MDP.jl: The Julia Library of Molecular Dynamics Potentials

Cuong Nguyen

MIT PSAAP-3 Team

05 February 2021

Classical Molecular Dynamics

- **MD simulations require a potential energy function to describe force field governing the interaction among atoms**
  - Determining a suitable potential is the most crucial task in molecular dynamics.
- **Empirical potentials**
  - Lennard-Jones, Tersoff, van der Walls, Coulomb potentials
  - AMBER, CHARMM potentials
  - Reactive potentials: ReaxFF and COMB potentials
  - LAMMPS provides a comprehensive library of empirical potentials
- **Machine learning potentials**
  - Neural network potentials (NNP)
  - Gaussian approximation potentials (GAP)
  - Spectral neighborhood analysis potentials (SNAP)
  - Moment tensor potentials (MTP)

Machine Learning Potentials

- **There are a number of open-source packages for training ML potentials:**
  - Neural network potentials: DeepMD, PyXtal_FF
  - Gaussian approximation potentials: GAP, FLARE
  - Spectral neighborhood analysis potentials: FitSNAP, MAML
  - Moment tensor potentials: MLIP
- **Pros and cons in adapting existing packages:**
  - As these packages has plugins for LAMMPS, they can be used to train ML potentials and run MD simulations through LAMMPS.
  - As most of the existing packages are written in Python, it is difficult to integrate them with our Julia-based software stack.
  - Most of the packages do not quantify uncertainties in the trained ML potential.
  - Several packages do not include an empirical potential in the training.

Zuo, Yunxing, et al. "**Performance and cost assessment of machine learning interatomic potentials**." *The Journal of Physical Chemistry A* 124.4 (2020): 731-745

The Julia Library of MD Potentials

- **Motivation for coupled empirical/ML potentials**
  - Existing empirical potentials (ReaxFF, COMB) have not produced satisfactory results for our target problem. Need to retrain ReaxFF and COMB.
  - ML potentials may be able to fill the gap.
  - ML potentials require significant amount of DFT data for training. Coupled ML/empirical potentials may reduce the amount of data needed to train the model.
- **MDP.jl The Julia library of MD potentials:**
  - Explore a variety of atomistic descriptors and ML methods
  - Exploit automatic differentiation to accelerate training and force calculation
  - Identify near optimal configurations to include in the training data
  - **Provide a comprehensive library of coupled empirical and ML potentials**
  - **Quantify uncertainties for the trained potentials**
  - **Provide fast and accurate potentials for classical MD**
  - **Provide LAMMPS plugins to accelerate MD simulations on exascale supercomputers**

Potential Energy Representation

- **Hybrid characterization of the potential energy surface**

$$E(\boldsymbol{r}^N) = E_{\mathrm{EP}}(\boldsymbol{r}^N) + E_{\mathrm{ML}}(\boldsymbol{r}^N)$$

- **Machine learning potential**

$$E_{\mathrm{ML}}(\boldsymbol{r}^N) = \mathcal{G}(\boldsymbol{d}(\boldsymbol{r}^N), \boldsymbol{w})$$

$$\boldsymbol{d}(\boldsymbol{r}^N) = \{d_1(\boldsymbol{r}^N), \ldots, d_M(\boldsymbol{r}^N)\} \qquad \text{(descriptors)}$$

$$\boldsymbol{w} = (w_1, \ldots, w_K) \qquad \text{(weights)}$$

- **The weights are determined by minimizing a loss function.**

Thompson, A. P., Swiler, L. P., Trott, C. R., Foiles, S. M., & Tucker, G. J. (2015). **Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials**. *Journal of Computational Physics*, *285*, 316-330.

