

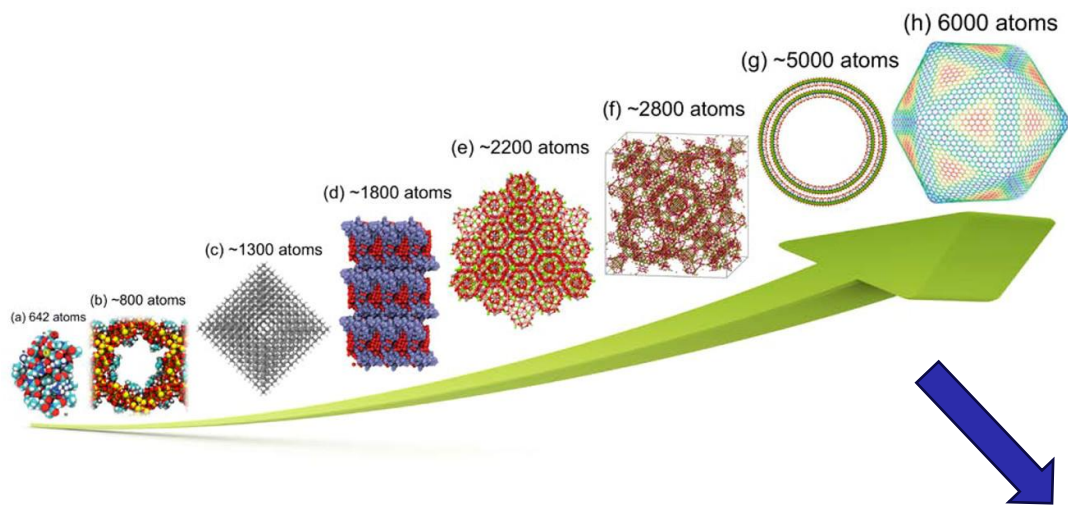
A Compact Tensor Regression Model for Accelerating First-Principle Molecular Dynamics Simulation

ISSP, Ozaki Group
D1, Li Hengyu

Outline

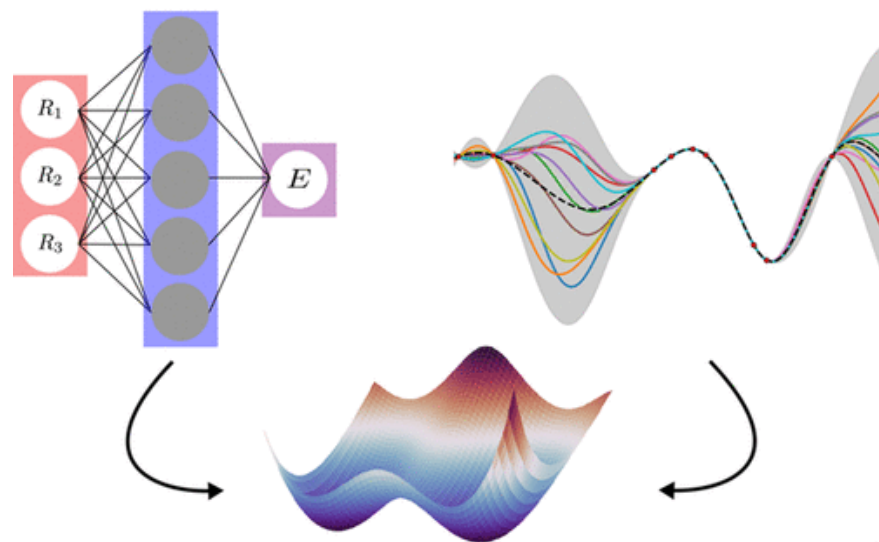
- Background
- Purpose of study
- Method
- Benchmark result
- Conclusion
- Future Prospects

