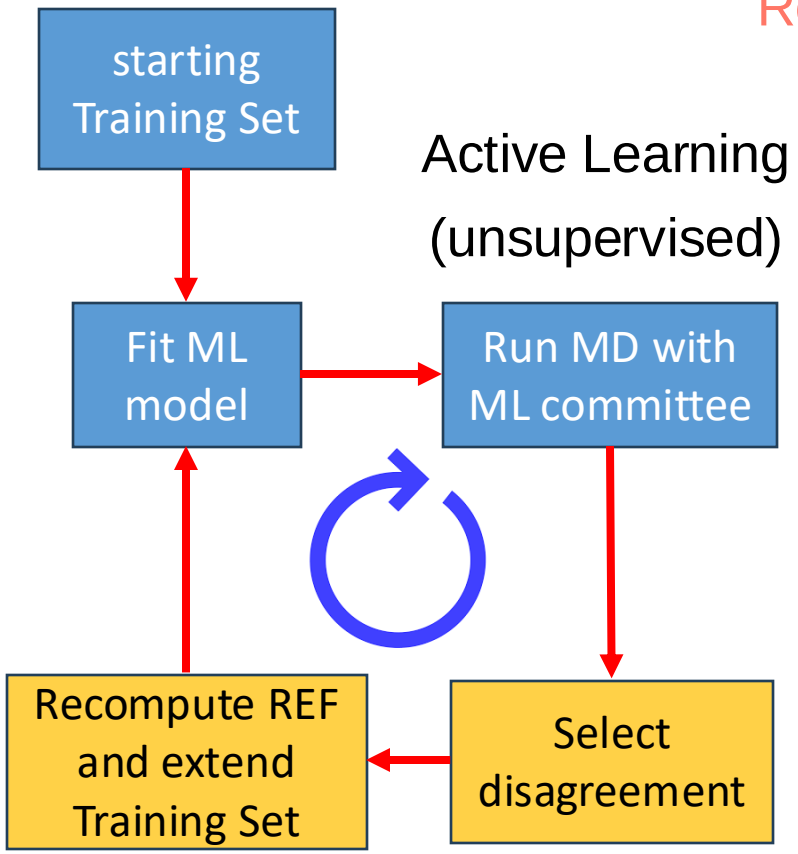
In practice, hard to know true error, assume it is predicted by the committee disagreement

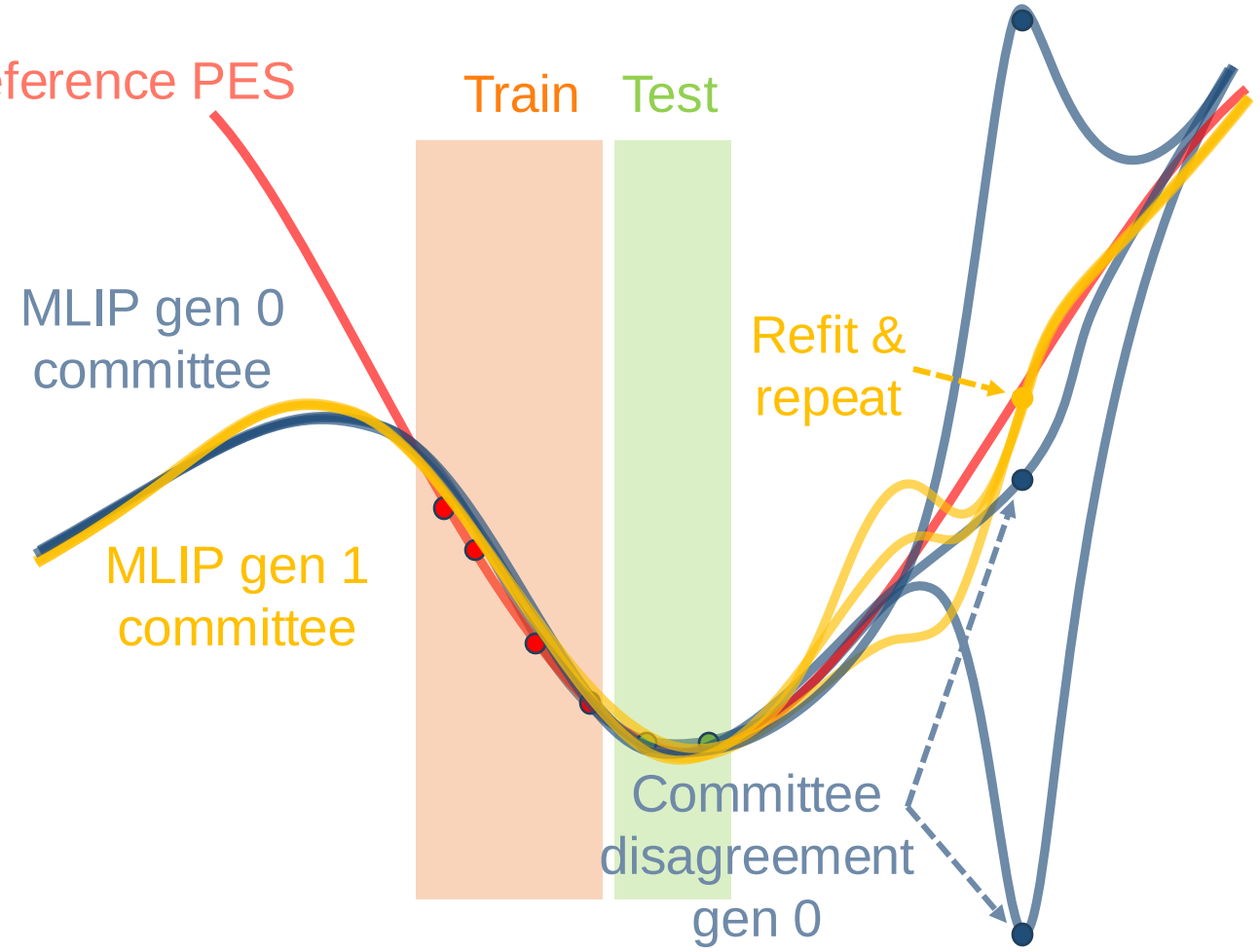




Uncertainty Prediction guides Active Learning

In practice, hard to know true error, assume it is predicted by the committee disagreement





Neural Network (MACE) Committee Models

Strategies for building NN committees:

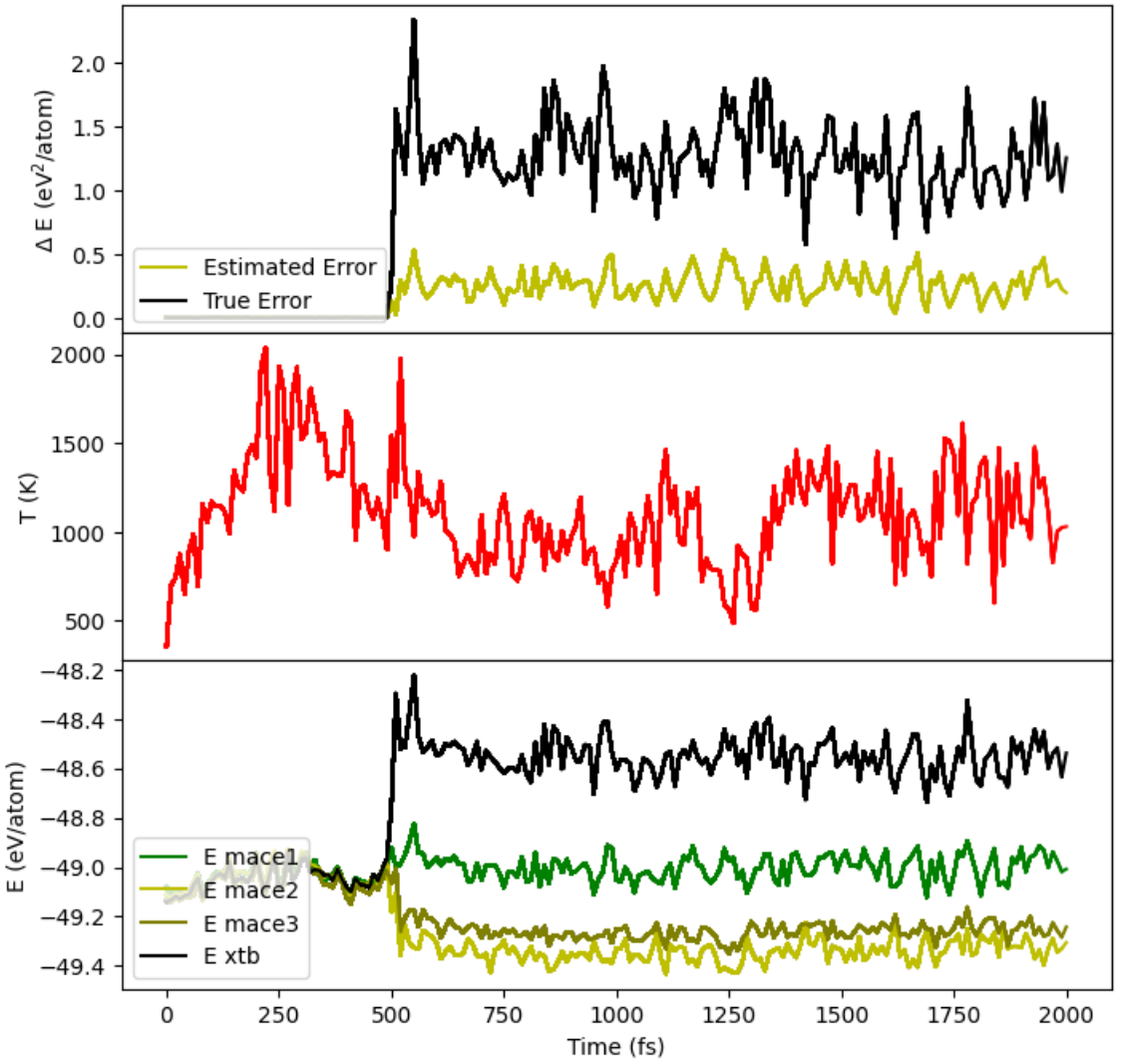
more diverse

- different training seeds → solutions
- different representative data subsets
- slightly different hyperparameters