Regression Methods

- **Linear regression (SNAP, MLIP):**

$$E_{\mathrm{ML}}(\boldsymbol{r}^N) = \sum_{m=1}^{M} w_m d_m(\boldsymbol{r}^N)$$

- **Gaussian process regression (GAP, FLARE):**

$$E_{\mathrm{ML}}(\boldsymbol{r}^N) = \sum_{k=1}^{K} w_k \mathcal{K}(\boldsymbol{d}(\boldsymbol{r}^N), \boldsymbol{d}(\boldsymbol{\chi}_k))$$

- **Neural network (nonlinear) regression (DeepMD, MAML):**

$$E_{\mathrm{ML}}(\boldsymbol{r}^N) = \sum_{i=1}^{N} \left(b_i + \sum_{k=1}^{N_h} w_{ik} \mathcal{H}(\boldsymbol{d}(\boldsymbol{r}^N), \boldsymbol{u}_{ik}) \right)$$

Descriptors

- **Descriptors are crucial to the performance of ML potentials**
  - Atom centered symmetry functions (ACSF): neural network potentials
  - SO(3) descriptors: Smooth overlap of atomic position (SOAP) potentials, moment tensor potentials, DeepMD potentials
  - SO(4) descriptors: GAP and SNAP potentials
- **They key requirements of the descriptors:**
  - The descriptors are invariant to permutation, translation, and rotation
  - The descriptors and their derivatives are fast to evaluate
  - The descriptors are able to capture various atomic interaction phenomena
- **In addition to existing descriptors, we consider a new SO(3) descriptor.**

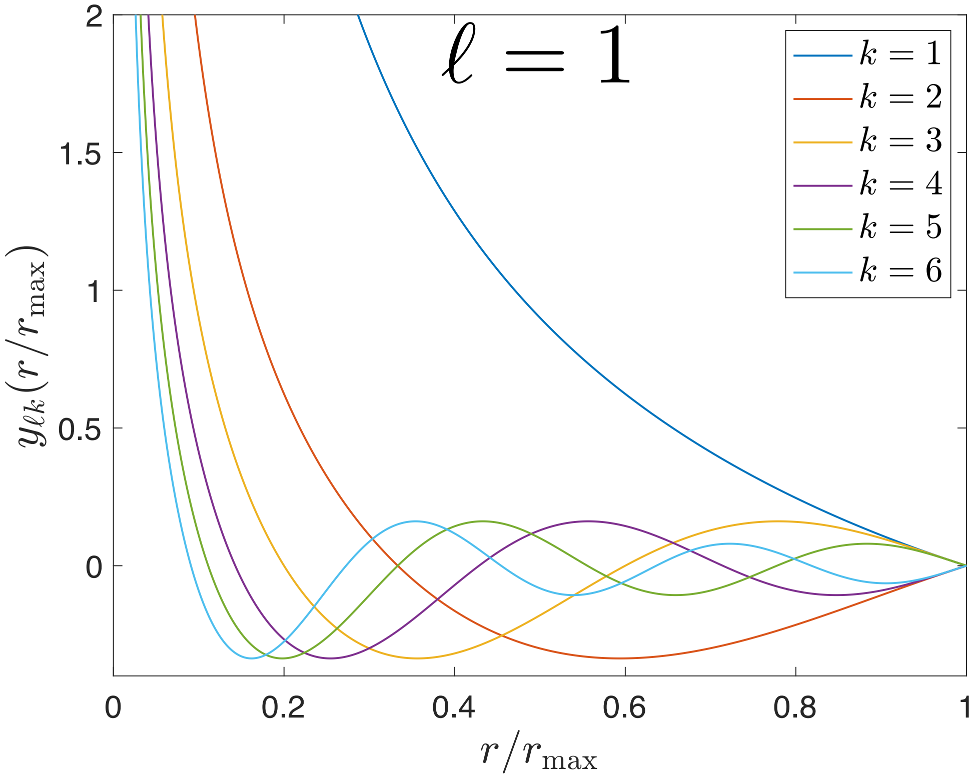
Bartók, Albert P., Risi Kondor, and Gábor Csányi. "***On representing chemical environments***." *Physical Review B* 87.18 (2013): 184115.