Background



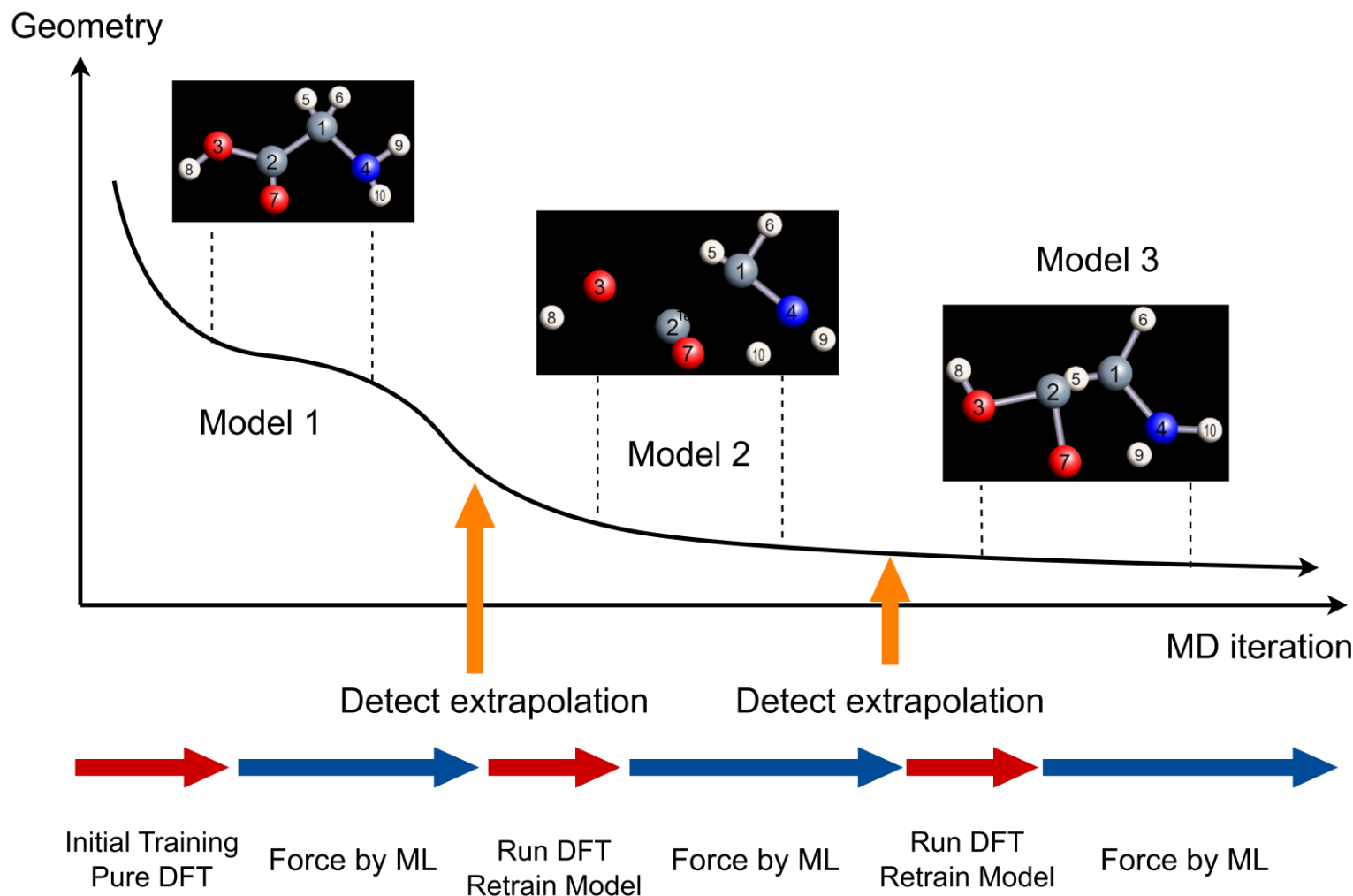
The desired system size for first-principles molecular dynamics (FPMD) simulation is largely increased.

Machine learning techniques has been proven to fit potential energy surface¹. For better performance, researchers seek to develop **more complex model**, like deep neural network.



¹ K. T. Butler et al., Nature 559, 547 (2018).

Overview of on-the-fly fitting (active learning)



Model is not fixed by pre-train, also updated along MD simulation.

Purpose of study

Main purpose

We try to fit the **local PES** for **each atom** with a **compact atomic decomposed model** and **on-the-fly update** along MD simulation.

Highlight features

- Fitting by solving linear equations problem (**numerical robustness** and **low retrain cost**)
- Independent model for each atom (**$O(N)$ time complexity** and **ability for complex chemical environment**)
- Training data can be iteratively accumulated (**Cold start and online training**)

Tensor representation of atomic energy

Total energy is represented by summation of atomic energy:

$$E_{\text{tot}}^{\text{model}} = \sum_i E_i^{\text{model}}$$

Then we consider the **atomic energy** for atom i :

$$E_i^{\text{model}} = \langle V_i | \left((A^{(n)} |V_i\rangle) |V_i\rangle \dots |V_i\rangle \right),$$

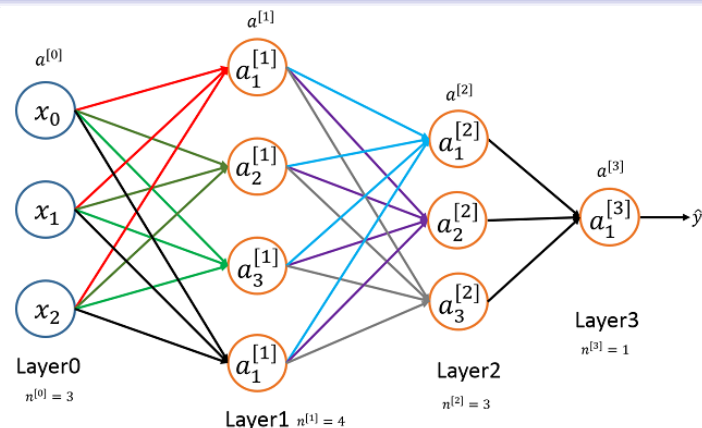
$n - 1$ tiems

$$A^{(n)} |V_i\rangle = \sum_j A_{i,j,k\dots,n}^{(n)} v_j = A^{(n-1)},$$

Where $A^{(n)}$ is a **n-rank tensor**, $|V_i\rangle$ is descriptor vector. For 1st and 2nd rank case:

$$E_i(A^{(1)}) = A_1 v_1 + A_2 v_2 + \dots + A_m v_m,$$

$$E_i(A^{(2)}) = E_i(A^{(1)}) + A_{22} v_1 v_1 + A_{23} v_1 v_2 + \dots$$



$$E = \sum_i \omega_i \sum_j \omega_j f_j \left(\sum_k f_k(\omega_k x_k) \right),$$

With **Taylor expansion** of f :

$$f_k(\omega_k x_k) = f(x_0) + f'(x_0)(\omega_k x_k - x_0) + \frac{1}{2!} f''(x_0) (\omega_k x_k - x_0)^2 + \dots$$

So, we have:

$$E = b_0 + \sum_i b_i x_i + \sum_{i,j} b_{ij} x_i x_j + \dots + \sum_{i,j,\dots,n} b_{ij\dots n} x_i x_j \dots x_n$$



