Machine Learning Based Multi-scale Modeling:

How can we construct interpretable and truly reliable models using concurrent learning?

Weinan E

Princeton University

Joint work with:

Jiequn Han, Linfeng Zhang

Chao Ma, Zheng Ma, Han Wang

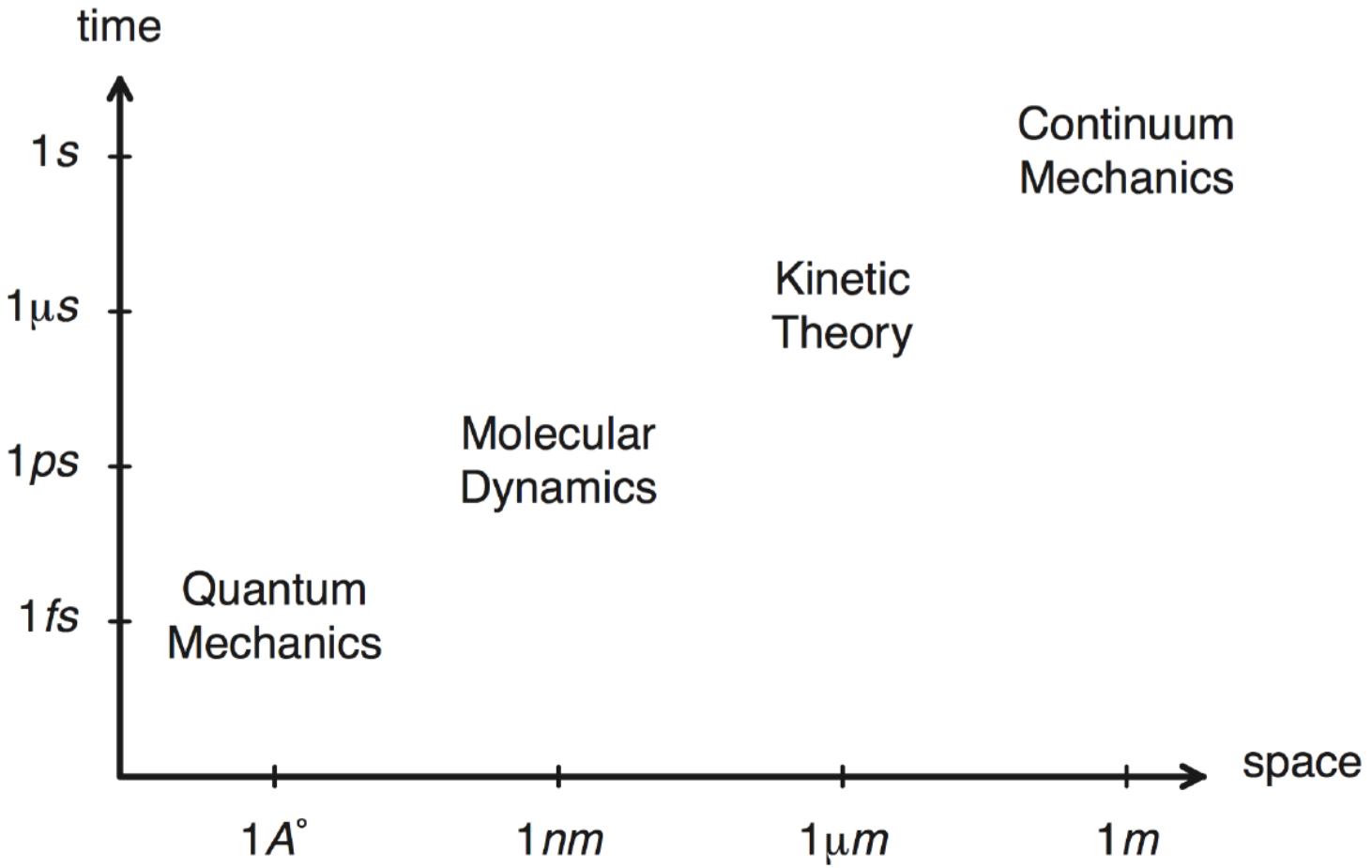
Roberto Car, Wissam A. Saidi

Outline

Outline

Multi-scale modeling: A very ambitious program

Objective: remove the ad hoc part of the modeling process and use only truly reliable models.



