Center for the Exascale Simulation of Material Interfaces in Extreme Environments

Massachusetts Institute of Technology

Cuong Nguyen: Performance and Accuracy of Machine Learning Potentials

Dionysios Sema: E(3)-Equivariant ML-DFT

Spencer Wyant: Developing Machine-Learning Interatomic Potentials with *julia*

Performance and Accuracy of Machine Learning Potentials

Dionysios Sema, Yeongsu Cho, Ngoc Nguyen, Youssef Marzouk, Nicolas Hadjiconstantinou, Heather Kulik

Massachusetts Institute of Technology

Introduction

We introduce the proper orthogonal descriptors for efficient and accurate interatomic potentials of complex chemical systems [1, 2, 3]. We compose the proper orthogonal descriptors to develop two interatomic potentials by expressing the per-atom energies as a linear and then as a linear and quadratic combination of proper orthogonal descriptors. We demonstrate the weak and strong scaling of these potentials and perform MD simulations to calculate material properties. We also perform MD simulations using Allegro [4] on complex systems to model the oxidation process of Hf and map the vapor-liquid dome of Al. Finally, we use POD and Allegro for accurate prediction of melting points and mechanical properties of our target systems.

E(3) Equivariant Deep NNs

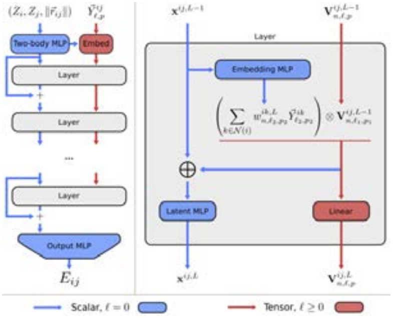

$(Z_i, Z_j, \|\vec{r}_{ij}\|)$
$\vec{Y}^{ij}_{\ell,p}$
Two-body MLP
Embed
Layer
Layer
...
Layer
Output MLP
E_{ij}
$\mathbf{x}^{ij,L-1}$
$\mathbf{V}^{ij,L-1}_{n,\ell,p}$
Layer
Embedding MLP
$\left(\sum_{k\in\mathcal{N}(i)} w^{ik,L}_{n,\ell_2,p_2} \vec{Y}^{ik}_{\ell_2,p_2}\right) \otimes \mathbf{V}^{ij,L-1}_{n,\ell_1,p_1}$
$\oplus$
Latent MLP
Linear
$\mathbf{x}^{ij,L}$
$\mathbf{V}^{ij,L}_{n,\ell,p}$
Scalar, $\ell = 0$
Tensor, $\ell \geq 0$

Figure 1: Allegro architecture (left) and the component of each layer (right). Features in the invariant and equivariant latent spaces interact at each layer with tensor products.

The Allegro Network contains separate latent spaces for invariant features (scalar, rotation order $l = 0$) and equivariant features of rotation order $l > 0$. The 2-body interactions are encoded with Bessel basis functions and a polynomial envelope and ACE-like many body interactions are constructed in each layer with: $N_{order} = N_{layers} + 2$. The final output are pairwise energies, E_{ij}

Proper Orthogonal Descriptors

We introduce the following set of snapshots on $(r_{\min}, r_{\max})$:

$$\xi_\ell(r_{ij}, \boldsymbol{\eta}) = V^{(2)}(r_{ij}, \boldsymbol{\eta}, \boldsymbol{\mu}_\ell), \quad \ell = 1, \ldots, N_{\mathrm{s}} \tag{1}$$

and compute the covariance matrix

$$C_{ij} = \frac{1}{N_{\mathrm{s}}} \int_{r_{\min}}^{r_{\max}} \xi_i(x, \boldsymbol{\eta}) \xi_j(x, \boldsymbol{\eta}) dx, \quad 1 \leq i, j \leq N_{\mathrm{s}}.$$