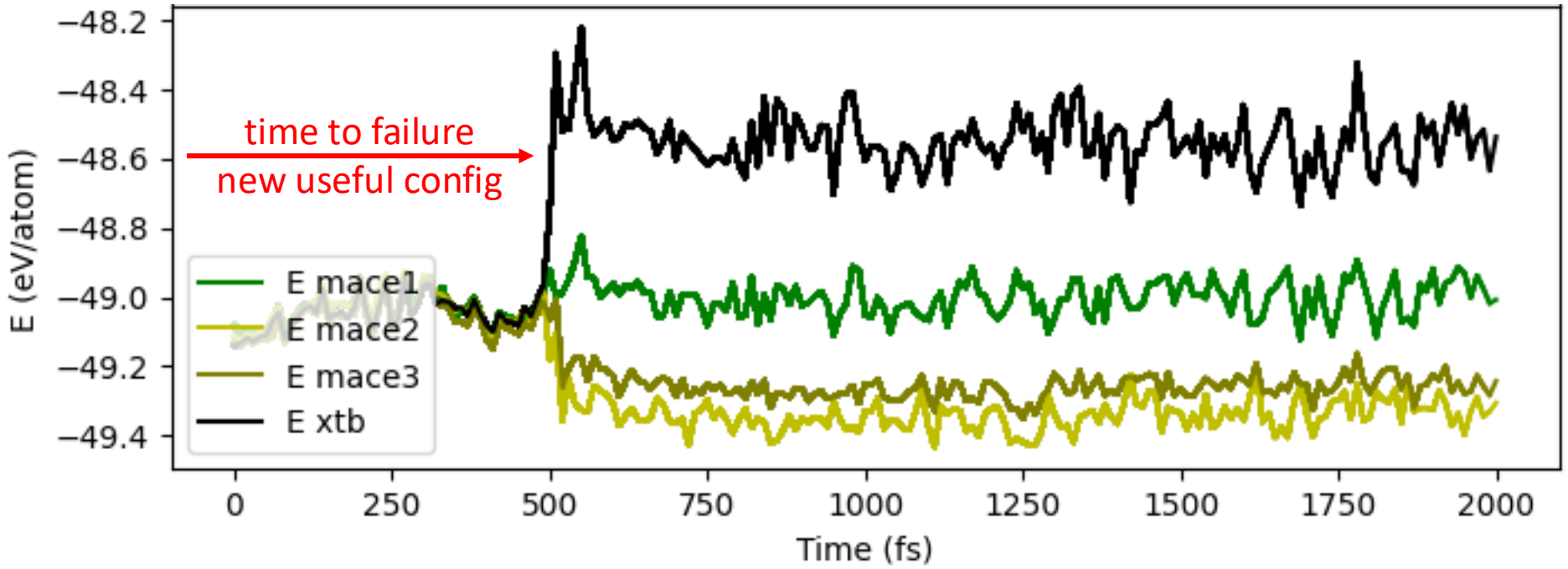
MACE result on EC/EMC cluster configs labeled with XTB (semiempirical/fast)

Shows convincing model disagreement (uncertainty) when true error explodes

Good correlation between uncertainty and true error!



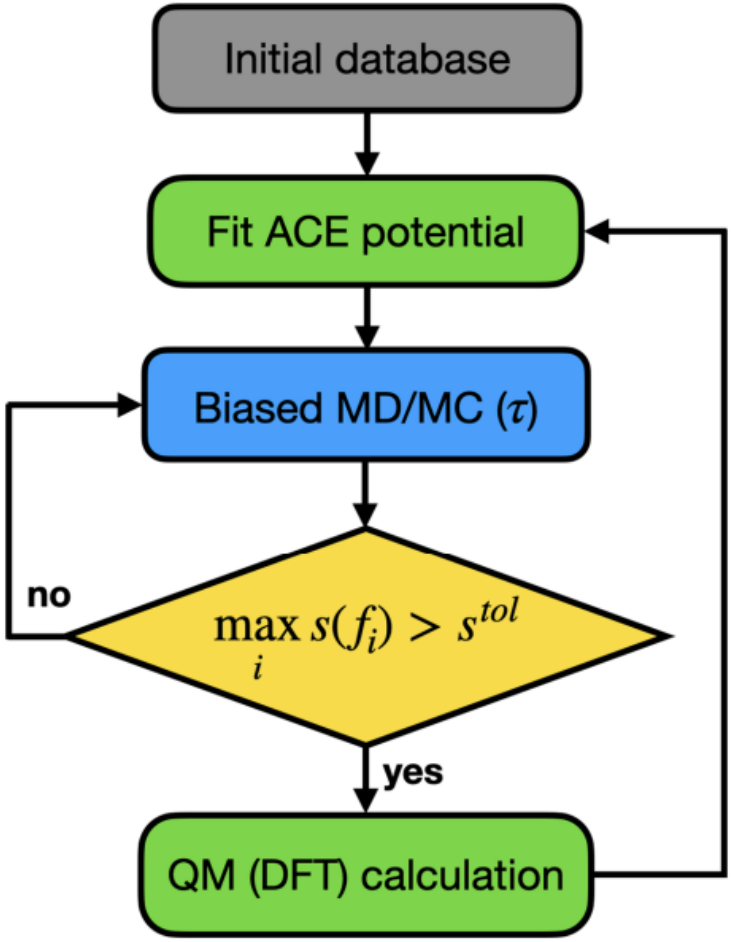
Beyond Active Learning



Often it can take a long time until we find a failed config (useful new data). This is can amount to a lot of wasted computing time.

This problem becomes more pronounced as the models get better. Can we find a way to speed up the failure?

Hyper-active Learning (HAL)



$$E_{\mathrm{HAL}} := E - \tau\sigma$$

$$E = \Sigma_i E_i = \mathbf{c} \cdot \boldsymbol{B}$$

$$\tilde{\sigma} = \sqrt{\frac{1}{\lambda} + \frac{1}{K}\sum_{k=1}^{K}(E^k - \bar{E})^2}$$

HAL uses the uncertainty prediction to bias the potential energy surface (PES)

This pushes the simulation to areas of high uncertainty


npj | computational materials

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › npj computational materials › articles › article

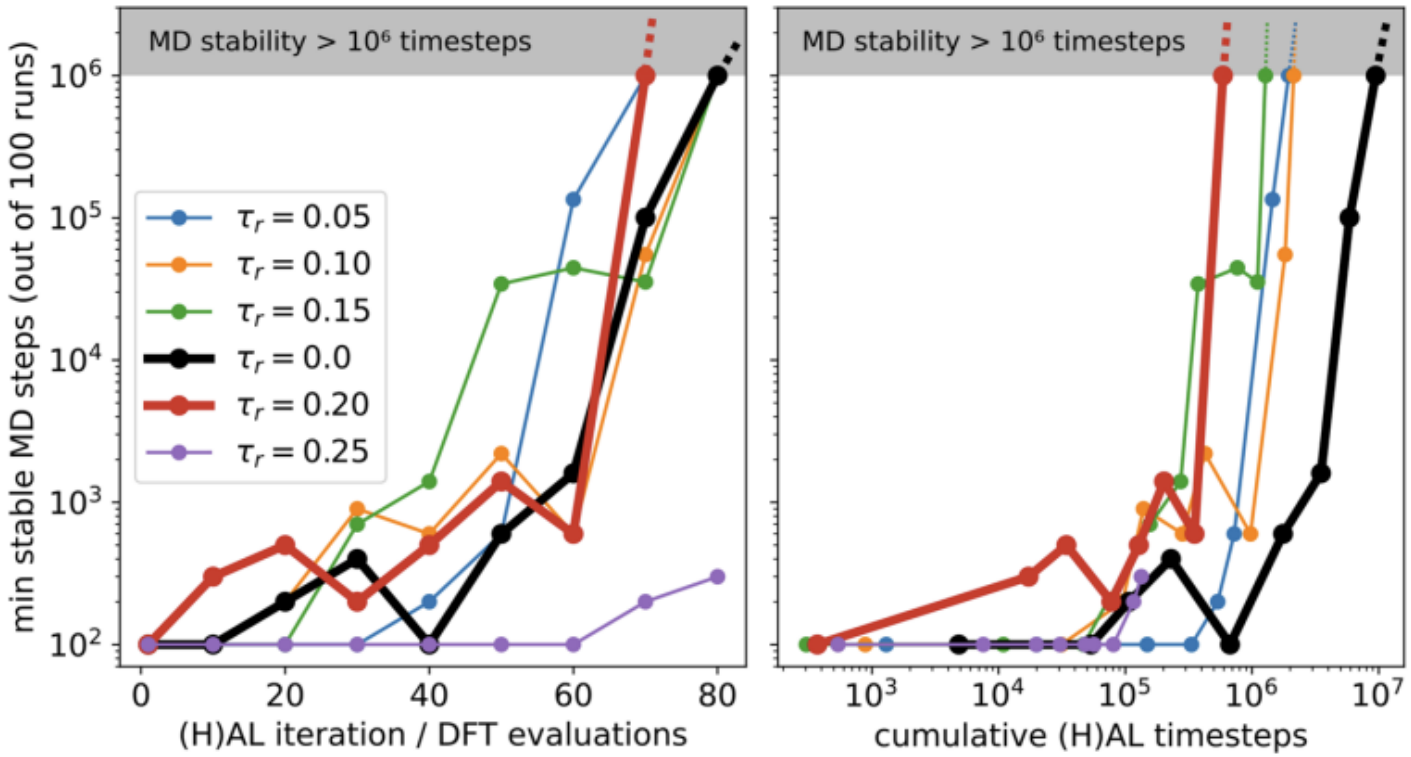
Article | Open access | Published: 13 September 2023

Hyperactive learning for data-driven interatomic potentials

Cas van der Oord ✉, Matthias Sachs, Dávid Péter Kovács, Christoph Ortner & Gábor Csányi

npj Computational Materials **9**, Article number: 168 (2023) | Cite this article

Hyper-active Learning



HAL vs simple Active Learning (AL)

HAL requires fewer iterative training cycles

HAL requires orders of magnitude less dynamics

HAL uses the uncertainty prediction to bias the potential energy surface (PES)