Sequential vs concurrent coupling

- sequential coupling: precompute the macro-scale model using the micro-scale model.
- concurrent coupling: couple the macro-scale and the micro-scale models "on-the-fly".

Figure: schematic of the heterogeneous multi-scale method (E and Engquist 2003)

Difficulty: parametrizing the components needed in the macro-model, using data from the micro-scale model. For problems without separation of scales, this has been an essential obstacle.

Machine learning comes to the rescue

Two objectives:

- multi-scale modeling in situations without scale separation
- interpretable and truly reliable physical models with machine learning

Outline

Sequential vs concurrent learning

Where are the data sets? (It is very expensive to get the data)

- **sequential learning**: first collect labeled data $\{\boldsymbol{x}_j, y_j\}$, then perform learning
- **concurrent learning**: generate the data set on the fly as learning proceeds

compare with "**active learning**": having unlabeled data $\{\boldsymbol{x}_j\}$, and decide which ones to label and use them to perform learning

concurrent learning: generate "**optimal data set**" (both unlabeled and labeled, representative enough yet as small as possible)

the latter is a more interactive process

The exploration-labeling-training procedure for concurrent learning

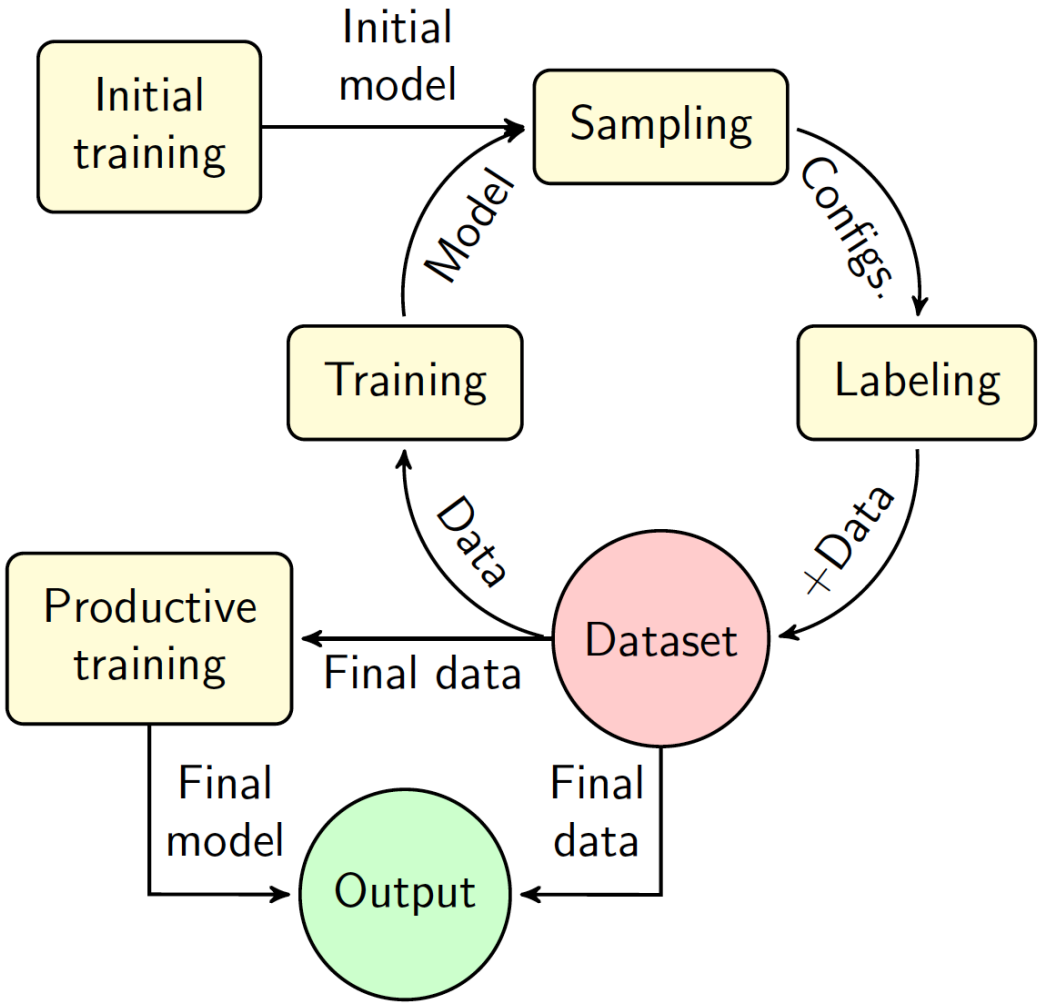
Zhang, Wang and E (2018), J. Chem. Phys.