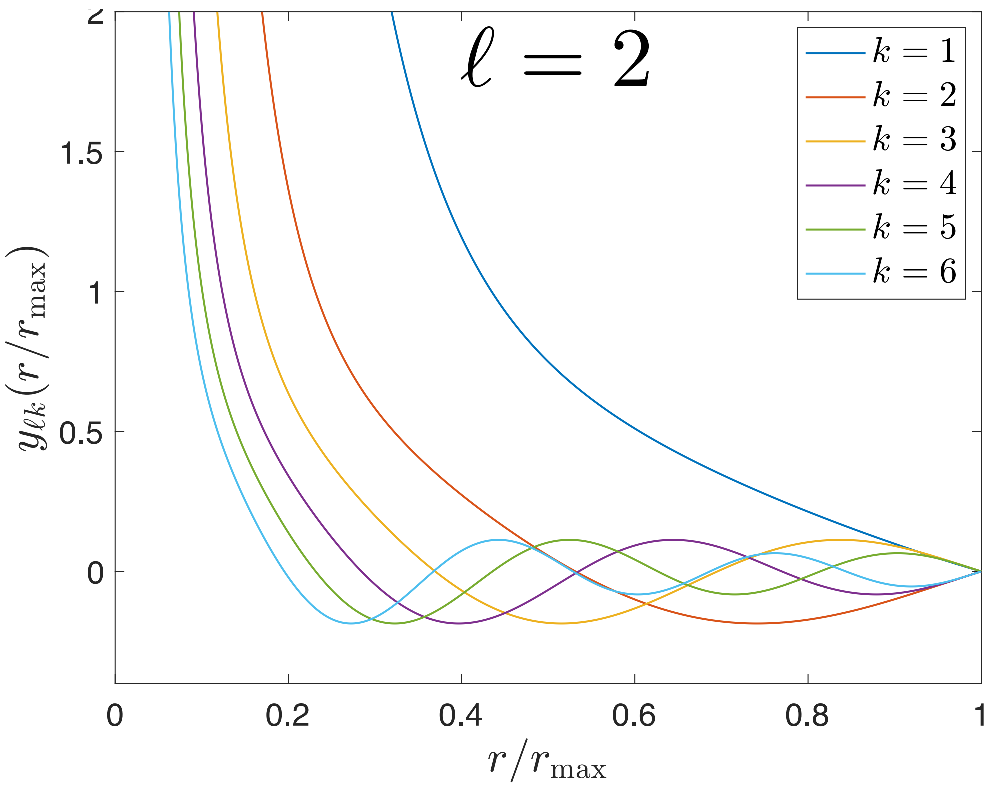
Behler, Jörg. "***Perspective: Machine learning potentials for atomistic simulations***." *The Journal of chemical physics* 145.17 (2016): 170901

New SO(3) Descriptors

- The descriptors are based on products of spherical harmonics $\boldsymbol{Y_{lm}(\theta, \phi)}$ and spherical Bessel functions $\boldsymbol{y_{lk}(r)}$

$$u_{klm}(\boldsymbol{r}) = y_{lk}(r)Y_{lm}(\theta, \phi), \quad 1 \leq k \leq K, 0 \leq l \leq L, -l \leq m \leq l$$





New SO(3) Descriptors

- **Introduce local atomic functions**

$$a_{iklm}(\boldsymbol{r}^N) = \sum_{j}^{|\boldsymbol{r}_j - \boldsymbol{r}_i| \leq r_c} u_{klm}(\boldsymbol{r}_j - \boldsymbol{r}_i), \quad 1 \leq i \leq N$$