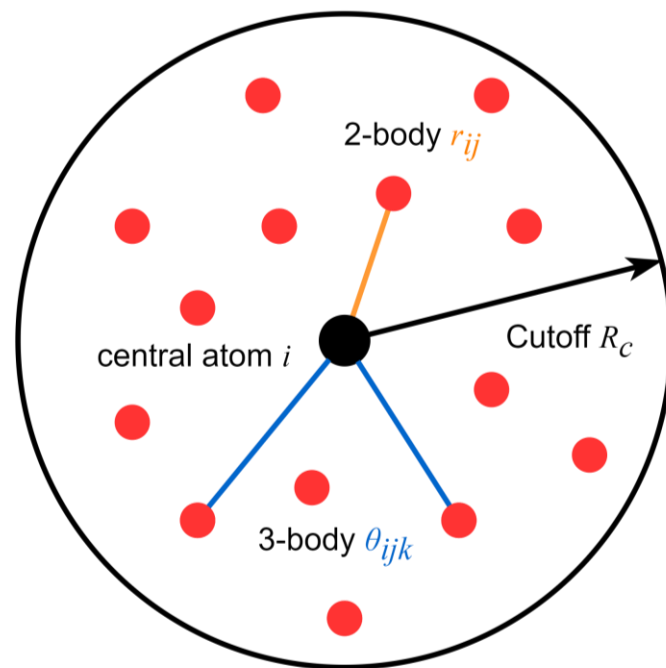
Descriptor vector

We consider the atomic energy is contributed by **2-body** and **3-body interactions** in a finite range.

$$|V\rangle = (1, V_1^{rad}, \dots, V_p^{rad}, V_1^{ang}, \dots, V_p^{ang})$$

$$V_p^{rad} = \sum_{j, j \in \text{neighbor}} h(Z_j) G_p^{rad}(r_{ij})$$

$$V_p^{ang} = \sum_{j, k; j, k \in \text{neighbor}} h(Z_j, Z_k) G_p^{ang}(\theta_{ijk})$$

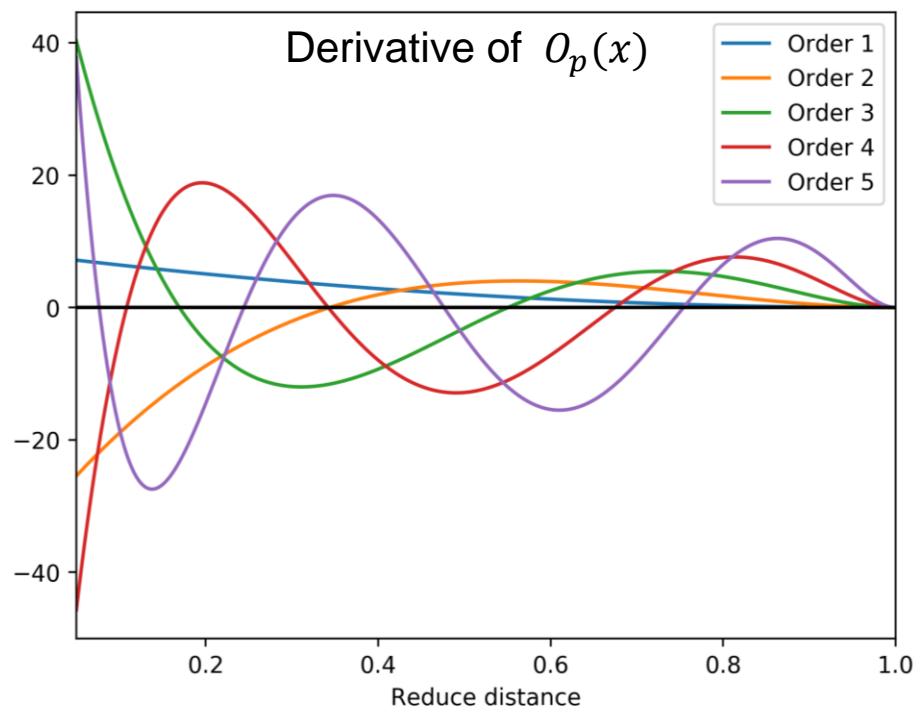
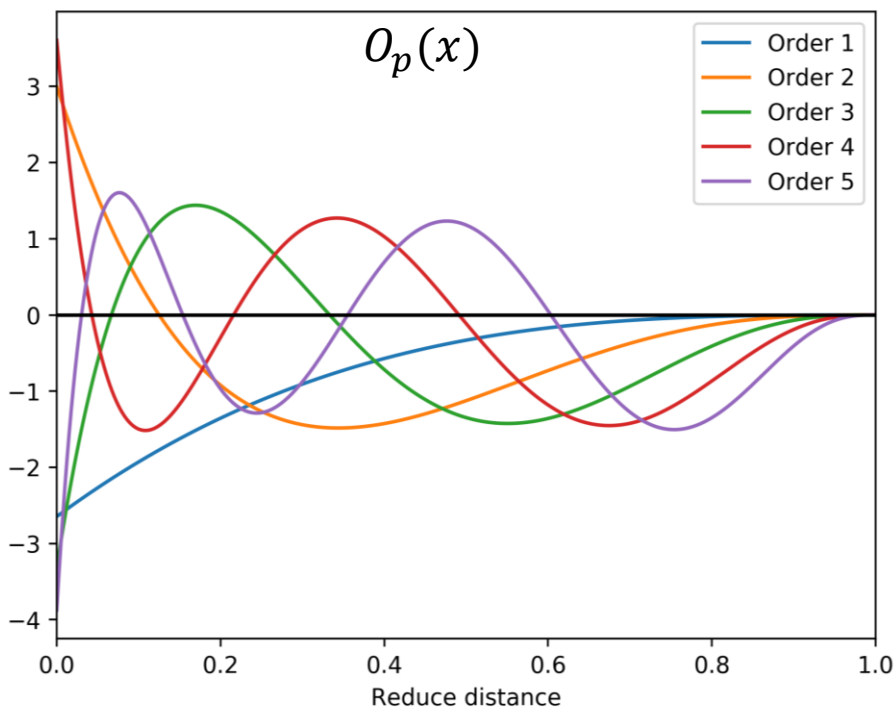


We use series of symmetry function G_p to **sample all many body pairs inside cutoff range**. Z_j is the normalized atomic number for distinguish **element type**.