We then solve the eigenvalue problem $\boldsymbol{C}\boldsymbol{a} = \lambda \boldsymbol{a}$ to obtain the orthogonal basis functions

$$U^{(2)}_m(r_{ij}, \boldsymbol{\eta}) = \sum_{\ell=1}^{N_{\mathrm{s}}} a_{\ell m}(\boldsymbol{\eta})\, \xi_\ell(r_{ij}, \boldsymbol{\eta}),$$
$$U^{(3)}_{mn}(r_{ij}, r_{ik}, \theta_{ijk}, \boldsymbol{\eta}) = U^{(2)}_m(r_{ij}, \boldsymbol{\eta}) U^{(2)}_m(r_{ik}, \boldsymbol{\eta}) \cos(n\theta_{ijk})$$

and compute the proper orthogonal descriptors

$$\begin{aligned} D^{(2)}_{im}(\boldsymbol{\eta}) &= \sum_j U^{(2)}_m(r_{ij}, \boldsymbol{\eta}), \\ D^{(3)}_{imn}(\boldsymbol{\eta}) &= \sum_j \sum_k U^{(3)}_{mn}(r_{ij}, r_{ik}, \theta_{ijk}, \boldsymbol{\eta}). \end{aligned} \tag{2}$$

The linear POD potential is defined as

$$E(\boldsymbol{\eta}) = c^{(1)} + \sum_{m=1}^{N^{(2)}_{\mathrm{d}}} c^{(2)}_m d^{(2)}_m(\boldsymbol{\eta}) + \sum_{n=1}^{N^{(3)}_{\mathrm{d}}} c^{(3)}_n d^{(3)}_n(\boldsymbol{\eta}) \tag{3}$$

with $d^{(q)}_m = \sum_{i=1}^N D^{(q)}_{im}$. The quadratic POD potential is

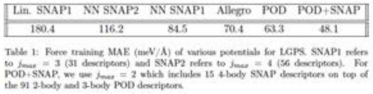
$$E = c^{(1)} + \sum_{m=1}^{N^{(2)}_{\mathrm{d}}} \left(c^{(2)}_m + b^{(2)}_m\right) d^{(2)}_m + \sum_{n=1}^{N^{(3)}_{\mathrm{d}}} \left(c^{(3)}_n + b^{(3)}_n\right) d^{(3)}_n \tag{4}$$

with $b^{(2)}_m = \sum_{n=1}^{N^{(3)}_{\mathrm{d}}} c^{(23)}_{mn} d^{(3)}_n$ and $b^{(3)}_n = \sum_{m=1}^{N^{(2)}_{\mathrm{d}}} c^{(23)}_{mn} d^{(2)}_m$. The coefficients $c^{(1)}, c^{(2)}_m, c^{(3)}_n, c^{(23)}_{mn}$, and $\boldsymbol{\eta}$ are found by solving a nonlinear least-squares regression against DFT data.
Extension to multi-element systems is carried out by computing the PODs in (2) for different atom types. The complexity of the resulting POD potentials is $O(NN_{\mathrm{n}}^2 N_{\mathrm{f}})$, where N_{n} is the number of neighbors and N_{f} is the number of basis functions. The complexity of the multi-element SNAP potential is $O(NN_{\mathrm{n}} N_{\mathrm{f}}^2 N_{\mathrm{e}}^2)$. Hence, the cost ratio between multi-element SNAP and POD potentials is $N_{\mathrm{f}} N_{\mathrm{e}}^2 / N_{\mathrm{n}}$.

Ongoing Research and Integration

We plan to develop NN/GCN potentials based on the proper orthogonal descriptors. We plan to equip POD potentials with UQ and active learning methods developed by our CESMIX team. We are also working on a Kokkos implementation to gain significant performance gains using multiple GPUs. As we aim to make the POD potentials available to the LAMMPS community, we would like to collaborate the DOE labs to implement POD models in LAMMPS and FitSNAP.

Lithium Ion Diffusivity



Lin. SNAP1	NN SNAP2	NN SNAP1	Allegro	POD	POD+SNAP
180.4	116.2	84.5	70.4	63.3	48.1

Table 1: Force training MAE (meV/Å) of various potentials for LGPS. SNAP1 refers to $j_{max} = 3$ (31 descriptors) and SNAP2 refers to $j_{max} = 4$ (56 descriptors). For POD+SNAP, we use $j_{max} = 2$ which includes 15 4-body SNAP descriptors on top of the 91 2-body and 3-body POD descriptors.