Start out with no (macro-scale) model, no data; but with a micro-scale model.
Repeat the following steps:

1. **exploration**: explore the configuration space, and decide which configurations need to be labeled.
2. **labeling**: compute the micro-scale solutions for the configurations that need to be labeled. This is our data set.
3. **training**: train the macro-scale model, and use it to help the exploration

Similar to "active learning" but more interactive.....

DP-GEN: concurrent learning for uniformly accurate model



Indicator: $\epsilon = \max_i \sqrt{\langle \| \boldsymbol{f}_i - \bar{\boldsymbol{f}}_i \|^2 \rangle}$, $\bar{\boldsymbol{f}}_i = \langle \boldsymbol{f}_i \rangle$ "Active Learning of Uniformly Accurate Inter-atomic Potentials for Materials Simulation." arXiv:1810.11890 (2018).

Outline

Example 1: Molecular dynamics

Traditional dilemma: accuracy vs cost.

$$E = E(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_N),$$

$$m_i \frac{d^2 \boldsymbol{x}_i}{dt^2} = \boldsymbol{F}_i = -\nabla_{\boldsymbol{x}_i} E.$$

Two ways to calculate E and $\boldsymbol{F}$:

- Computing the inter-atomic forces on the fly using QM, e.g. the Car-Parrinello MD. Accurate but expensive:

$$E = \langle \Psi_0 | H_e^{KS} | \Psi_0 \rangle, \ \mu \ddot{\phi}_i = H_e^{KS} \phi_i + \sum_j \Lambda_{ij} \phi_j.$$

- Empirical potentials: efficient but unreliable. The Lennard-Jones potential:

$$V_{ij} = 4\epsilon[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6}], \ E = \frac{1}{2} \sum_{i \neq j} V_{ij}.$$

Integrating ML with molecular modeling

New paradigm:

- quantum mechanics model – data generator
- machine learning – parametrize (represent) the model
- molecular dynamics – simulator

Interaction potential between atoms in molecular dynamics

Consider a system of N atoms, $\boldsymbol{r} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_N\}$.

the coordinate matrix $\mathcal{R} \in \mathbb{R}^{N\times 3}$:

$$\mathcal{R} = \{\boldsymbol{r}_1^T, \cdots, \boldsymbol{r}_i^T, \cdots, \boldsymbol{r}_N^T\}^T, \ \boldsymbol{r}_i = (x_i, y_i, z_i).$$

r_c: a pre-defined cut-off radius.

For atom i, defined its neighbors $\mathcal{N}_{r_c}(i) = \{j | r_{ij} < r_c\}$, and $\boldsymbol{r}_{ji} \equiv \boldsymbol{r}_j - \boldsymbol{r}_i$.