Orthogonal polynomial descriptor

Orthogonal polynomials with $(x - 1)^p$ basis

The $O_p(x) = \text{Orth}[(x - 1)^p]$, $p \geq 3$ series polynomial is forced to be orthogonal.



O_p , O'_p , O''_p should converge to 0 for **smooth energy change respect to neighboring atom change.**

Loss function

The fitting process is to determine the tensor element in A_i^n by minimizing the loss function:

$$L = \sum_N (E_N^{\text{model}} - E_N^{\text{DFT}})^2 + \sum_N \sum_{\text{axis}} (F_{N,\text{axis}}^{\text{model}} - F_{N,\text{axis}}^{\text{DFT}})^2 + \lambda_2 \sum_N \|A_i^n\|$$

Thus, to minimize the loss function we can let derivative to 0. (2nd rank tensor case):

$$\frac{\partial L}{\partial A_{m'n'}} = \sum_N (E_N^{\text{model}} - E_N^{\text{DFT}}) \frac{\partial E_N^{\text{model}(i)}}{\partial a_{i,j,p}} + \sum_N \sum_{\text{axis}} (F_{N,\text{axis}}^{\text{model}} - F_{N,\text{axis}}^{\text{DFT}}) \frac{\partial F_{N,\text{axis}}^{\text{model}(i)}}{\partial A_{m'n'}} + 2\lambda_2 \sum_N A_{m'n'}$$

Then the fitting process could be done alone the MD simulation and solving the **linear equations problem**:

$$\left[\frac{\partial L}{\partial A_{m'n'}} \right] [A_{m'n'}]^T = 0$$

Implementation

Decouple atomic force

$$F_{i,axis}^{\text{model}} = \frac{\partial E_{tot}^{\text{model}}}{\partial R_{i,axis}} = \frac{\partial E_i^{\text{model}}}{\partial R_{i,axis}} + \sum_{j \neq i} \frac{\partial E_j^{\text{model}}}{\partial R_{i,axis}} = F_{\text{self}}(A_{i,N}) + F_{\text{neighbor}}(A_{j,N})$$

$F_{\text{neighbor}}(A_{j,N})$ contains parameters from neighboring atom which **can not solved independently**. So, we use the parameters $A_{j,N-1}$ to approximate $F_{\text{neighbor}}(A_{j,N})$ and use **autoregressive model to minimize residuals**.

$$F_{\text{diff}}(A_{j,N}, A_{j,N-1}) = \alpha_1 F_{\text{diff}}(A_{j,N-1}, A_{j,N-2}) + \alpha_2 F_{\text{diff}}(A_{j,N-2}, A_{j,N-3}) + \dots$$

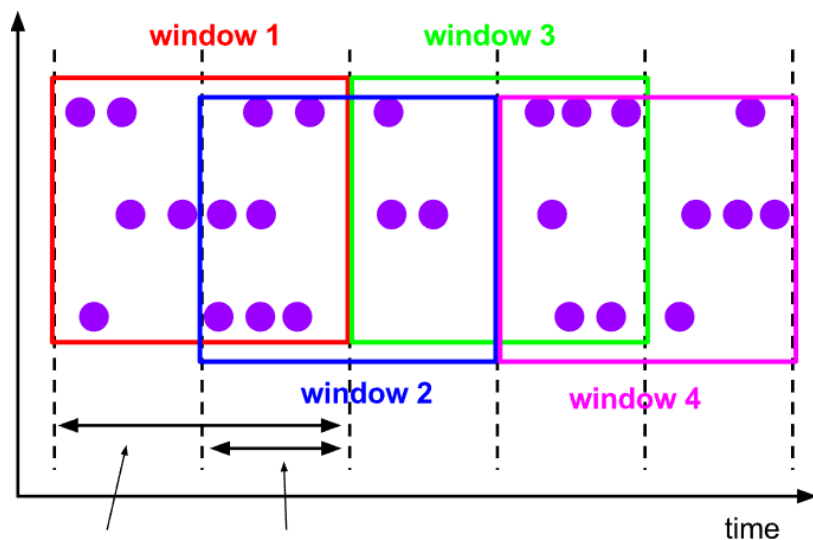
Implementation

Online training

Cause the dimension of parameter tensor A_i^n is fixed, thus we can **iteratively accumulate** the training data by:

$$\frac{\partial L}{\partial A_{m'n'}} = A_{m'n'}^n = \omega * A_{m'n'}^{n-1} + A_{m'n'}^{\text{new}}$$

ω is a **forgetting ratio** which is little smaller than 1 to **discard history step information** that is too far away from current configuration.



Force residuals given by autoregressive model reflect **long-time scale interaction** in MD trajectory. The idea is same in **recurrent neural networks (RNN)** and **graph neural networks (GNN)**.