This pushes the simulation to areas of high uncertainty

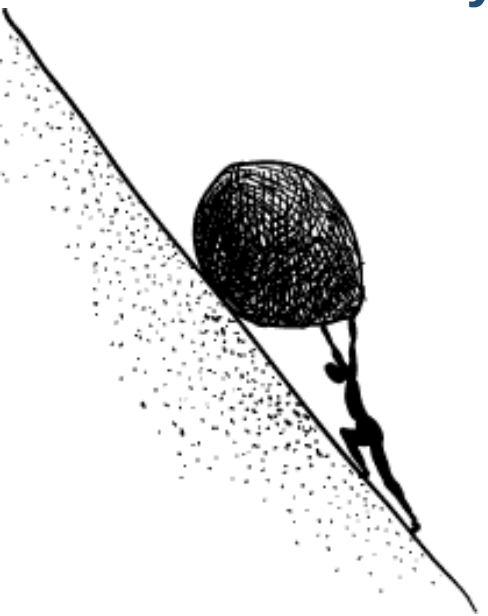
$$E_{\mathrm{HAL}} := E - \tau\sigma$$

The Advent of Foundational Models

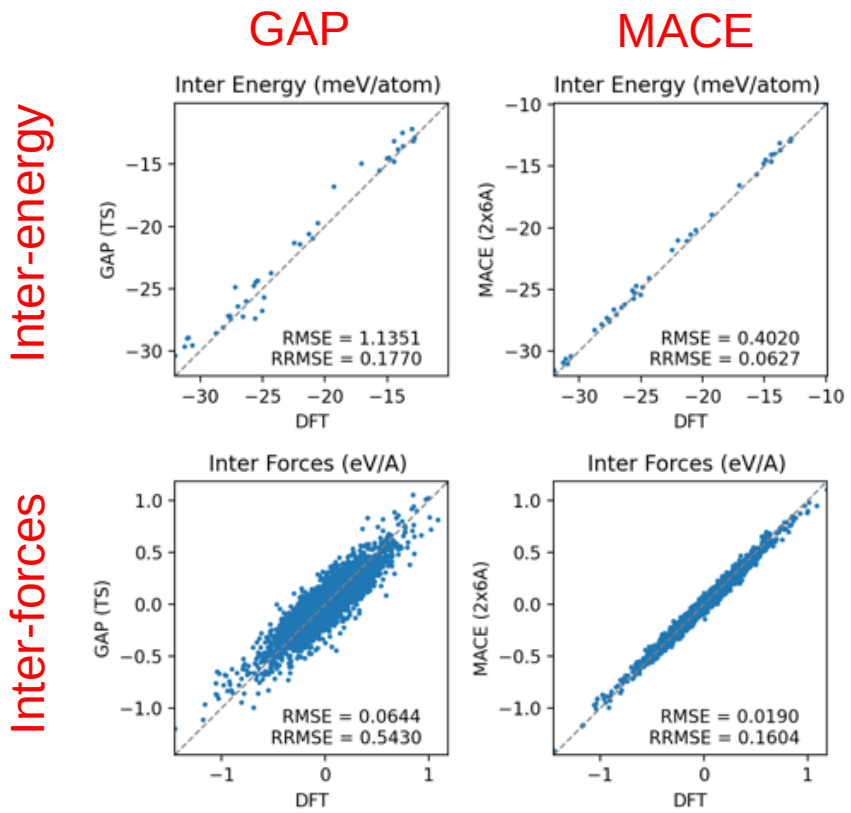
Training a MLIP for every application/system is not sustainable!

Big Question:

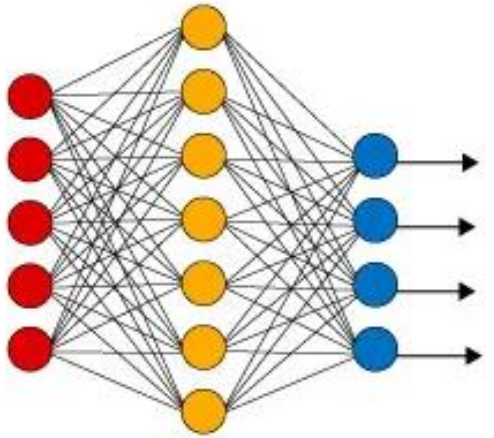
Can we train one big **foundational model** that works for **all systems**?



Message-passing Atomic Cluster Expansion (MACE)



Innovations

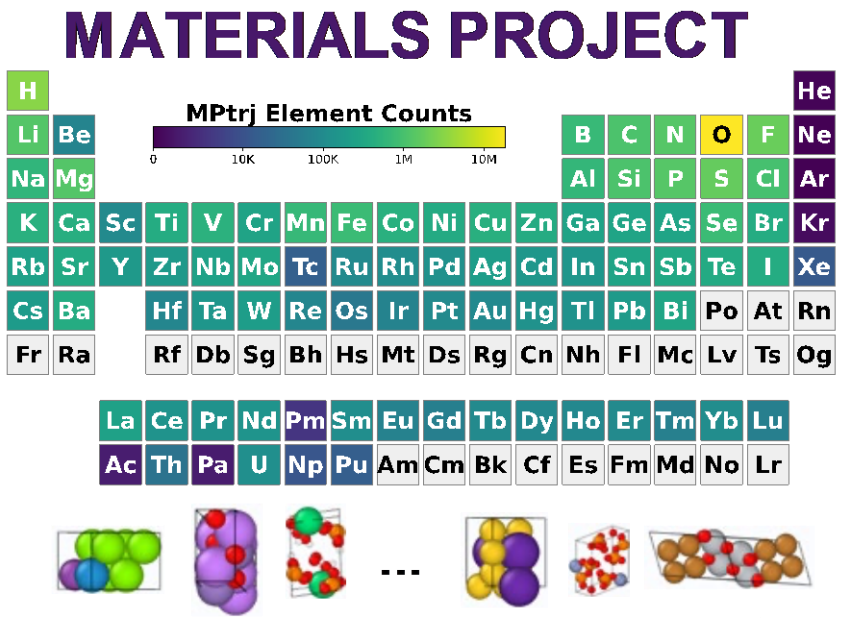


- **species embedding** → constant scaling with elements, transferability
- **body-ordered** expansion → accuracy, large capacity
- **regularity/smoothness** → stable MD
- **availability of Big Data**

MACE-MP-0: a First Generation Foundational Model

- Materials Project (Berkeley Lab): 89 elements, ~150K materials (90% < 70 atoms)
- MPtraj: ~1.5M configs + structural relaxations
- XC-functional: PBE+U

- MACE: trained on MPtraj, no iterative training



https://next-gen.materialsproject.org/

Model ⓘ	CPS ↑	F1 ↑	DAF ↑	Prec ↑	Acc ↑	MAE ↓	R^2 ↑	κ_{SRME} ↓	RMSD ↓	Training Set	Params	Targets	Date Added
eSEN-30M-OAM	0.896	0.925	6.069	0.928	0.977	0.018	0.866	0.1704	0.0096	6.6M (113M) (OMat24+MPtrj+sAlex)	30.2M	EFS_G	2025-03-17
SevenNet-MF-ompa	0.849	0.901	5.825	0.89	0.969	0.021	0.867	0.317	0.0115	6.6M (113M) (OMat24+sAlex+MPtrj)	25.7M	EFS_G	2025-03-13
GRACE-2L-OAM	0.841	0.88	5.774	0.883	0.963	0.023	0.862	0.294	0.0121	6.6M (113M) (OMat24+sAlex+MPtrj)	12.6M	EFS_G	2025-02-06
eSEN-30M-MP	0.800	0.831	5.26	0.804	0.946	0.033	0.822	0.3398	0.0142	146k (1.58M) (MPtrj)	30.1M	EFS_G	2025-03-17
MACE-MPA-0	0.796	0.852	5.582	0.853	0.954	0.028	0.842	0.412	0.0143	3.37M (12M) (MPtrj+sAlex)	9.06M	EFS_G	2024-12-09
DPA3-v2-OpenLAM	0.765	0.89	5.747	0.879	0.966	0.022	0.869	0.687	0.0127	163M (OpenLAM)	7.02M	EFS_G	2025-03-14
GRACE-1L-OAM	0.762	0.824	5.255	0.803	0.944	0.031	0.842	0.516	0.0139	6.6M (113M) (OMat24+sAlex+MPtrj)	3.45M	EFS_G	2025-02-06
MatterSim v1 5M	0.716	0.862	5.852	0.895	0.959	0.024	0.863	0.574	0.0733	17M (MatterSim)	4.55M	EFS_G	2024-12-16
SevenNet-l3i5	0.709	0.76	4.629	0.708	0.92	0.044	0.776	0.55	0.0182	146k (1.58M) (MPtrj)	1.17M	EFS_G	2024-12-10
MatRIS v0.5.0 MPtrj	0.680	0.809	5.049	0.772	0.938	0.037	0.803	0.861	0.0156	146k (1.58M) (MPtrj)	5.83M	EFS_GM	2025-03-13
GRACE-2L-MPtrj	0.678	0.691	4.163	0.636	0.896	0.052	0.741	0.525	0.0186	146k (1.58M) (MPtrj)	15.3M	EFS_G	2024-11-21
DPA3-v2-MPtrj	0.647	0.786	4.822	0.737	0.929	0.039	0.804	0.959	0.0164	146k (1.58M) (MPtrj)	4.92M	EFS_G	2025-03-14
MACE-MP-0	0.640	0.669	3.777	0.577	0.878	0.057	0.697	0.647	0.0194	146k (1.58M) (MPtrj)	4.69M	EFS_G	2023-07-14
AlphaNet-MPTrj	0.562	0.799	4.863	0.743	0.933	0.041	0.745	1.31	0.0227	146k (1.58M) (MPtrj)	16.2M	EFS_G	2025-03-05
eqV2 M	0.558	0.917	6.047	0.924	0.975	0.02	0.848	1.771	0.0138	3.37M (102M) (OMat24+MPtrj)	86.6M	EFS_D	2024-10-18
ORB v2	0.540	0.88	6.041	0.924	0.965	0.028	0.824	1.732	0.016	3.25M (32.1M) (MPtrj+Alex)	25.2M	EFS_D	2024-10-11
eqV2 S DeNS	0.526	0.815	5.042	0.771	0.941	0.036	0.788	1.676	0.0138	146k (1.58M) (MPtrj)	31.2M	EFS_D	2024-10-18
ORB v2 MPtrj	0.476	0.765	4.702	0.719	0.922	0.045	0.756	1.725	0.0185	146k (1.58M) (MPtrj)	25.2M	EFS_D	2024-10-14
M3GNet	0.430	0.569	2.882	0.441	0.813	0.075	0.585	1.412	0.0217	62.8k (188k) (MPF)	228k	EFS_G	2022-09-20
CHGNet	0.391	0.613	3.361	0.514	0.851	0.063	0.689	1.717	0.0216	146k (1.58M) (MPtrj)	413k	EFS_GM	2023-03-03
GNoME	n/a	0.829	5.523	0.844	0.955	0.035	0.785	n/a	n/a	6M (89M) (GNoME)	16.2M	EF_G	2024-02-03