- **Define the power spectrum components of the descriptors:**

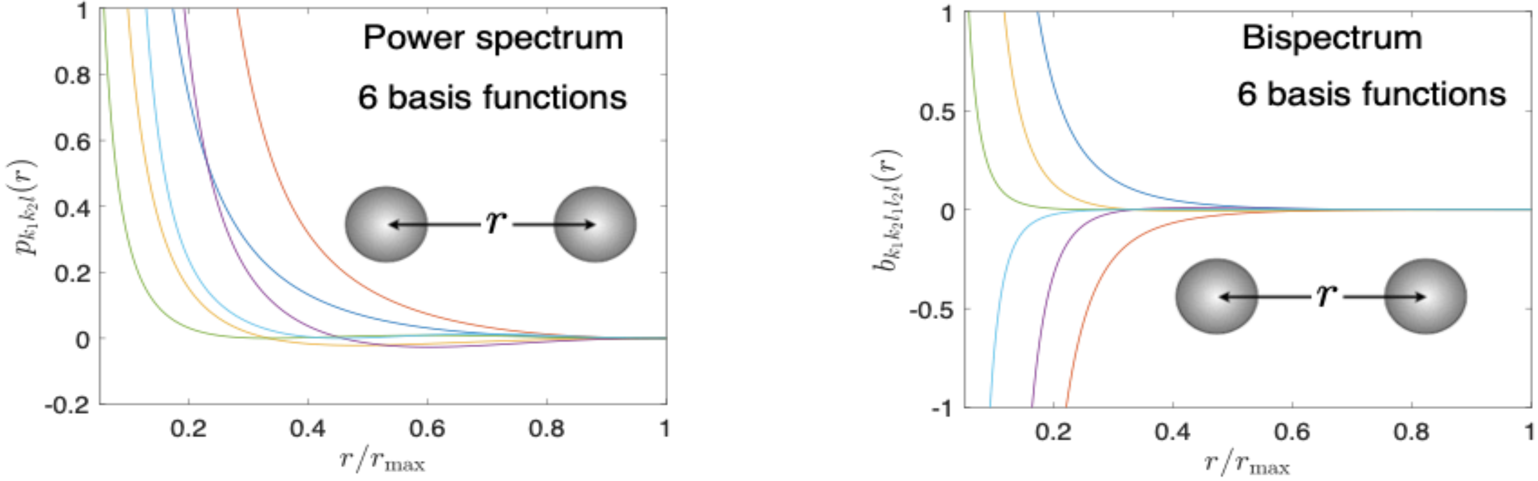
$$p_{ikk'l}(\boldsymbol{r}^N) = \sum_{m=-l}^{l} a^*_{iklm}(\boldsymbol{r}^N) a_{ik'lm}(\boldsymbol{r}^N), \quad 1 \leq i \leq N$$

- **Define the bispectrum components of the descriptors:**

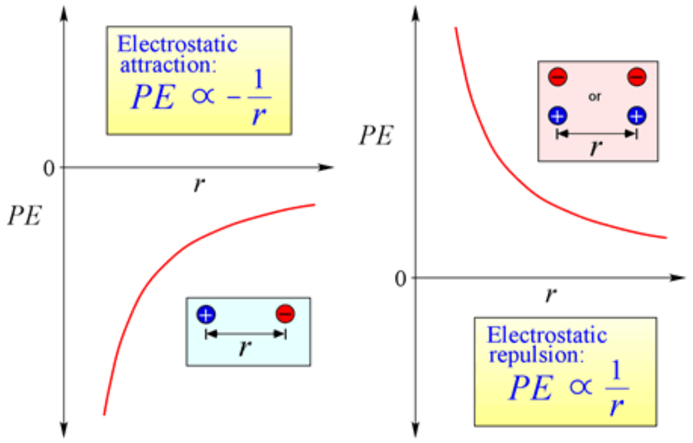
$$b_{ikk'll_1l_2}(\boldsymbol{r}^N) = \sum_{m=-l}^{l} \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} a^*_{iklm}(\boldsymbol{r}^N) C^{l_1l_2l}_{m_1m_2m} a_{ik'l_1m_1}(\boldsymbol{r}^N) a_{ik'l_2m_2}(\boldsymbol{r}^N)$$