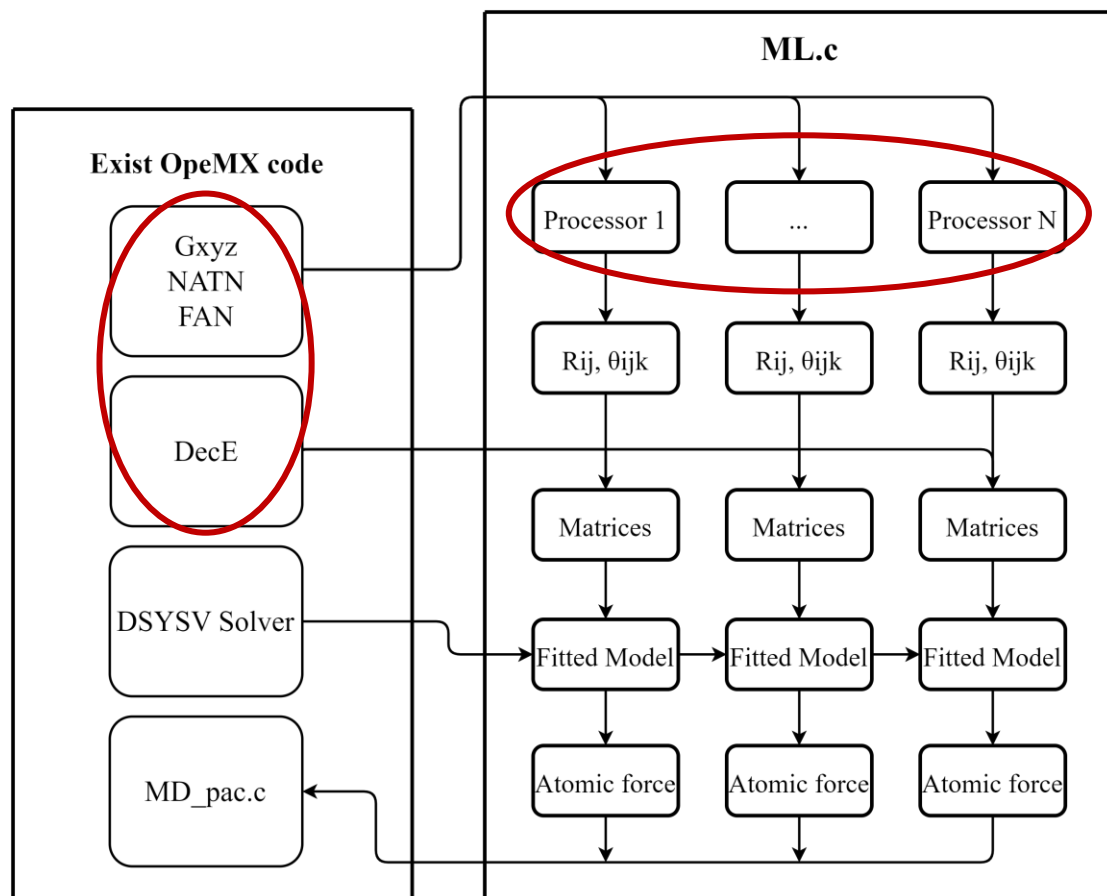
Workflow with OpenMX

Scheme of ML code and OpenMX¹

Fully parallelized for each atom

Necessary data from OpenMX

- **Gxyz**
atom position
- **FNAN**
neighboring atoms number
- **NATN**
neighboring atoms serial number
- **DecE**
decomposed energy



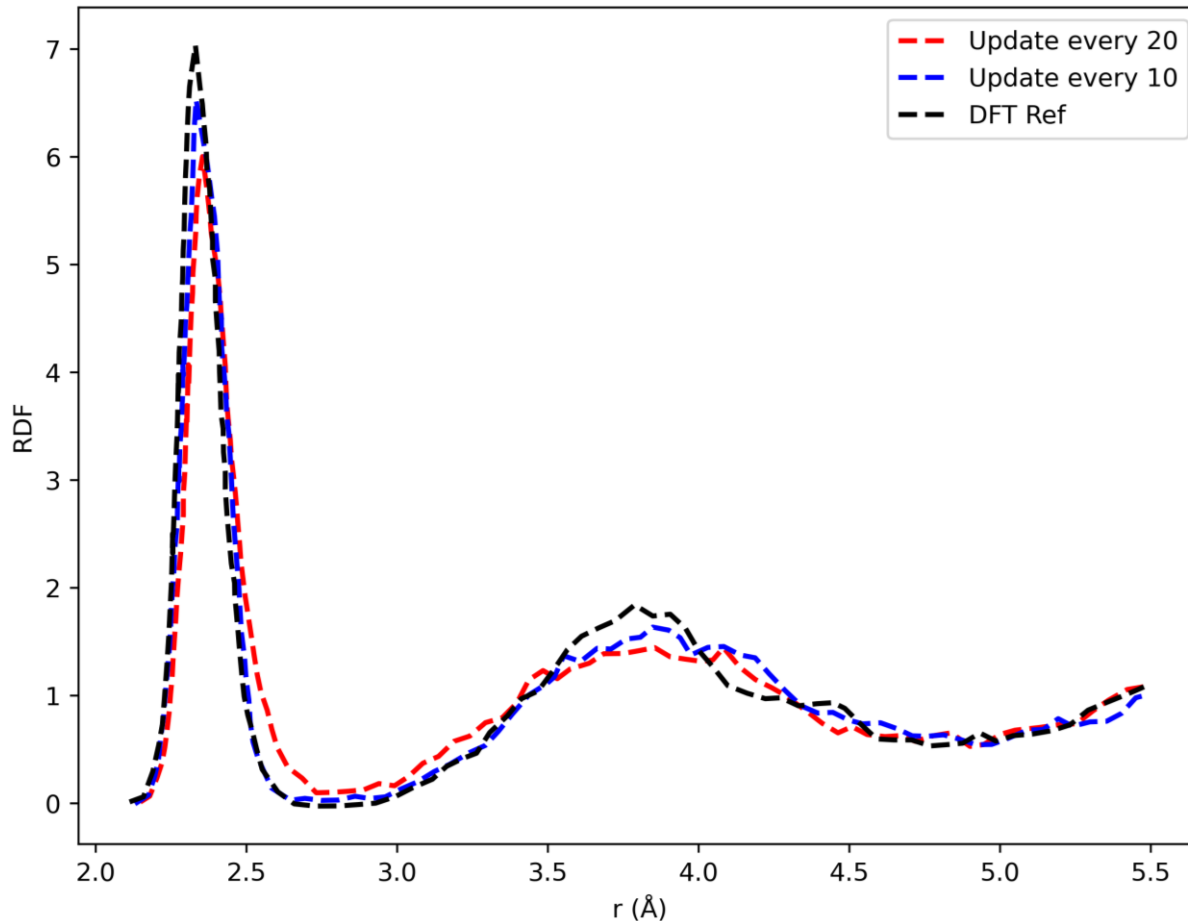
[1] OpenMX: <https://www.openmx-square.org/>