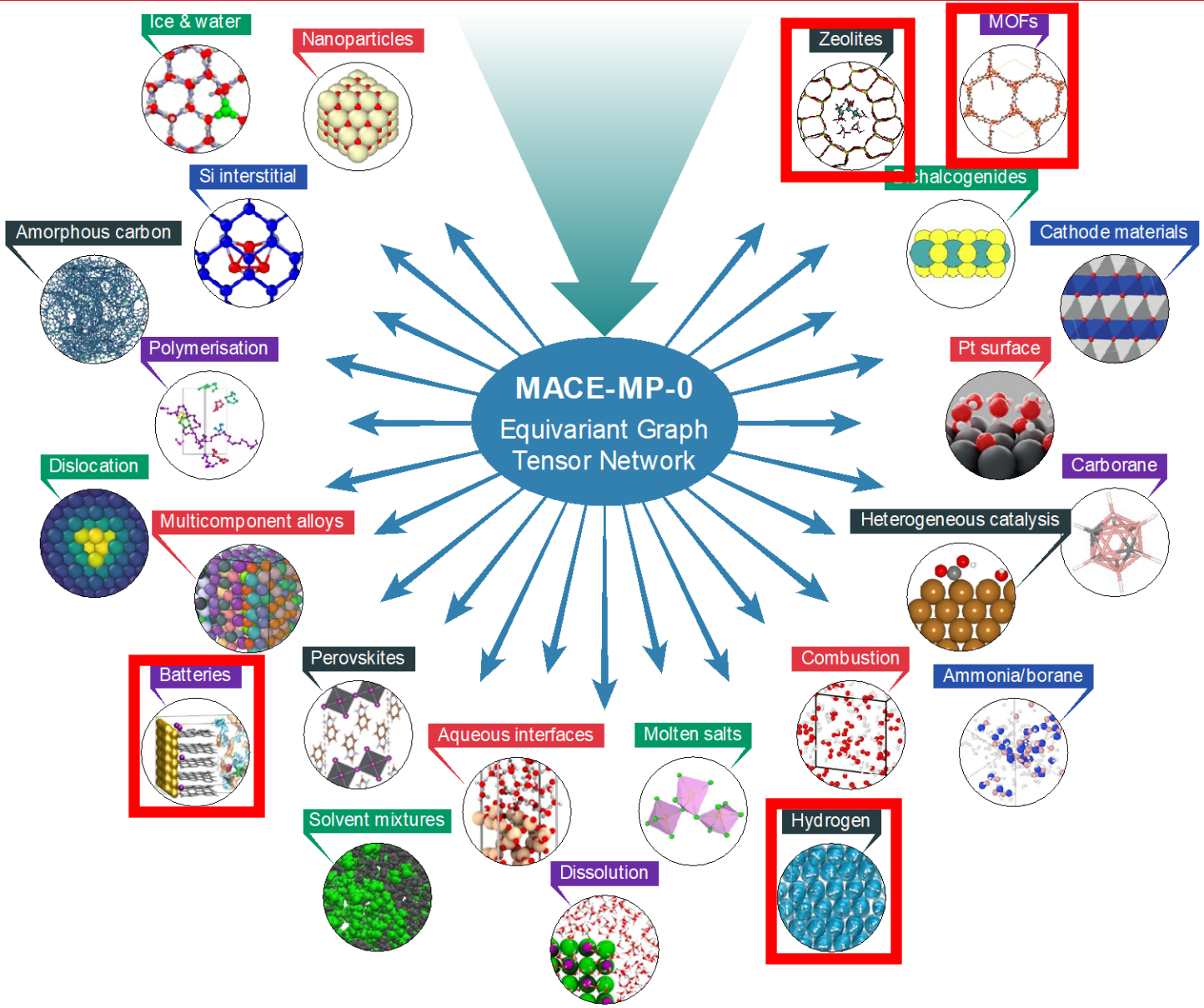
The Emergence of Foundational Models

Batatia, I et al, arXiv preprint, arXiv:2401.00096 (2023)

A foundation model for atomistic materials chemistry

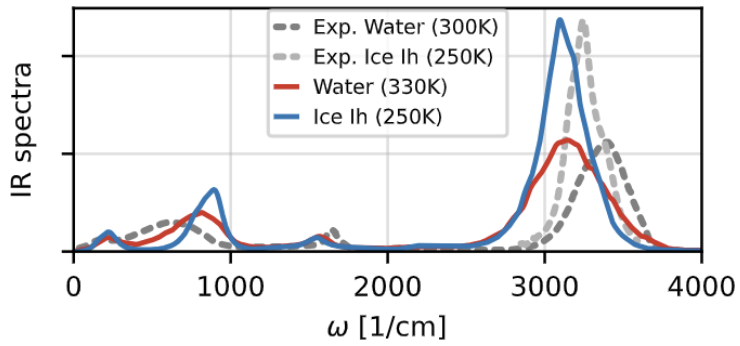
Ilyes Batatia[†1], Philipp Benner[†2], Yuan Chiang[†3,4], Alin M. Elena[†17], Dávid P. Kovács[†1], Janosh Riebesell[†4,13], Xavier R. Advincula[12,13], Mark Asta[3,4], Matthew Avaylon[30], William J. Baldwin[1], Fabian Berger[12], Noam Bernstein[11], Arghya Bhowmik[25], Samuel M. Blau[10], Vlad Cărare[1,13], James P. Darby[1], Sandip De[18], Flaviano Della Pia[12], Volker L. Deringer[16], Rokas Elijošius[1], Zakariya El-Machachi[16], Fabio Falcioni[31], Edvin Fako[18], Andrea C. Ferrari[26], Annalena Genreith-Schriever[12], Janine George[2,6], Rhys E. A. Goodall[15], Clare P. Grey[12], Petr Grigorev[27], Shuang Han[18], Will Handley[13,19], Hendrik H. Heenen[9], Kersti Hermansson[23], Christian Holm[22], Stephan Hofmann[1], Jad Jaafar[1], Konstantin S. Jakob[9], Hyunwook Jung[9], Venkat Kapil[12, 21], Aaron D. Kaplan[4], Nima Karimitari[20], James R. Kermode[28], Namu Kroupa[13,19,1], Jolla Kullgren[23], Matthew C. Kuner[3,4], Domantas Kuryla[12], Guoda Liepuoniute[1,26], Johannes T. Margraf[8], Ioan-Bogdan Magdău[24], Angelos Michaelides[12], J. Harry Moore[1], Aakash A. Naik[2,6], Samuel P. Niblett[12], Sam Walton Norwood[25], Niamh O'Neill[12,13], Christoph Ortner[5], Kristin A. Persson[3,4,7], Karsten Reuter[9], Andrew S. Rosen[3,4], Lars L. Schaaf[1], Christoph Schran[13], Benjamin X. Shi[12], Eric Sivonxay[10], Tamás K. Stenczel[1], Viktor Svahn[23], Christopher Sutton[20], Thomas D. Swinburne[27], Jules Tilly[31], Cas van der Oord[1], Santiago Vargas[29], Eszter Varga-Umbrich[1], Tejs Vegge[25], Martin Vondrák[8,9], Yangshuai Wang[5], William C. Witt[14], Fabian Zills[22], and Gábor Csányi[*1]

- **MD-stable out-of-box**
- **New frontiers in molecular modelling**
- **vast range of applications: organic, inorganic, interfaces, full devices**

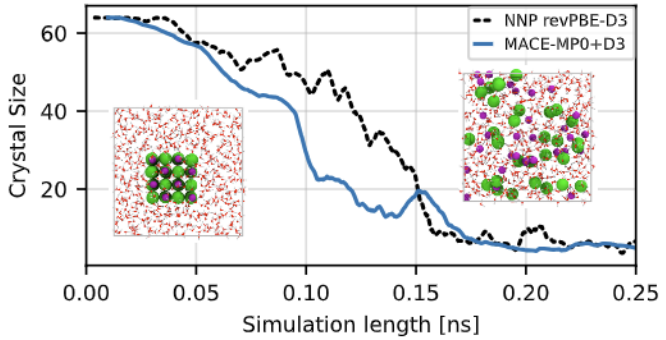


Now possible to compute with ab initio accuracy!