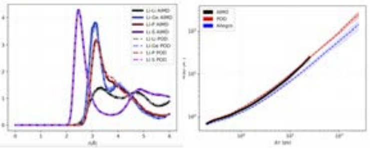


Figure 2: Simulation of Lithium ion diffusivity using AIMD, POD, and Allegro [5]. Our results show the ability of linear POD to accurately model diffusion in Li superionic conductors.

Vapor-Liquid Equilibria

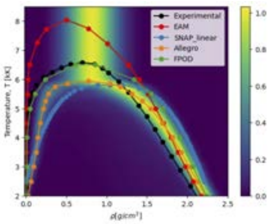


Figure 3: Prediction of vapor-liquid Al phases using SNAP, POD, Allegro, and EAM. The critical point for the experiment is $(\rho_c, T_c) = \left(0.745g/cm^3, 6500K\right)$. POD matches the experimental data better than Allegro, SNAP and the empirical EAM potentials.

Performance Scaling

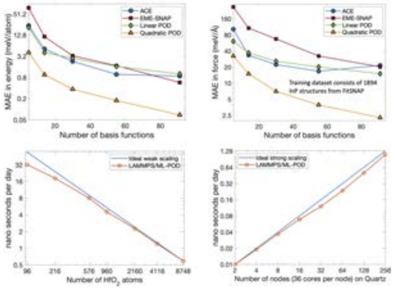


Figure 4: Accuracy comparison of different MLIP models for InP [3] (top) and performance scaling of POD using LAMMPS/ML-POD. For weak scaling (left), we performed MD on various HfO_2 sized systems on 36 cores. For strong scaling (right), we performed MD simulations of 1M HfO_2 atoms.

Melting and Mechanical QOIs

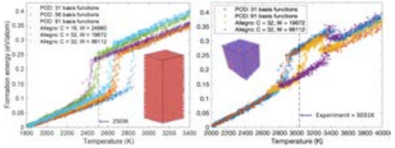
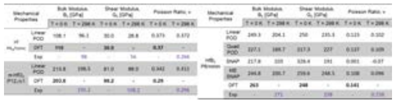


Figure 5: Predictions of the melting point of Hf (left) and HfO_2 (right) for POD and Allegro. The ML-IAPs exhibit close aggreement with the experimental values.



Mechanical Properties		Bulk Modulus B, [GPa]		Shear Modulus G, [GPa]		Poisson Ratio ν	
		T = 0 K	T = 298 K	T = 0 K	T = 298 K	T = 0 K	T = 298 K
Hf P6₃/mmc	Linear POD	108.1	96.1	30.0	26.8	0.375	0.372
	DFT	110	-	30.9	-	0.37	-
	Exp	-	110	-	34	-	0.294
m-HfO₂ P2₁/c	Linear POD	216.8	196.5	81.0	80.0	0.342	0.411
	DFT	203.8	-	88.2	-	0.29	-
	Exp	-	191.2	-	106.2	-	0.296

Mechanical Properties		Bulk Modulus B, [GPa]		Shear Modulus G, [GPa]		Poisson Ratio ν	
		T = 0 K	T = 298 K	T = 0 K	T = 298 K	T = 0 K	T = 298 K
HfB₂ P6/mmm	Linear POD	249.3	204.1	250	235.5	0.123	0.102
	Quad POD	227.1	189.7	217.3	227	0.137	0.109
	SNAP	217.9	220	129.4	191	0.001	-0.07
	ME SNAP	244.8	230.7	259.6	248.5	0.108	0.096
	DFT	263	-	248	-	0.141	-
	Exp	-	271	-	226	-	0.136

Figure 6: Predictions of mechanical properties for Hf, HfO_2 (left) and HfB_2 (right) for POD at 0K and room temperature in comparison with DFT and experiments.