Benchmark calculation of various systems

Isolated System	Energy MAE	Force MAE
Methene (CH ₄)	0.08 meV/atom	7 meV/Å
Nitrobenzene (C ₆ H ₅ NO ₂)	0.73 meV/atom	68 meV/Å
Sialic Acid (C ₁₁ H ₁₉ NO ₉)	1.07 meV/atom	114 meV/Å
Adenosine Triphosphate (C ₁₀ H ₁₆ N ₅ O ₁₃ P ₃)	1.32 meV/atom	145 meV/Å
Bulk System	Energy MAE	Force MAE
2D Graphene	0.23 meV/atom	27 meV/Å
Aluminum	0.47 meV/atom	76 meV/Å
Amorphous Silica (SiO ₂)	0.92 meV/atom	127 meV/Å

The typical accuracy range for ML potential is **~2 meV/atom and ~200 meV/Å**.

Radial Distribution Function

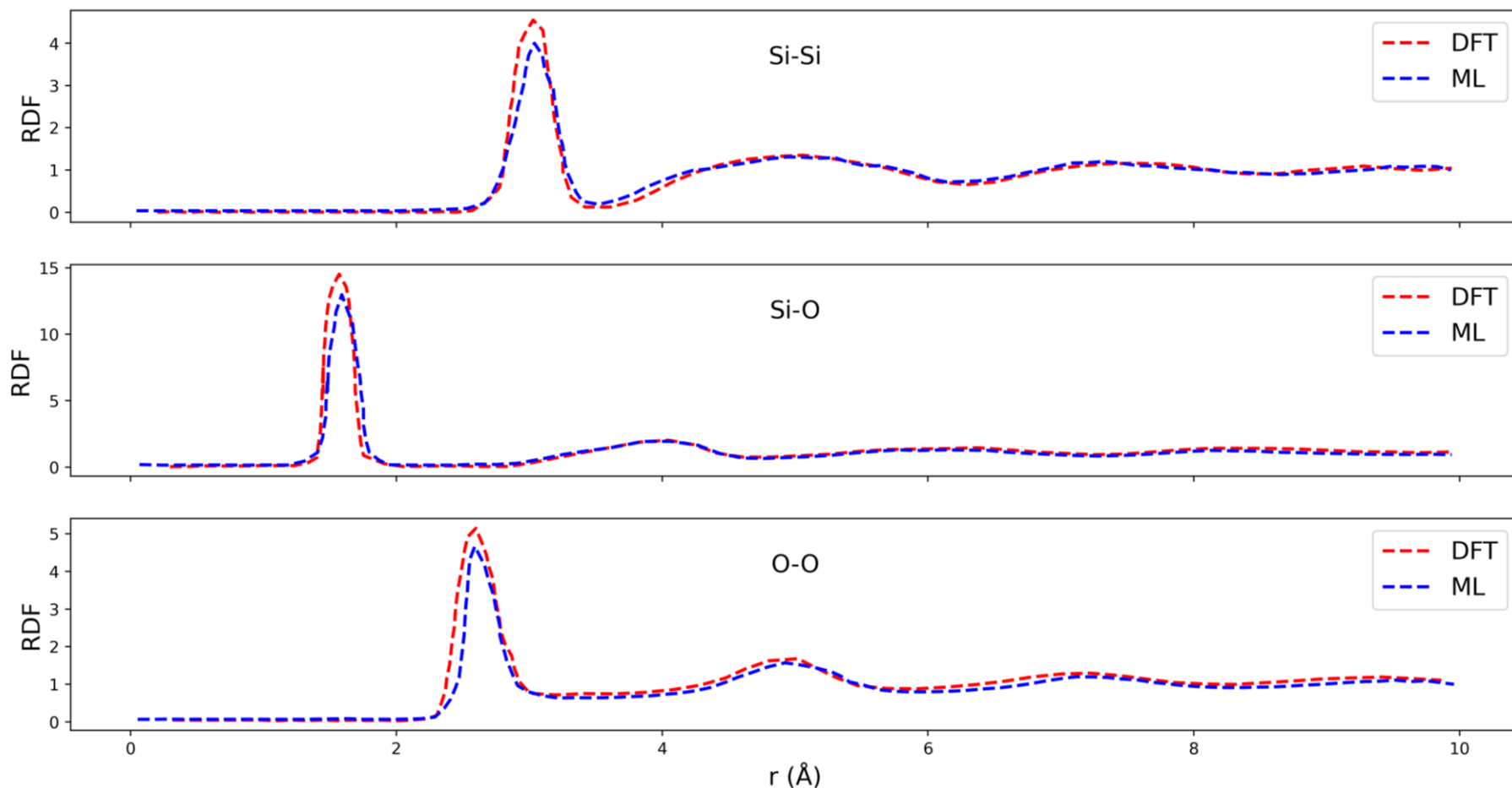


Two peaks are precisely reproduced by our tensor regression model.