IR spectra of water

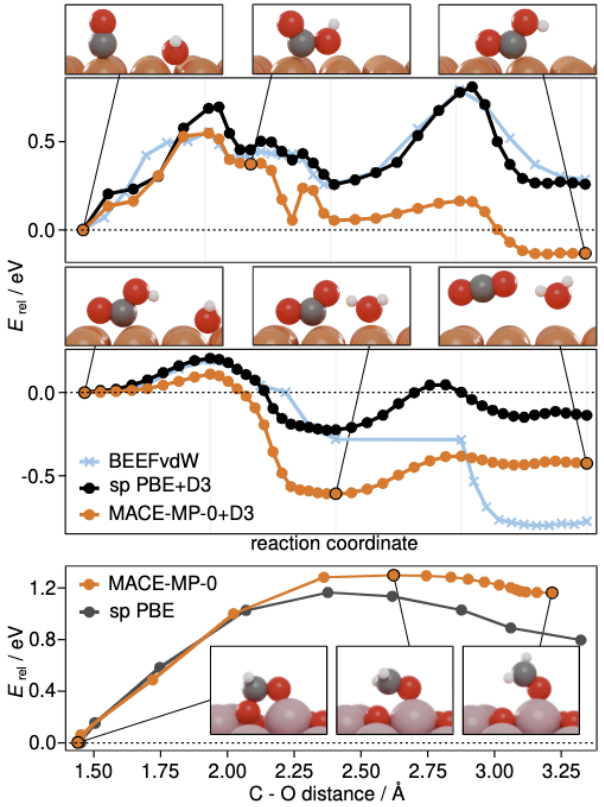


NaCl dissolution

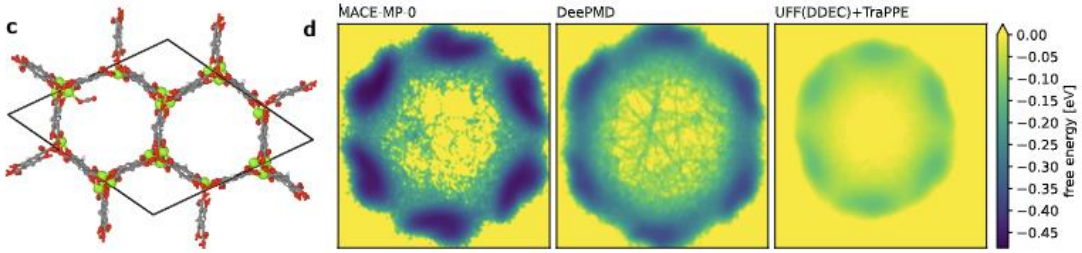


Nudge-Elastic-Band

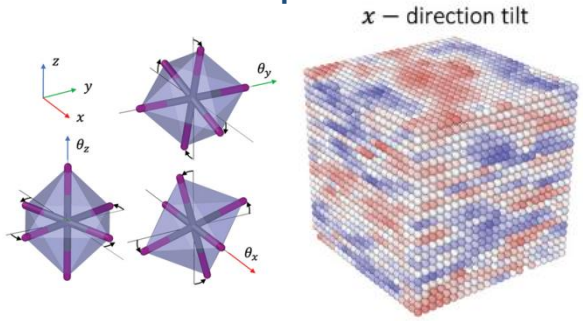
multistep catalysis on surfaces



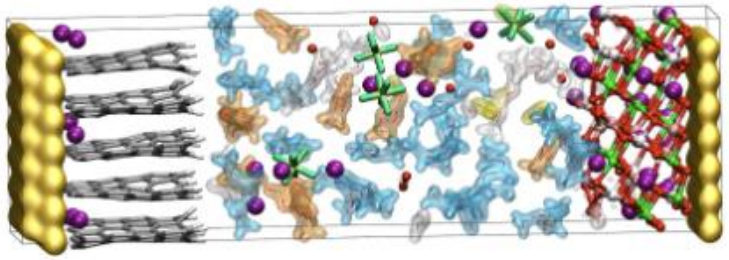
Free energy landscape: CO_2 in MOF



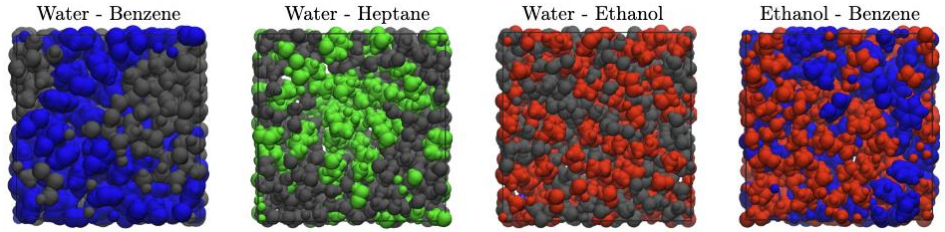
Million atom perovskite



Formation of SEI in Li-ion battery

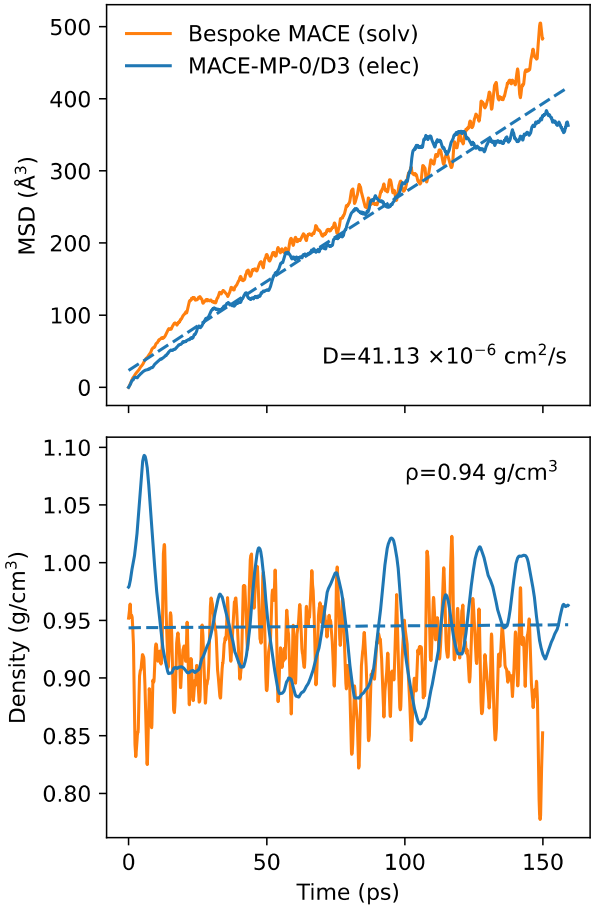


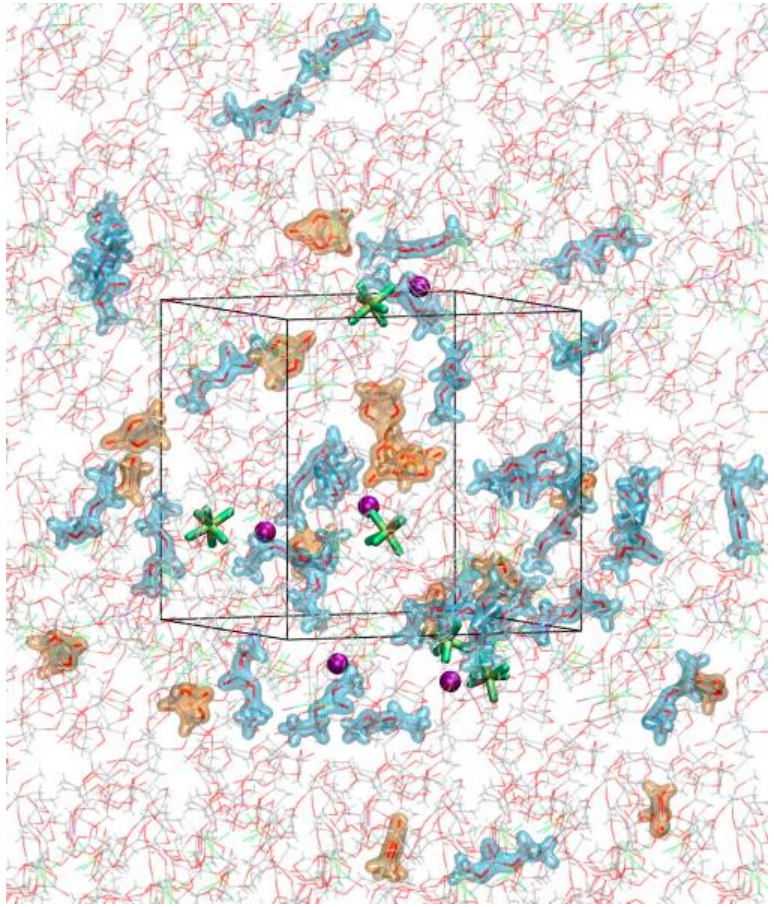
Solvent mixing: ~ns timescale



Back to the Electrolyte: stability

EC/EMC $LiPF_6$, 500K NPT, MACE-MP-0 (PBE+D3)
EC/EMC, 500K NPT, bespoke MACE model (PBE+D2)



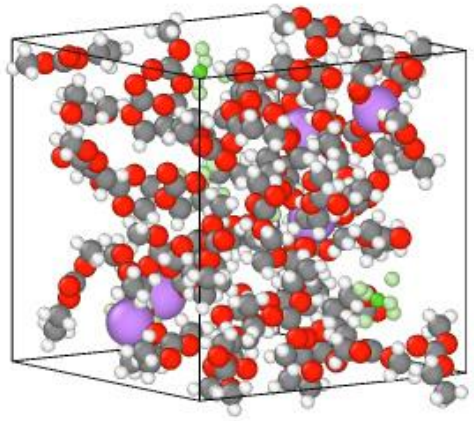


Out-of-box MACE-MP-0

no iterative training

simulation is stable at 500K, NPT, 150ps:

- molecules remain intact!
- density is stable!
- electrolyte is flowing, exploring phase-space



Back to the Electrolyte: Accuracy

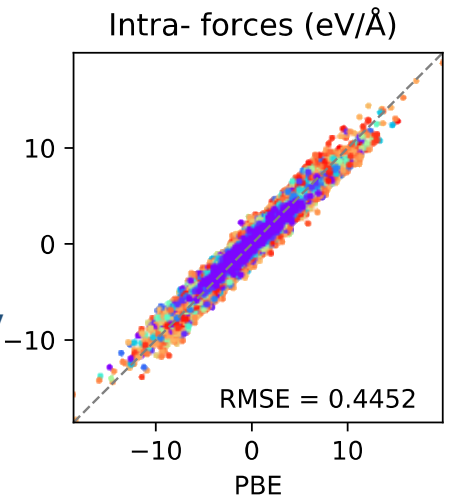
B. Intra-Inter errors

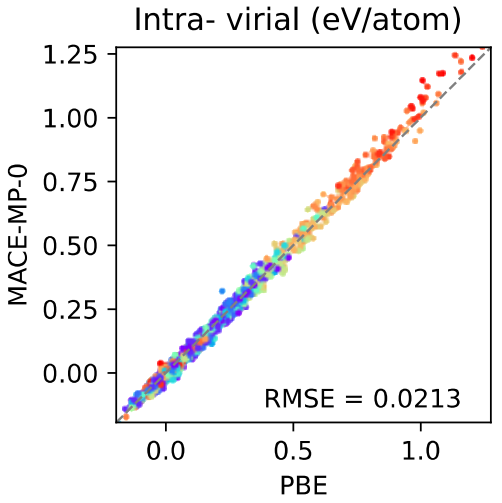
- independent PBE test set
- 0.1-2.5 g/cm^3, all EC/EMC $LiPF_6$ compositions
- largest errors at unphysically high-density