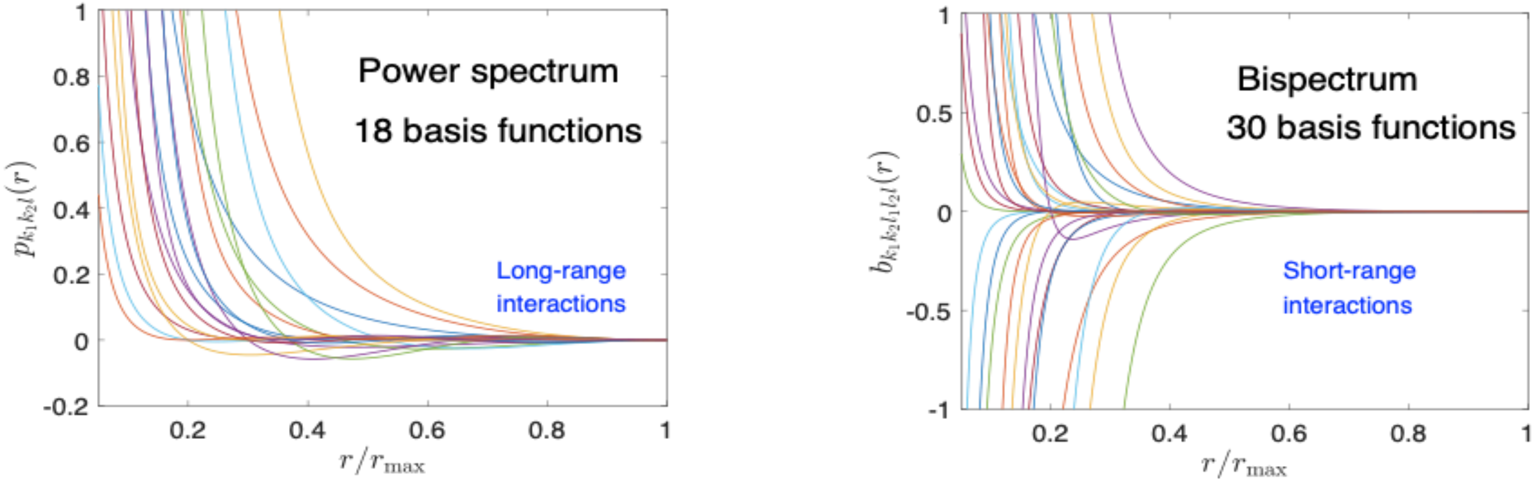
New SO(3) Descriptors

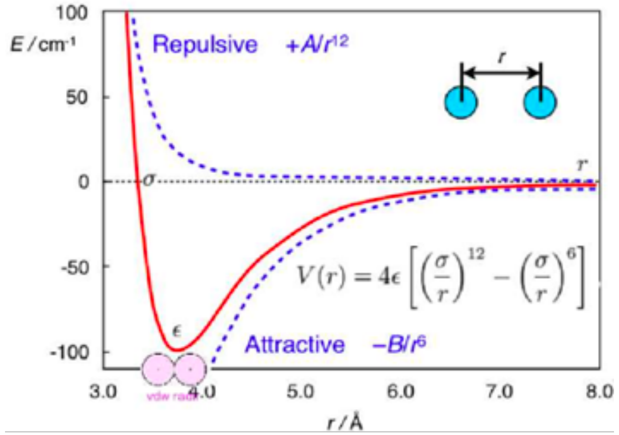
- **Radial components of the descriptors**



Radial basis functions for $K = 2$ and $L = 1$ (first-order)







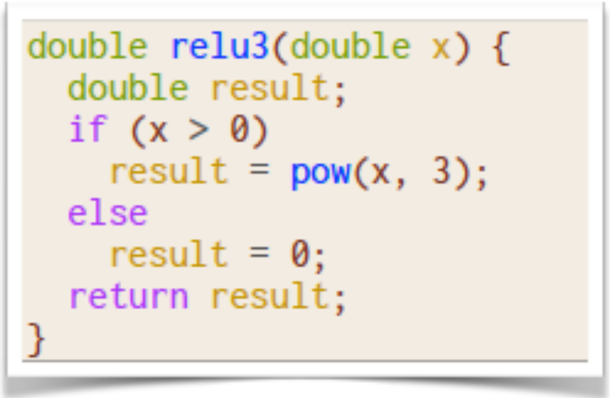
The Lennard-Jones potential

Radial basis functions for $K = 3$ and $L = 2$ (second-order)

Automatic Differentiation

- Force calculation needs the partial derivatives of the descriptors
- For SO(3) and SO(4) descriptors, their analytical derivatives are an order of magnitude more expensive than to compute the descriptors.
- We will use automatic differentiation to calculate the derivatives to reduce the cost of force calculation.
- **Enzyme** is a high-performance AD software developed by our CS group.