References

[1] N. C. Nguyen and A. Rohskopf. *Journal of Computational Physics*, 480:112030, 2023.

[2] N. C. Nguyen. *Physical Review B*, 107(14):144103, apr 2023.

[3] N. C. Nguyen. *Journal of Computational Physics*, 2023. In submission.

[4] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth, and B. Kozinsky. *Nature Communications*, 14(1):579, 2023.

[5] A. Rohskopf et al. *Journal of Materials Research*, 2023. In submission.

E(3)-equivariant ML-DFT

Dionysios Sema
Massachusetts Institute of Technology

Introduction

The development of an accurate and transferable machine-learned interatomic potentials (ML-IAPs) can take several months and requires computing resources and expertise across disciplines. In active learning approaches, we can use uncertainty metrics from an ensemble of trained potentials to drive an active learning approach, expand our datasets with additional DFT calculations and retrain the next generation of ML-IAPs. In this process, data generation is the most time-consuming part. Our goal is to accelerate this process and achieve potential-in-a-day development cycles. To this end, we propose a E(3)-equivariant ML-DFT that can be used as a proxy DFT model to accelerate DFT calculations or perform DFT calculations that predict the electron density, energy and atomic forces of large and complex systems ($\mathcal{O}(10^5)$ atoms) that are intractable for regular DFT codes.

Selection Scheme

We use a minimax sampling method to select M configurations from a set of K configurations $\{\mathcal{C}_k\}_{k=1}^K$ with $M \ll K$. Define the similarity matrix

$$S_{ij} = \frac{D(\mathcal{C}_i) \cdot D(\mathcal{C}_j)}{\|D(\mathcal{C}_i)\| \|D(\mathcal{C}_j)\|}, \quad i,j = 1,\ldots,K, \tag{1}$$

where $D(\mathcal{C}_i)$ is a vector of descriptors for $\mathcal{C}_i$. Choose a set of M indices $\{i_1,\ldots,i_M\}$, where the first two indices are

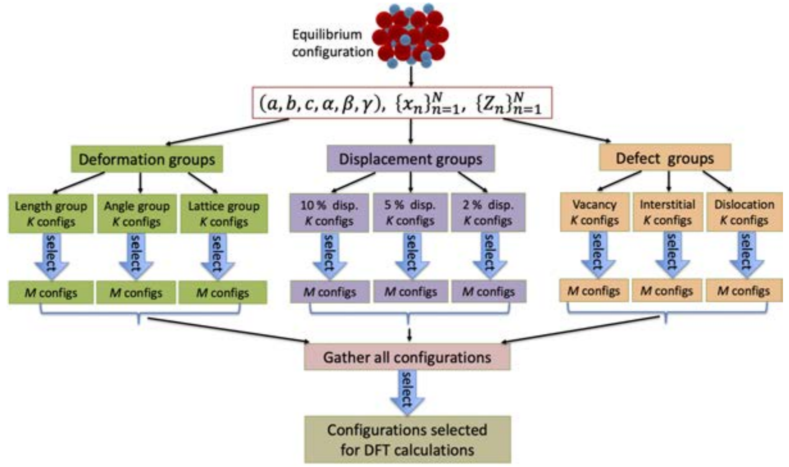
$$(i_1, i_2) = \arg \min_{1 \le i,j, \le N} S_{ij} \tag{2}$$

and, for $m = 3,\ldots,M$,

$$i_m = \arg \min_{1 \le i \le N} \max_{j \in \{i_1,\ldots,i_{m-1}\}} S_{ij}. \tag{3}$$

This method is used in the following algorithm.