RDF of **amorphous silicon** Si-Si distance under 300K.

(2000 fs in total, 400 fs for initial training, update every 10 steps).

Radial Distribution Function



RDF of **amorphous silica** under 300K.

(2000 fs in total, 500 fs for initial training, update every 10 steps).

Conclusion

- We developed **effective model of fitting local atomic energy surface**, the **accuracy is comparable to other ML potentials**. Also, it able to reproduce the **long-time scale statistical property**.
- The accelerate ratio is around 10 by **simple on-the-fly algorithm** but lower than existing active learning method ($\sim 10^3$).
- Our model shows enlightenment in the development of future machine learning potentials. **Compact models which based on the physical system also possible for ML potentials compare with the more complex and data-driven large models.**

Future Prospects

- Design **on-the-fly algorithm** based on our model, for example the **analysis of changes in atomic chemical environment**.
- Application for **electrode surface**, because there are many possible surface structures and a long AIMD time is required for screening.
- Extending the model prediction capabilities to more complex **high-entropy alloy surfaces** that always face **reconfiguration** during dynamics.
- Our final target system is **electrochemical interface systems** which are still challenge for existing ML potentials and DFT packages .

Thank you for your listening.

Density functional theory

Many-body Schrödinger equation

$$\hat{H}\Psi(\mathbf{r}, \mathbf{R}) = E\Psi(\mathbf{r}, \mathbf{R})$$

With Born-Oppenheimer approximation, the electronic motion and the nuclear motion in molecules can be **separated**.

$$\Psi(\mathbf{r}, \mathbf{R}) = \phi_{ele}(\mathbf{r})\chi_{nuc}(\mathbf{R})$$

Electron Schrödinger equation in Hartree unit

$$\left[-\frac{1}{2} \sum_i \nabla_i^2 + \frac{1}{2} \sum_{i \neq j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{i,I} \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \right] \phi_{ele}(\mathbf{r}) = E \phi_{ele}(\mathbf{r})$$

Density functional theory

Hohenberg-Kohn theorem I

For any system of interacting particles in an external potential $V_{ext}(\mathbf{r})$, the potential $V_{ext}(\mathbf{r})$ is determined uniquely by the ground-state particle density $n_0(\mathbf{r})$.

Hohenberg-Kohn theorem II

A *universal functional* for the energy $E[n]$ can be defined in terms of the density. The exact ground state is the global minimum value of this functional.

With Hohenberg-Kohn theorem, the solution of the system ground state can be shifted from the solution of the wave function to the electron density. The system degrees of freedom are **reduced from $3N$ to 3**.

Density functional theory

Kohn-Sham DFT

In Kohn-Sham scheme constructed an auxiliary system which the atoms are all independent of each other, and in turn we can solve the **single-electron Schrödinger equation**.

$$E[n] = -\frac{\nabla^2}{2} + \int d\mathbf{r} n(\mathbf{r})V_{ext}(\mathbf{r}) + \frac{1}{2} \int d\mathbf{r}d\mathbf{r}' \frac{n(\mathbf{r})n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + E_{xc}(n)$$

With Hohenberg-Kohn theorem II, the electron density that minimizes $E[n]$ is the real electron density of the system. By the Lagrange multiplier method we can obtain the KS equation.

$$\left(-\frac{\nabla^2}{2} + V_{ext}(\mathbf{r}) + V_H(\mathbf{r}) + \frac{\delta E_{xc}[n]}{\delta n} \right) \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r})$$

After determining the form of $E_{xc}[n]$, KS equation can be solved by self-consistent field method.

Linear scaling DFT

The core idea of the $O(N)$ method is based on the assumption that the electron density n is given by the summation over each site i .

$$n^\sigma(\mathbf{r}) = \sum_{i\alpha, j\beta} \chi_{i\alpha}(\mathbf{r}) \chi_{j\beta}(\mathbf{r}) \rho_{i\alpha, j\beta}^\sigma = \sum_i \left(\sum_{\alpha, j\beta} \chi_{i\alpha}(\mathbf{r}) \chi_{j\beta}(\mathbf{r}) \rho_{i\alpha, j\beta}^\sigma \right) = \sum_i n_i^\sigma(\mathbf{r}),$$

The density matrix ρ is calculated by solving following linear equations:

$$H^{(i)} c_\mu^{(i)} = \epsilon_\mu^{(i)} S^{(i)} c_\mu^{(i)}.$$

where, $H^{(i)}$ is the local Hamiltonian matrix, $S^{(i)}$ is the overlap matrix. By truncating system the time complexity is reduced:

$$O(N^3) \rightarrow N \times O(N_{constant})$$

Further, the $N_{constant}$ can be reduced by introducing Krylov subspace method, The Krylov subspace method used in OpenMX is a combination of DC and recursive methods to improve computational efficiency and robustness.