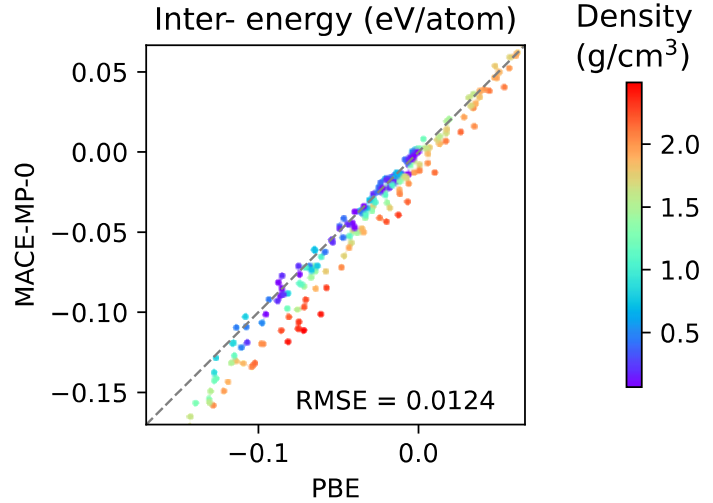
C. Rigid-molec. Volume Scans

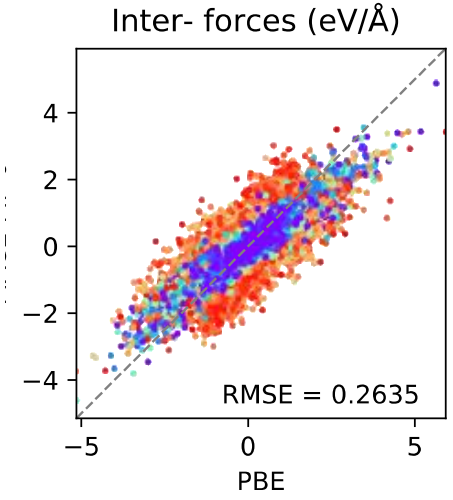
- neat solvent
- full electrolyte (incl. ions)
- surprisingly good performance
- notice the potential reproduces the 1/r behavior without formal charges

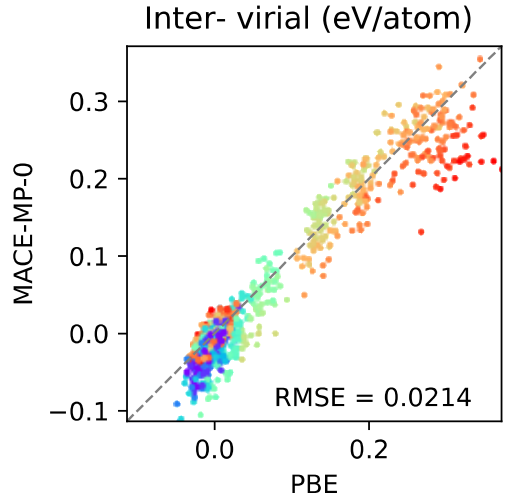
B. Intra/inter errors for electrolyte



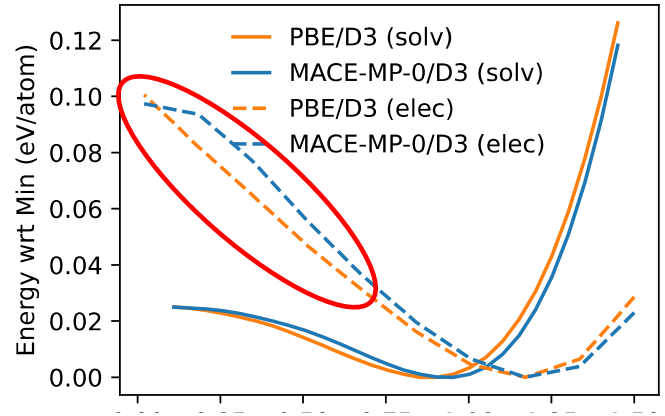








C. Volume scans

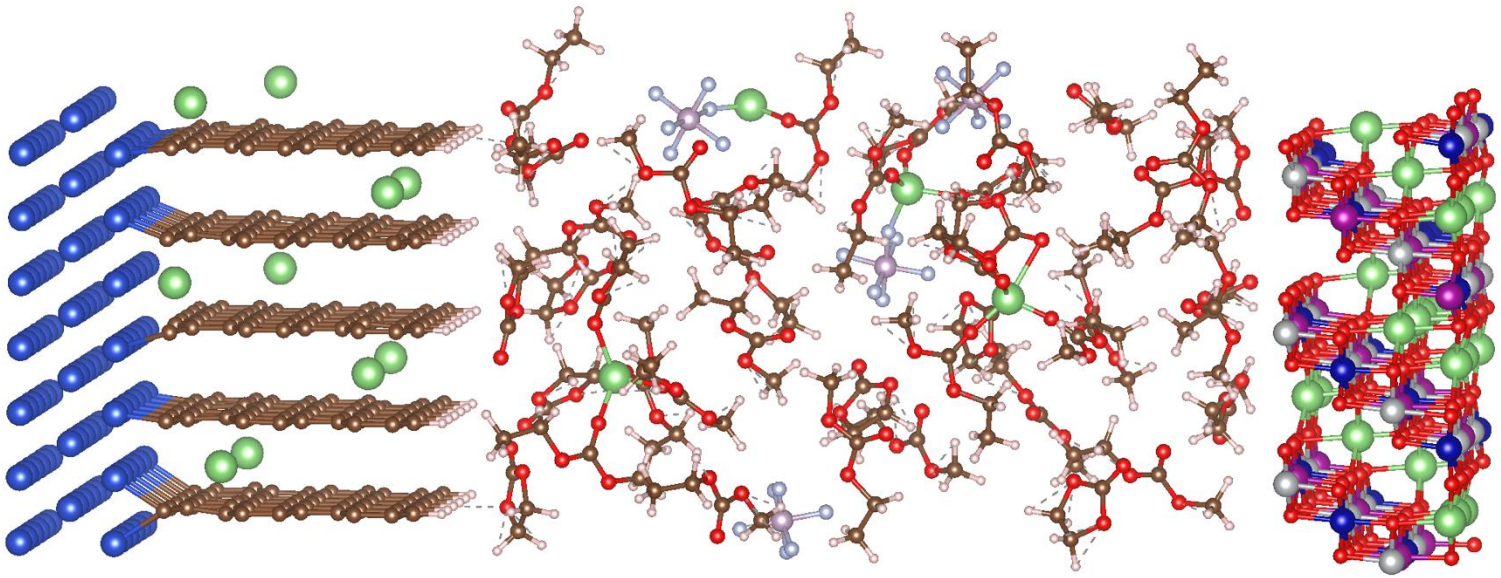


Full Battery 'baby' Model: can we Break the Simulation?

This is a difficult test for MACE-MP-O,

it is not a battery!

500K, NVT, MACE-MP-0+D3, 170ps



Cu | H-capped graphite+Li | EC/EMC+$LiPF_6$ | NMC+Li

Simulating the Li-ion battery environments

- ~ 1800 atoms
- 9 chemical elements
- 4 different materials: metal Copper, Li-loaded graphite, EC/EMC + LiPF6, Li-loaded (partial) NMC
- 4 different chemical interfaces

all local atomic environments are represented to some extend in MPtraj

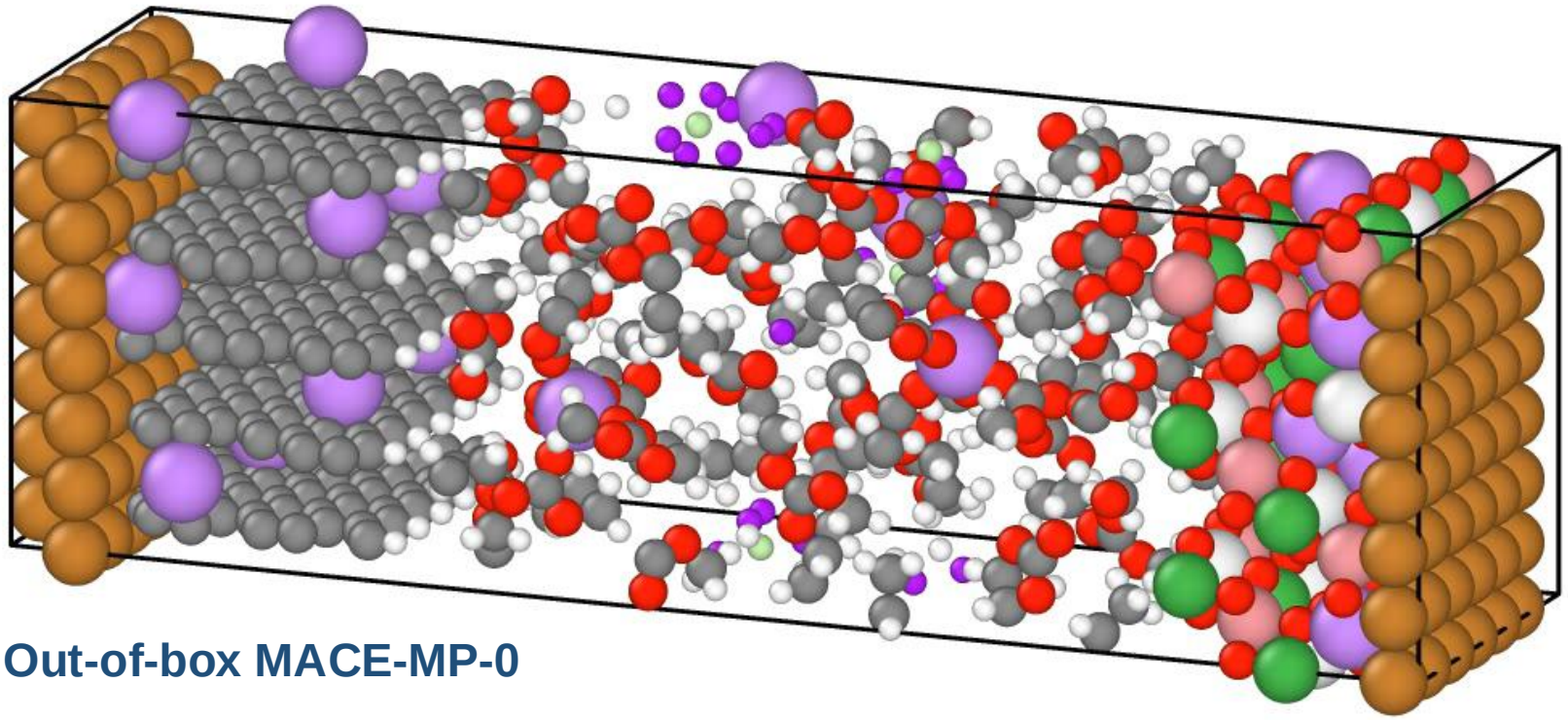
Will this simulation be stable with the out-of-the-box model?