Automatic Configuration Generation


Equilibrium configuration
$(a,b,c,\alpha,\beta,\gamma)$, $\{x_n\}_{n=1}^N$, $\{Z_n\}_{n=1}^N$
Deformation groups
Displacement groups
Defect groups
Length group *K* configs
Angle group *K* configs
Lattice group *K* configs
10 % disp. *K* configs
5 % disp. *K* configs
2 % disp. *K* configs
Vacancy *K* configs
Interstitial *K* configs
Dislocation *K* configs
select
M configs
Gather all configurations
select
Configurations selected for DFT calculations

E(3)-GCN Electron Density

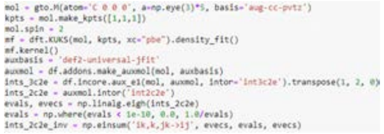
The electron density is a scalar value over all 3D space. We typically represent it using a "basis set". The functions of the basis set have the mathematical form:

$$\Phi_{l,m} = Y_l^m \exp(-\alpha_{l,m} \|r - R_i\|^2), \tag{4}$$

where the first term are the spherical harmonics, and the second is a Gaussian radial basis. The density on a given atom, i, is represented by a linear combination of the basis functions projected onto a delta Dirac function (the origin of DFT formulation). Each basis function has a coefficient that is the weight of that function's contribution:

$$\rho_i = \sum_\lambda \delta(r - r_\lambda) \|\psi_\lambda(r)\|^2 = \sum_l C_i^{l,m} \Phi_{l,m}. \tag{5}$$

We perform DFT runs with pyscf with pbc and project the electron density onto a density fitting basis. We calculate the 3-center 2e tensor integral, $P|ij$, and the 2c2e integral, $P|Q$:



```
mol = gto.M(atom='C 0 0 0', a=np.eye(3)*5, basis='aug-cc-pvtz')
kpts = mol.make_kpts([1,1,1])
mol.spin = 2
mf = dft.KUKS(mol, kpts, xc="pbe").density_fit()
mf.kernel()
auxbasis = 'def2-universal-jfit'
auxmol = df.addons.make_auxmol(mol, auxbasis)
ints_3c2e = df.incore.aux_e1(mol, auxmol, intor='int3c2e').transpose(1, 2, 0)
ints_2c2e = auxmol.intor('int2c2e')
evals, evecs = np.linalg.eigh(ints_2c2e)
evals = np.where(evals < 1e-10, 0.0, 1.0/evals)
ints_2c2e_inv = np.einsum('ik,k,jk->ij', evecs, evals, evecs)
```

We can then extract the final irreps that are determined for each atom type, following the approach of Dunlap et al. [1]. The coefficients and exponents of the basis functions is the data we will train the model with. The raw data is subtracted from the density of the isolated atoms and converted to a molecular graph.

The GCN contains 3 layers of fully connected tensor products with gated block non-linearities. The input are the coordinates and atoms types concatenated to the radial basis as a one-hot vector of length, N, with irreps $Nxoe$. The hidden features have 16 copies with $l_{max} = 4$ with even and odd parity, $p = \{-1, 1\}$. The cutoff radius was set to $r_{cut} = 4$Å

Developing Machine-Learning Interatomic Potentials with julia

Spencer Wyant, Youssef Marzouk

(with package contributions from Emmanuel Lujan, Dallas Foster, Joanna Zou, and other CESMIX contributors)

MITCESMIX

Background

When performing **molecular dynamics (MD)**, the motion of a given particle *i* in a system of *N* atoms is governed by Newton's equation of motion:

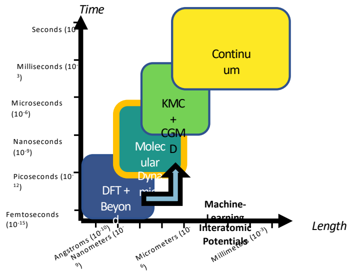
$$-\underbrace{\nabla_i E(\boldsymbol{r},\boldsymbol{z})} = \boldsymbol{F}_i(\boldsymbol{r},\boldsymbol{z}) = m_i \frac{d^2\boldsymbol{r}_i}{dt^2}$$

Potential Energy Surface (PES)

$$E(\boldsymbol{r},z): \mathcal{R} \times \mathcal{Z} \to \mathbb{R} \qquad \mathcal{R} := \bigcup_{N=1}^{\infty} \mathcal{R}_N\,,\ \mathcal{R}_N := \mathbb{R}^{3N}$$

$$\mathcal{Z} := \{1, \dots, N_e\}$$

The **PES** maps the positions and chemical identities of each atom in the system to the total potential energy (in the absence of external forces). A model of the PES should be capable of treating systems with arbitrary numbers of atoms.