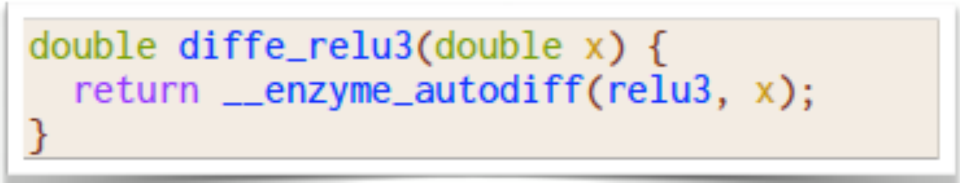
C Source



```
double relu3(double x) {
  double result;
  if (x > 0)
    result = pow(x, 3);
  else
    result = 0;
  return result;
}
```

https://enzyme.mit.edu/

Enzyme Usage



```
double diffe_relu3(double x) {
  return __enzyme_autodiff(relu3, x);
}
```

Thompson, A. P., Swiler, L. P., Trott, C. R., Foiles, S. M., & Tucker, G. J. (2015). **Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials**. *Journal of Computational Physics*, *285*, 316-330.

Potential Fitting

- **Find the weights to minimize the following loss function**

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{j=1}^{J} \alpha_j (E_{\mathrm{ML}}(\boldsymbol{\chi}_j, \boldsymbol{w}) - \bar{E}(\boldsymbol{\chi}_j))^2 + \beta_j \|\boldsymbol{F}_{\mathrm{ML}}(\boldsymbol{\chi}_j, \boldsymbol{w}) - \bar{\boldsymbol{F}}(\boldsymbol{\chi}_j)\|^2,$$

where

$$\bar{E}(\boldsymbol{\chi}_j) = E_{\mathrm{DFT}}(\boldsymbol{\chi}_j) - E_{\mathrm{EP}}(\boldsymbol{\chi}_j)$$

$$\bar{\boldsymbol{F}}(\boldsymbol{\chi}_j) = \boldsymbol{F}_{\mathrm{DFT}}(\boldsymbol{\chi}_j) - \boldsymbol{F}_{\mathrm{EP}}(\boldsymbol{\chi}_j)$$

- **How do we choose the configurations $\{\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_J\}$?**
- **How do we ensure the accuracy of the ML potential?**
- **How do we choose new configurations if necessary?**