Full Battery 'baby' Model

MLIP performance:

~1000s atoms

~100ps a day/GPU



Out-of-box MACE-MP-0

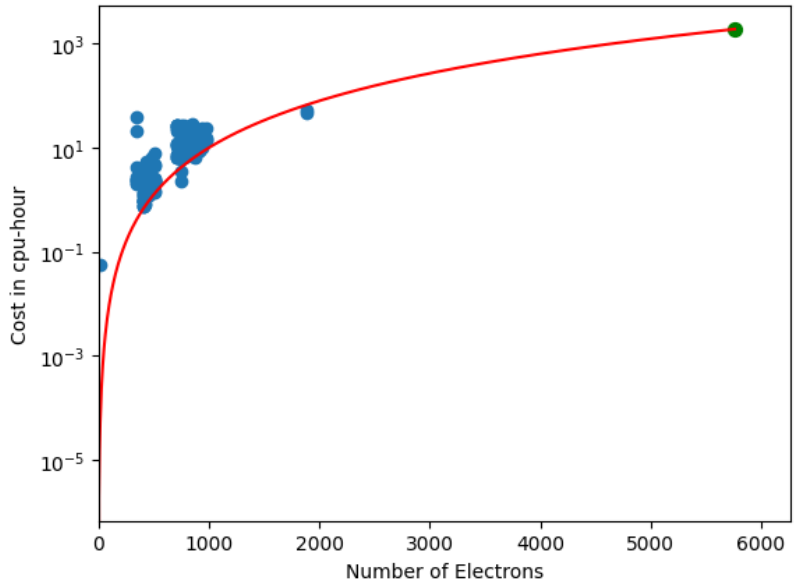
1800 atoms, simulation is stable at 500K, NVT, 170ps

DFT cost/performance

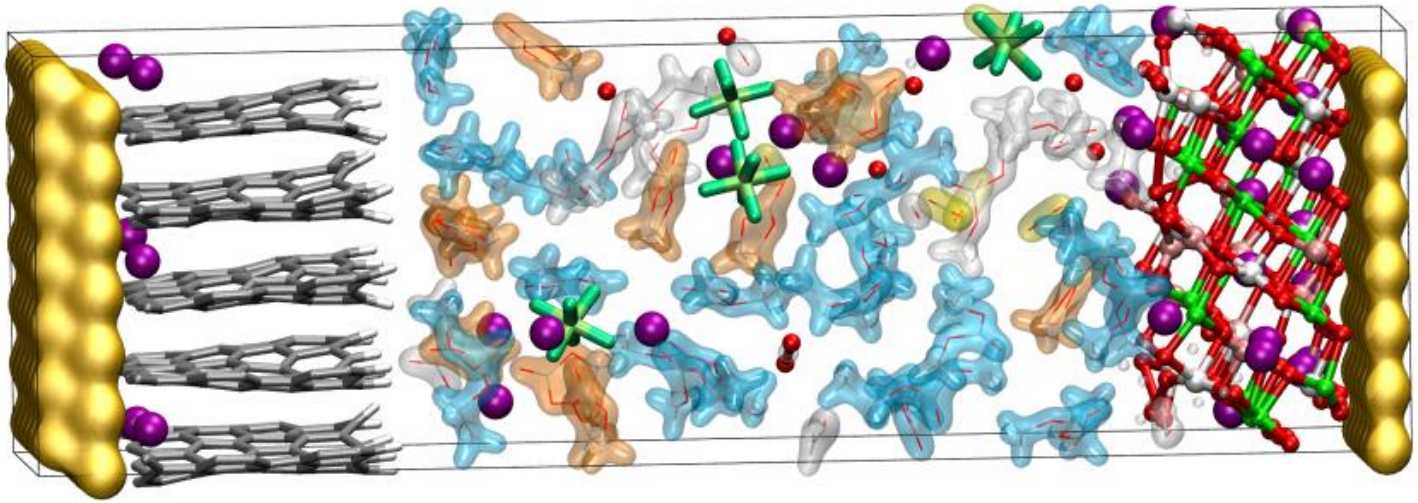
(rough estimate, linear-scaling DFT can certainly do much better)

this system: ~6000 electrons

~single point 2 day / 20 CPUs

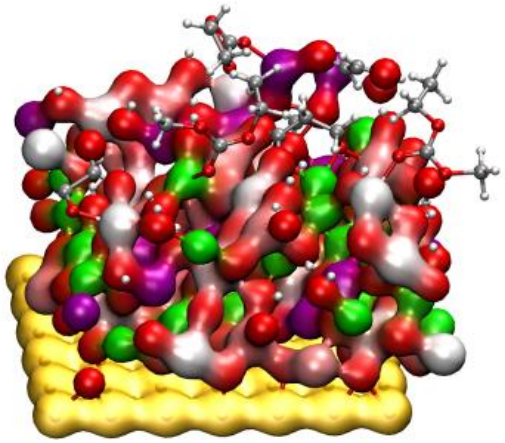


Beginning of SEI Formation? Is the Science Right?



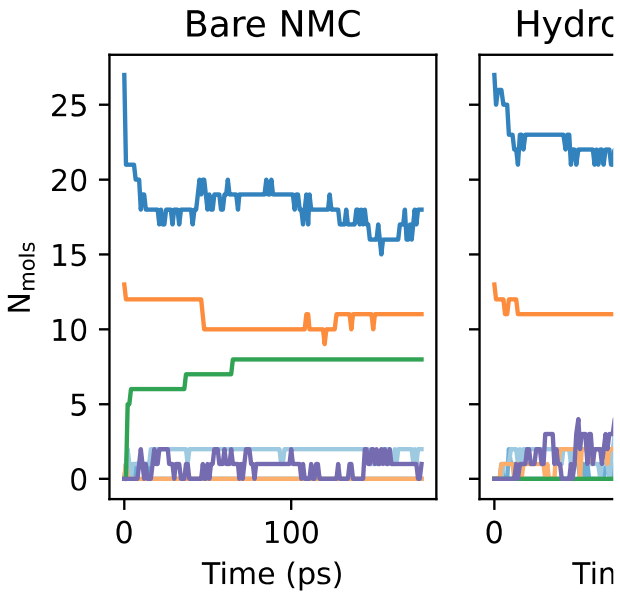
New Compounds in the Electrolyte / Interface

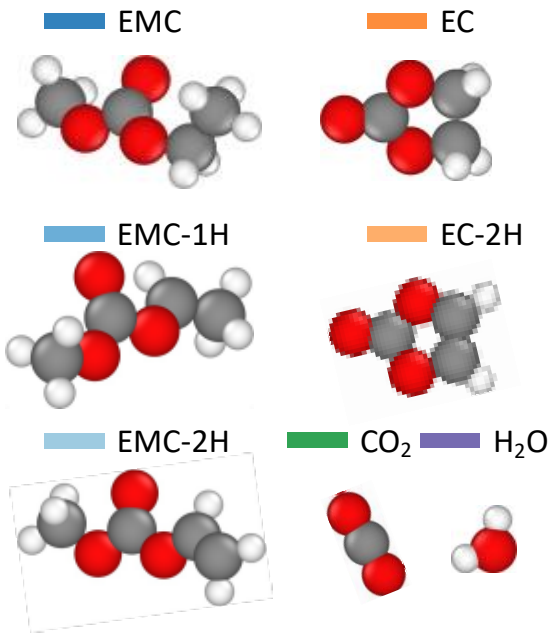
- bare NMC (no H): CO_2 forming, some EC, EMC become radical
- hydrogenated NMC (surface O atoms): less CO_2, more H_2O forming, mostly EMC become radical



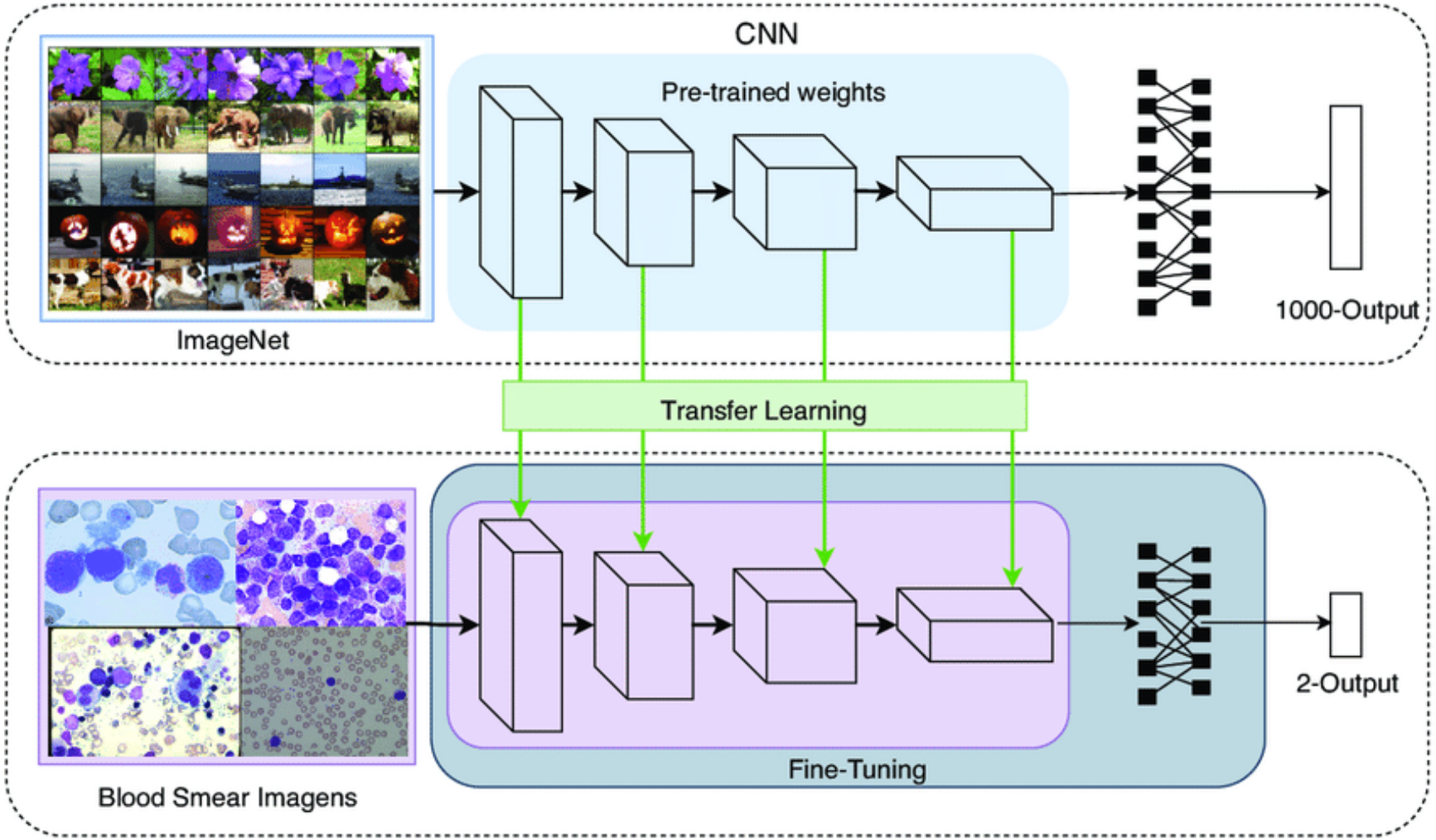
Final MD snapshot

- gray: products of reacted solvent molecules
- red: oxygen atoms originating from cathode, forming new CO_2 and H_2O molecules
- solvent molecules chemisorbed on cathode surface