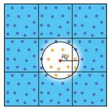


While **density functional theory (DFT)** calculations can provide an accurate PES and downstream properties, their computational cost and O(N3) scaling prevents accessing systems larger than a few hundred atoms and timescales beyond tens of picoseconds. **Machine-learning interatomic potentials (MLIPs)**, which leverage flexible models trained on DFT-computed forces and energies, act as a bridge to the larger spatial and temporal scales accessed with traditional MD methods, bringing the *ab initio* accuracy of more expensive methods to these scales.

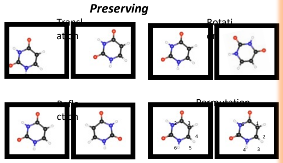
Key Features of MLIPs

Additivity and Locality

$$E(\boldsymbol{r}) = \sum_{i=1}^{N} E_i\left(\{r_j\}_{j\in\mathcal{N}(i)}\right)$$



Symmetry Preserving



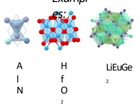
MLIPs can model systems with arbitrary number of atoms by decomposing the total energy into a sum of atom-wise or pairwise terms. Likewise, they exploit the typically local nature of interparticle forces by employing a cutoff function, which only considers atoms within some neighborhood $\mathcal{N}$. Finally, interatomic potentials need to respect the Euclidean group E(3) symmetries (energy invariance and force equivariance) and, from a computational perspective, should be invariant to the ordering of atoms.

Goals

Problem Input

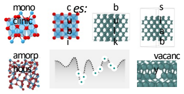
Material System(s)

Examples:



Types of Atomic Environments (Target QoIs)

Examples:



Constraints

- Minimum Inference Speed
  Need fast force routines, especially if simulating long time scales and/or many atoms
- Data Budget
  Since DFT calculations are expensive, want to minimize # of fitting data needed (even more so if using more expensive methods like CCSD)

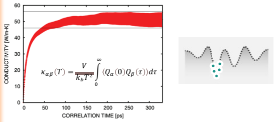
Both constraints depend on available compute resources

MLIPs are typically developed to model particular material system(s) and specific quantities of interest (QoIs), which in turn require modeling different kinds of atomistic environments. Some QoIs require diverse atomistic environments, effectively sampling a larger "volume" of configuration space.

Additionally, it may not be known which environment are relevant *a priori*, necessitating a generalized MLIP. Thus, the choice of material systems and the scope of atomic environments are the primary inputs to the MLIP development process, subject to the constraints listed on the right.

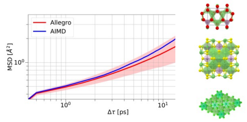
Example Problems

Thermal Properties



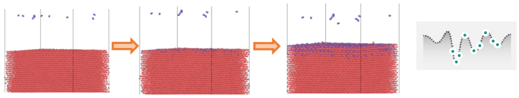
- Only need to sample a small portion of configuration space (i.e., a single minima of the PES).
- However, need high accuracy to capture curvature of the minima, which controls thermal properties

High-Throughput Diffusivity Computation



- Lower accuracies are acceptable, as the goal is down-selection.
- However, need highly sample-efficient strategies (i.e., small number of fitting data for each material system)
- One possible strategy could be to develop a universal MLIP that is subsequently fine-tuned on specific material systems

Hafnium Oxidation



- Requires sampling a wide configuration space, including hafnium metal, hafnia and related sub-oxides, surfaces and surface adsorption, and oxygen gas.
- A major research focus of the CESMIX project

Primary Tasks

Need to Select and/or Optimize

1. *Model Class*
   (e.g., linear, neural network, gaussian process, descriptor type, etc.)
2. *Model Hyperparameters*
   (# of basis vectors, NN architecture, etc.)
3. *Model Parameters*
   (standard optimization)
4. *Fitting Data Generation & Selection*
   (Experimental design, active learning)