Uncertainty Quantification

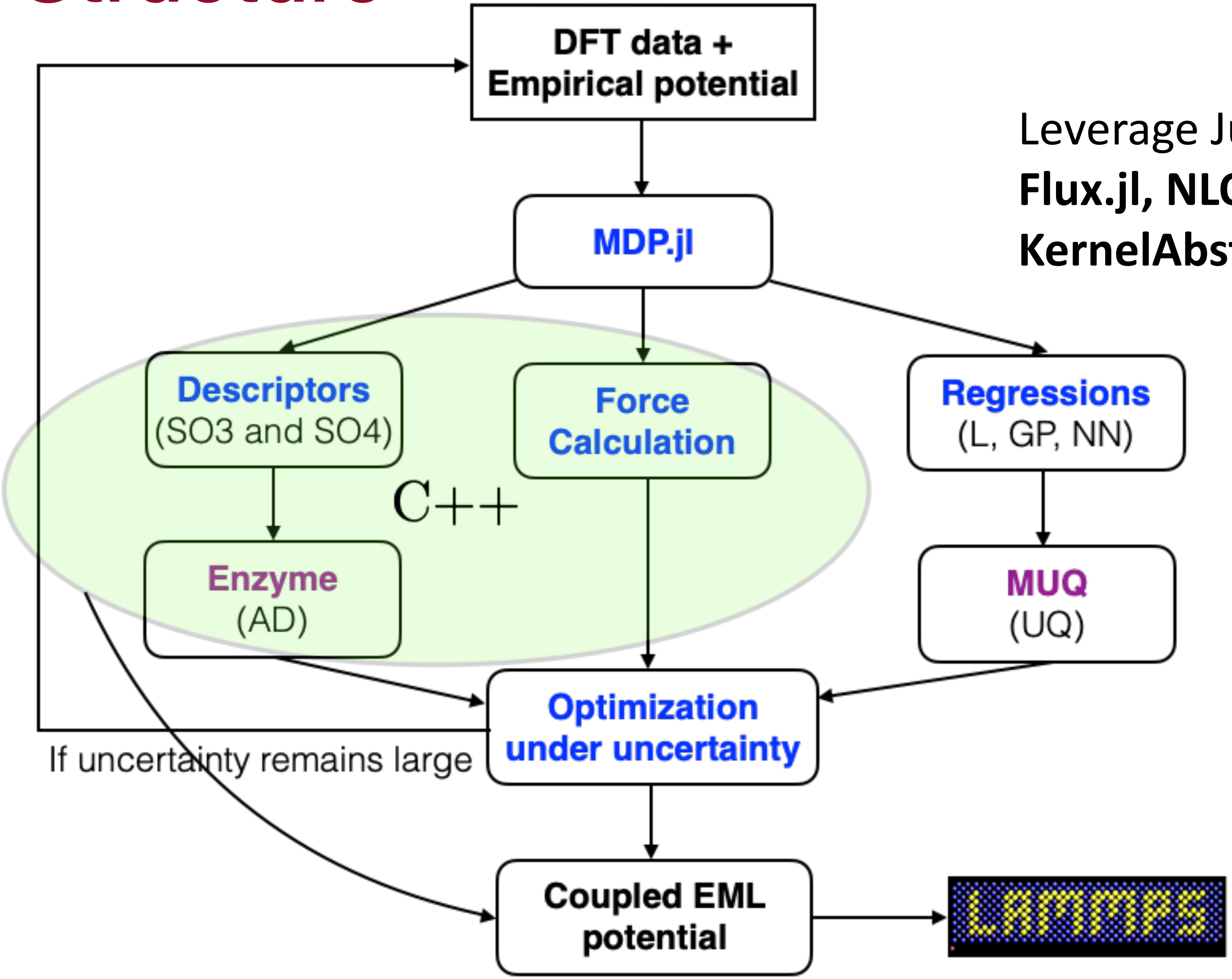
- A **key advantage** of developing our own potential-fitting code (besides flexibility, performance, integration) is the ability to **introduce UQ** from the ground up!
- **UQ strategy:**
  - Recast deterministic/optimization-driven potential fitting (point estimation) as *fully Bayesian inference*
  - Characterize posterior on potential parameters either using *sampling* algorithms (MCMC) or fast/approximate *variational inference*
    - Gradients of the misfit (log-likelihood) will drive either of these approaches
    - Gradient-based dimension reduction and sensitivity analysis also useful
  - Use posterior uncertainty in predictions to drive *active learning* approaches that select/demand new DFT data
  - Calculations much easier in the Gaussian process case (cf. GAP and FLARE), but we will seek greater generality

Uncertainty Quantification

- Longer-term: start thinking about *structural error* in potentials, and adaptation of model complexity
  - Model discrepancy terms
  - Adaptive multi-fidelity potentials
  - Robust Bayesian inference under misspecification

- Stand-up of our initial UQ strategy is a priority for latter half of Year 1
- Make use of experience/algorithms from MUQ (http://muq.mit.edu), but write new Julia libraries linked to MDP.jl

MDP.jl Structure



Leverage Julia’s packages:
Flux.jl, NLOPT.jl, CUDA.jl, KernelAbstractions.jl

Questions/Comments?

- **MDP.jl: The Julia library of MD potentials**
  - Implement a variety of atomistic descriptors and ML methods
  - Identify near optimal configurations to include in the training data via UQ methods
  - Employ automatic differentiation to accelerate training and force calculation
  - **Provide a comprehensive library of coupled empirical and ML potentials**
  - **Quantify uncertainties for the trained potentials**
  - **Enable fast and accurate MD simulations by implementing and auto-tunning new algorithms for force calculation (tomorrow talk)**