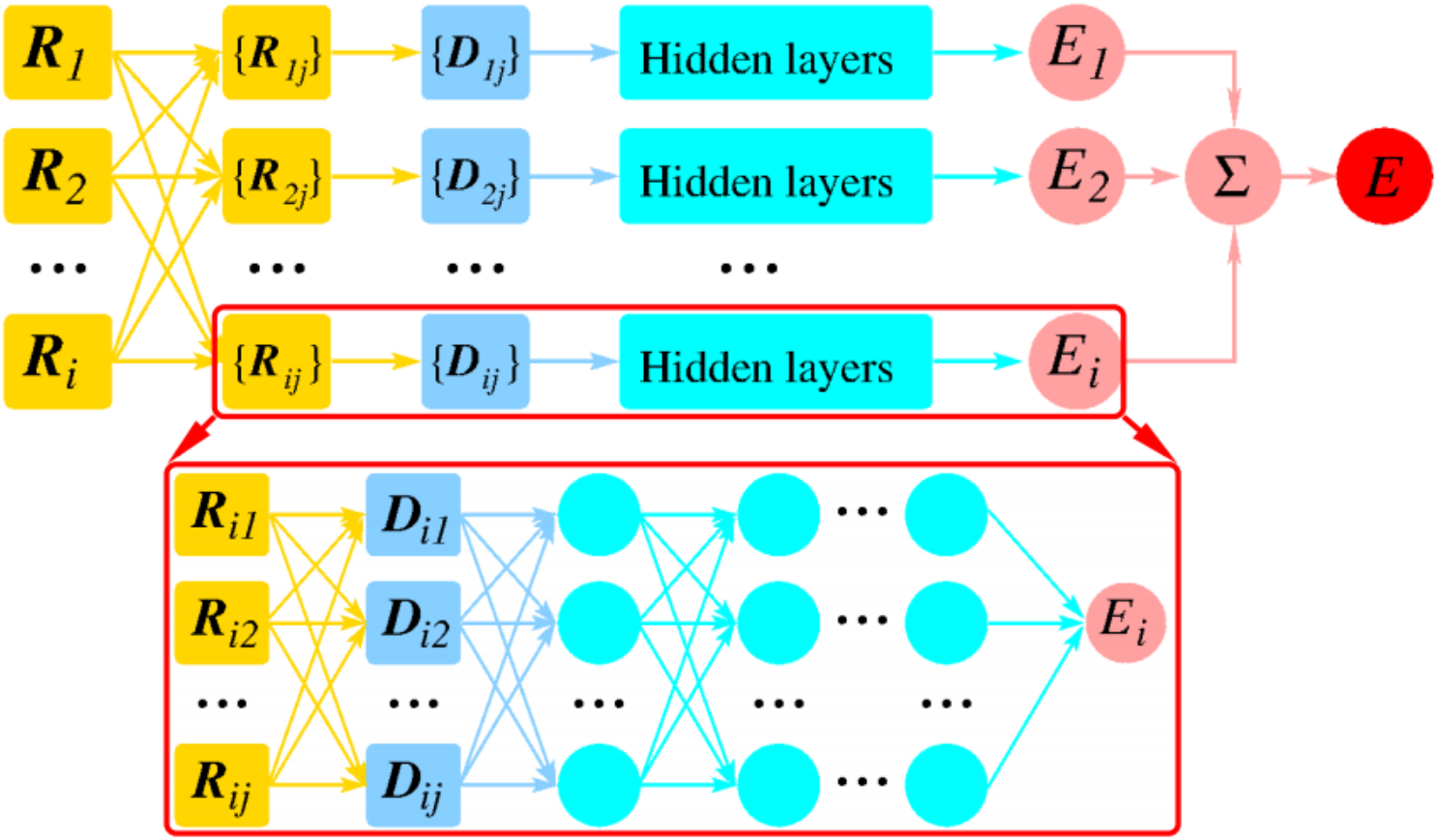
Define i's local environment matrix

$$\mathcal{R}^i = \{\boldsymbol{r}_{1i}^T, \cdots, \boldsymbol{r}_{ji}^T, \cdots, \boldsymbol{r}_{N_i,i}^T\}^T, \ \boldsymbol{r}_{ji} = (x_{ji}, y_{ji}, z_{ji}).$$

$E(\mathcal{R}) \equiv E$: a map from the coordinate matrix to the potential energy;