Energy decomposition in OpenMX

Because the pseudo-atomic orbitals χ for pseudopotential in OpenMX is in finite range, the total energy can be decomposed into contributions from each atomic site i and localized orbital α :

$$\begin{aligned} E_{\text{tot}} &= \sum_{\sigma=\uparrow,\downarrow} \sum_{\nu} f_{\nu} \langle \phi_{\nu}^{\sigma} | \hat{T} | \phi_{\nu}^{\sigma} \rangle + \sum_{\sigma\sigma'} \int \omega_{\sigma\sigma'} n_{\sigma'\sigma} + \frac{1}{2} \iint \frac{n'(r)n'(r')}{|r-r'|} dv dv' \\ &+ E_{\text{xc}}\{n_{\sigma\sigma'}\} + E_{\text{cc}} \\ &= E_{\text{band}} - \frac{1}{2} \int n' V_{\text{H}} dv - \int \text{Tr}(V_{\text{xc}} n) dv + E_{\text{xc}} + E_{\text{cc}} \\ &= \sum_{i\alpha} \left[\sum_{j\beta,\sigma\sigma'} \rho_{i\alpha,j\beta}^{\sigma\sigma'} F_{j\beta,i\alpha}^{\sigma'\sigma} \right] + \sum_{i\alpha j\beta} E_{\text{cc}}^{(i\alpha j\beta)} \\ &= \sum_{i\alpha} E_{i\alpha}, \end{aligned}$$

Energy decomposition in OpenMX

The total energy in OpenMX is organized by:

$$E_{\text{tot}} = E_{\text{kin}} + E_{\text{na}} + E_{\text{NL}} + E_{\text{ec}} + E_{\delta\text{ee}} + E_{\text{xc}} + E_{\text{scc}}.$$

Kinetic energy

$$\begin{aligned} E_{\text{kin}} &= \sum_{\sigma} \sum_{i\alpha} \left(\sum_{j\beta} \sum_n^N \rho_{\sigma,i\alpha,j\beta}^{(\mathbf{R}_n)} h_{i\alpha,j\beta,\text{kin}}^{(\mathbf{R}_n)} \right) \\ &= \sum_{\sigma} \sum_{i\alpha} E_{\sigma,i\alpha,\text{kin}} \end{aligned}$$

with

$$h_{i\alpha,j\beta,\text{kin}}^{(\mathbf{R}_n)} = \frac{1}{V_B} \int_{BZ} dk^3 \sum_{\mu}^{\text{Occ}} \langle \psi_{\sigma\mu}^{(\mathbf{k})} | \hat{T} | \psi_{\sigma\mu}^{(\mathbf{k})} \rangle$$

Energy decomposition in OpenMX

Electron-core Coulomb energy (Non-local part)

$$\begin{aligned} E_{ec}^{(NL)} &= \sum_{\sigma} \sum_{i\alpha} \left(\sum_n \sum_{j\beta} \rho_{\sigma i\alpha j\beta}^{(\mathbf{R}_n)} h_{i\alpha, j\beta, NL}^{(\mathbf{R}_n)} \right) \\ &= \sum_{\sigma} \sum_{i\alpha} \left(\sum_n \sum_{j\beta} \rho_{\sigma i\alpha j\beta}^{(\mathbf{R}_n)} \langle \phi_{i\alpha} | \sum_I V_{NL, I} | \phi_{j\beta} \rangle \right) \\ &= \sum_{\sigma} \sum_{i\alpha} E_{\sigma i\alpha}^{(NL)} \end{aligned}$$

Neutral atom energy

$$\begin{aligned} E_{na} &= \int dr n(r) V_{na, I} \\ &= \sum_{\sigma} \sum_{i\alpha} \left(\sum_{j\beta} \sum_n \rho_{\sigma, i\alpha, j\beta}^{(\mathbf{R}_n)} h_{i\alpha, j\beta, na}^{(\mathbf{R}_n)} \right) \\ &= \sum_{\sigma} \sum_{i\alpha} E_{\sigma, i\alpha}^{na} \end{aligned}$$

Energy decomposition in OpenMX

Screened core correction energy

$$\begin{aligned} E_{sec} &= \frac{1}{2} \sum_{I,J} \left(\frac{Z_I Z_J}{|\tau_I - \tau_J|} - \int dr n_I^{(a)}(r) V_{H,J}^{(a)}(r) \right) \\ &= \sum_{\sigma} \sum_{i\alpha} \left(\frac{1}{2N_i} \sum_j \frac{Z_I Z_J}{|\tau_I - \tau_J|} \right. \\ &\quad \left. - \int dr n_I^{(a)}(r) V_{H,J}^{(a)}(r) \right) \\ &= \sum_{\sigma} \sum_{i\alpha} E_{\sigma,i\alpha}^{sec} \end{aligned}$$

Electron-electron Coulomb energy

$$\begin{aligned} E_{\delta ee} &= \frac{1}{2} \int dr (n(r) - \sum_I n_I^{(a)}(r)) \delta V_H(r) \\ &= \sum_{\sigma} \sum_{i\alpha} \frac{1}{2} \left(\sum_{j\beta} \sum_n^N \rho_{\sigma,i\alpha,j\beta}^{(\mathbf{R}_n)} h_{i\alpha,j\beta}^{\delta V} \right. \\ &\quad \left. - \frac{1}{2} \int dr \frac{n_i^{(a)}(r)}{N_i} \delta V_H(r) \right) \\ &= \sum_{\sigma} \sum_{i\alpha} E_{\sigma,i\alpha}^{\delta ee} \end{aligned}$$

Energy decomposition in OpenMX

Exchange correlation energy

$$\begin{aligned} E_{xc} &= \int dr (n(r) + n_{pcc}(r)) \epsilon_{xc}(r) \\ &= \sum_{\sigma} \sum_{i\alpha} \left(\sum_n^N \sum_{j\beta} \rho_{\sigma,i\alpha,j\beta,xc}^{(\mathbf{R}_n)} h_{i\alpha,j\beta,xc}^{(\mathbf{R}_n)} \right. \\ &\quad \left. + \frac{1}{2N_i} \int dr n_{pcc,i}(r) \epsilon_{xc}(r) \right) \\ &= \sum_{\sigma} \sum_{i\alpha} E_{\sigma,i\alpha}^{xc} \end{aligned}$$