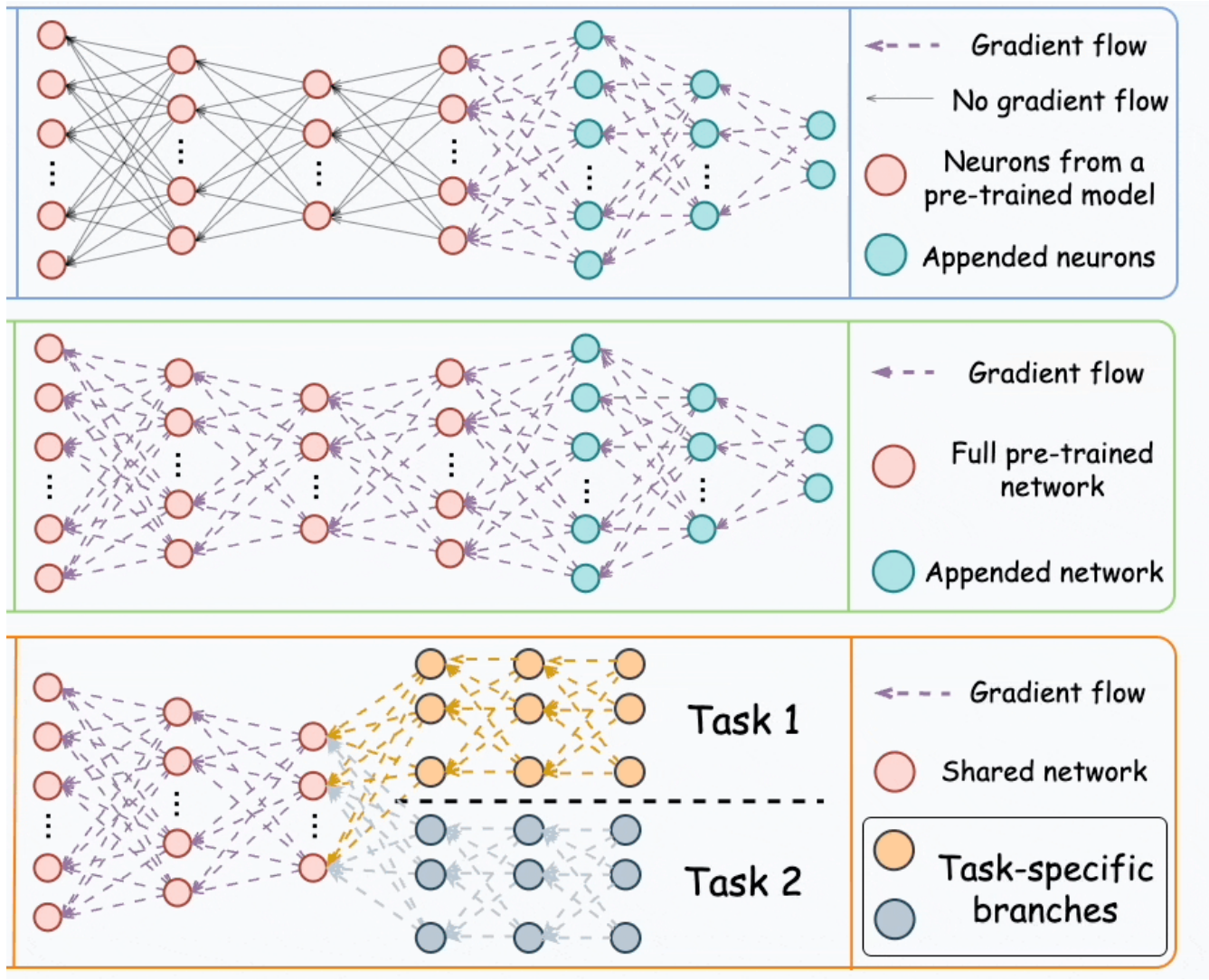


Fine Tuning: Pretrain on Big Data, Transfer to Small Data



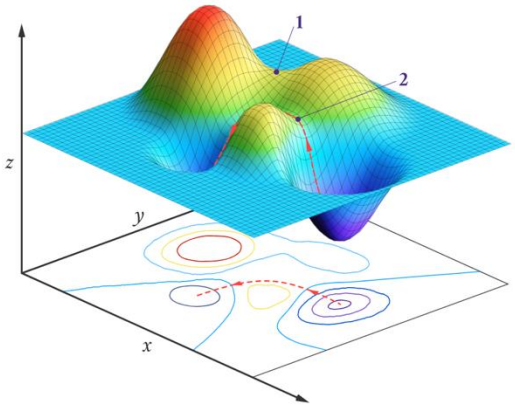
- working assumption: parts of the model are transferable (feature construction)
- pretrain model on available large dataset
- fine-tune on small dataset for specific application

Fine Tuning Strategies



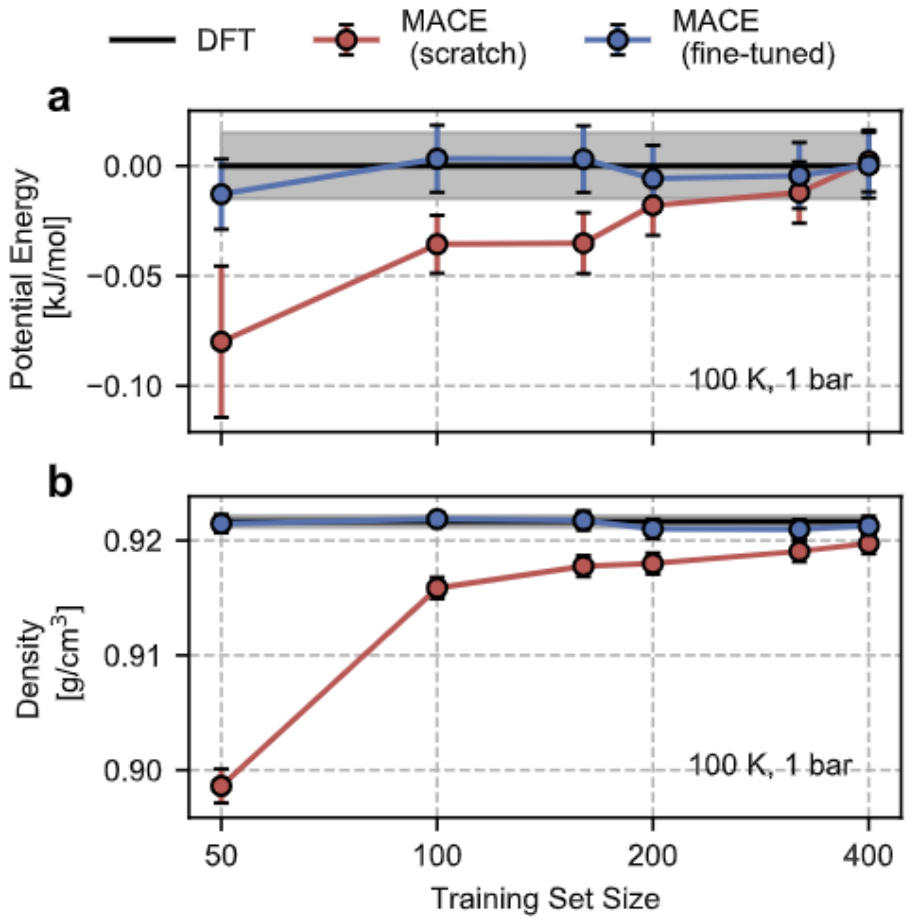
- freeze parts of the network (same features)
- unfreeze all network: start solution is closer to minimum on Loss Landscape
- multi-head: one head is the old data (keeps weights constrained), new head is new data

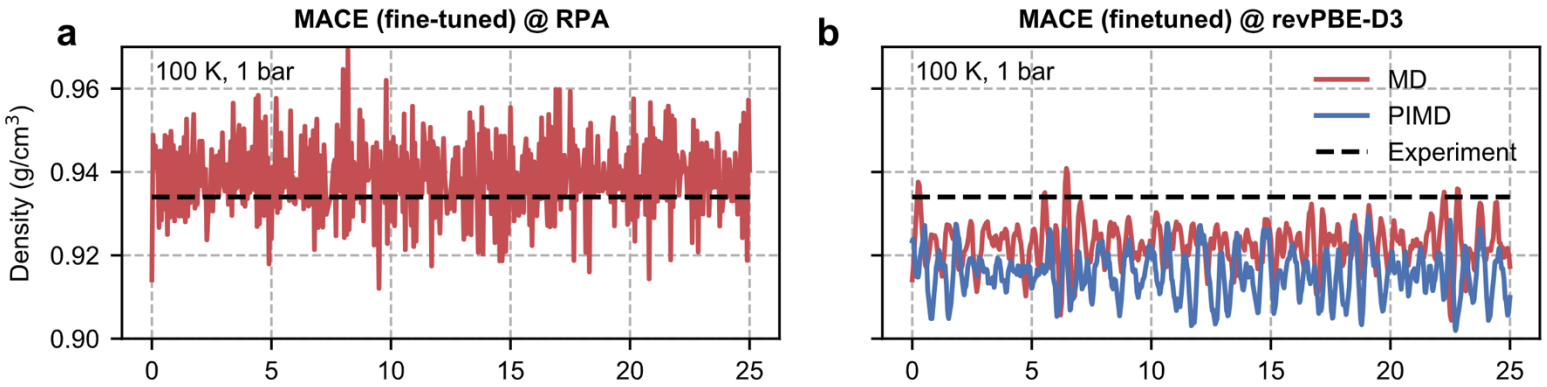
Loss Landscape



Fine Tuning in MLIPs: Early Days

- out-of-train systems (here ice Ih), same level of theory (PBE)
- new level of theory
- both out-of-train system and new level of theory (RPA, revPBE-D3)





Harveen Kaur, et al, arXiv preprint arXiv:2405.20217 (2024)

https://github.com/venkatkapil24/fine-tuning-MLPs-ice-polymorphs