Deep Potential: construction

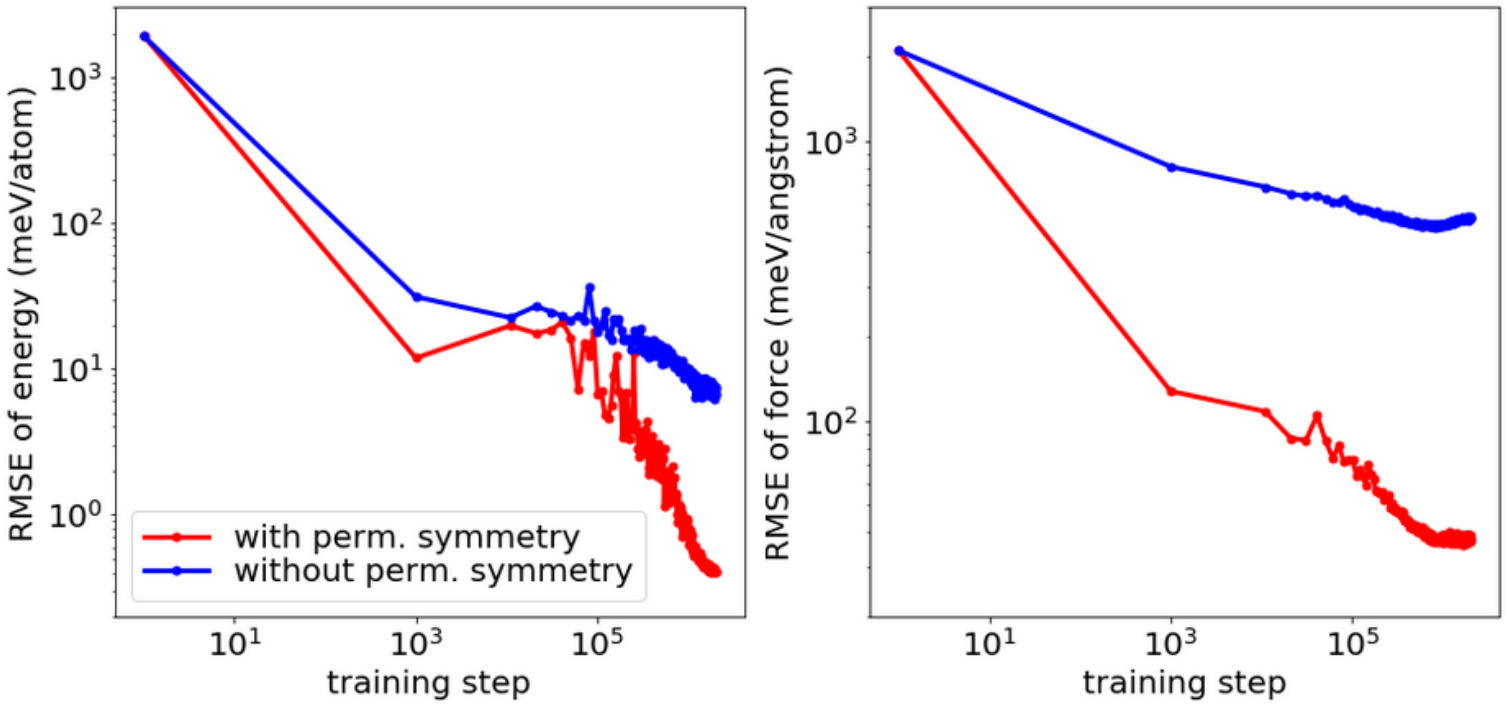
Structure: composite neural networks (NNs). $E = \sum_i E^i$.



Models of this type are extensible, which implies linear scaling. Behler, J., Parrinello, M. (2007). Phys. Rev. Lett., 98(14), 146401.