Two Perspectives

Scientist Perspective:

- Pre-implemented workflows to generate well-validated MLIPs that satisfy inputs and constraints
- Workflows are automated/semi-automated

Methods Developer Perspective:

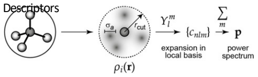
- Facilitate the development of new MLIP models and/or data selection and fitting strategies
- Provide an ability to comparatively evaluate different MLIPs and different

Packages Overview

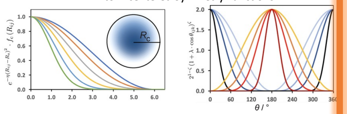
InteratomicPotentials.jl

Atomistic Descriptors *(examples below)*

Smooth Overlap of Atomic Positions (SOAP) Descriptors

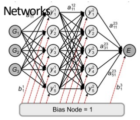


Atom-centered Symmetry Functions

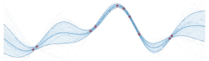


Regression Models *(examples below)*

Feed-Forward Neural Networks



Gaussian Process Regression

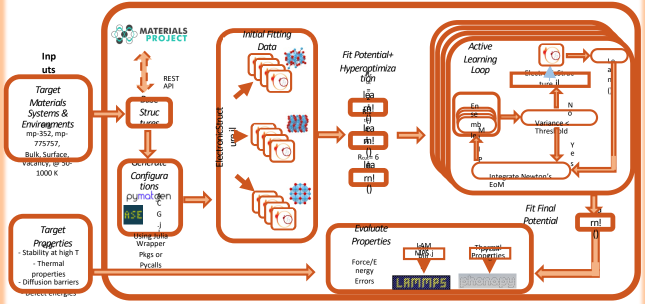


Linear Models

$$E_i(\boldsymbol{r}) = \sum_k \theta_k \phi_k(\boldsymbol{r})$$

InteratomicPotentials.jl is intended to provide a library of existing MLIPs and classical potentials implemented in Julia, along with abstractions that makes it easy to develop and compose new MLIPs. In part, this involves providing implementations of different atomistic descriptors (i.e., ways of representing local atomistic environments) that can be fed into different kinds of regression models, with the flexibility for users to create their own descriptors and/or regressors. More recent MLIP methods that *learn* descriptors (e.g., graph neural network methods) are an important development target, though they likely require a concurrent expansion/improvement of Julia's machine learning ecosystem. InteratomicPotentials.jl can integrate with Molly.jl, a Julia-native MD code, as well as with LAMMPS using the LAMMPS.jl wrapper package, also developed as part of CESMIX.

PotentialLearning.jl



The PotentialLearning.jl package facilitates the development of MLIPs, including data generation and selection, potential fitting, and hyperparameter optimization. It acts as a high-level orchestration script, providing the necessary abstractions to key actions like model fitting and data selection that can be stitched together into larger workflows like that presented above. As a development goal, this package should provide sufficient flexibility to recreate most of the learning strategies that exist in the MLIP literature (including a number of active learning variants), while making it easy for users to develop and test new strategies. One important challenge is to ensure that workflows like the one above are HPC-compatible, i.e., they can be easily run in an HPC environment in a nearly automated fashion. Two strategies are being considered: one approach leverages the Flux resource manager via FluxRM.jl; the other uses the AiiDA workflow manager, a package written in python that would require some Julia integration, but with key benefits including full provenance tracking and an error management & recovery framework.

AiiDA

Figure Credits:

- Behler, Jörg. *Chemical Reviews* 121.16 (2021): 10037-10072.
- Deringer, Volker L., et al. *Chemical Reviews* 121.16 (2021): 10073-10141.
- Musaelian, Albert, et al. *Nature Communications* 14.1 (2023): 579.
- Jones, Reese E., and Kranthi K. Mandadapu. s. *The Journal of chemical physics* 136.15 (2012): 154102.
- Behler, Jörg, *International Journal of Quantum Chemistry* 115.16 (2015): 1032-1050.
- https://docs.jaxgaussianprocesses.com/

Visit us at: github.com/orgs/cesmix-