Deep Potential (Comm. Comp. Phys. 23.3 (2018): 629-639.), DPMD (Phys. Rev. Lett. 120 (2018), 143001)

The importance of preserving the symmetries



Preserving symmetry: Poor man's version

- remove translational and rotational symmetry by fixing a local frame of reference
- remove permutational symmetry by fixing an ordering of the atoms in the neighborhood

Creates small discontinuity when atoms switch their orders.

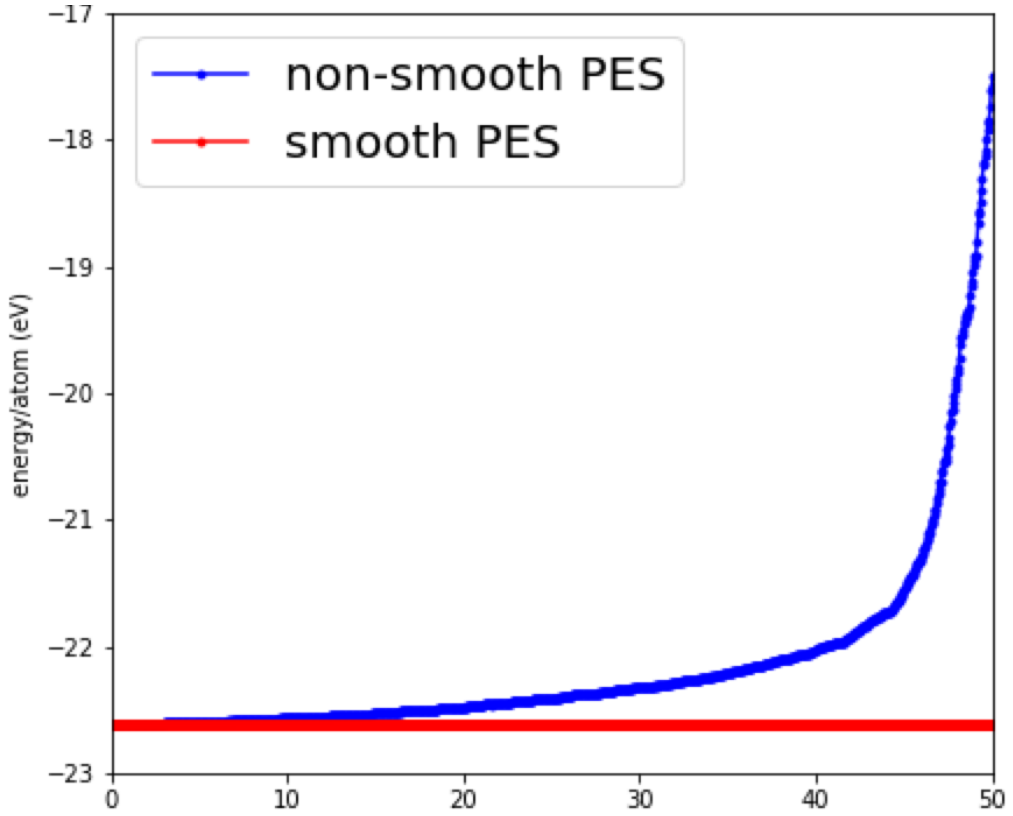


Figure: deep potential molecular dynamics (DPMD)

Preserving the symmetries

Translation, rotation, and permutation.

$$\hat{T}_{\boldsymbol{b}} f(\boldsymbol{r}) = f(\boldsymbol{r} + \boldsymbol{b}), \;\; \hat{R}_{\mathcal{U}} f(\boldsymbol{r}) = f(\boldsymbol{r}\mathcal{U}),$$

$$\hat{P}_{\sigma} f(\boldsymbol{r}) = f(\boldsymbol{r}_{\sigma(1)}, \boldsymbol{r}_{\sigma(2)}, ..., \boldsymbol{r}_{\sigma(N)})$$

- Translation and Rotation:

$$\Omega^i_{jk} = \boldsymbol{r}_{ji} \cdot \boldsymbol{r}_{ki}.$$

Lemma: Ω^i_{jk} is an overcomplete array of basic invariants with respect to rotation, reflection, and translation.

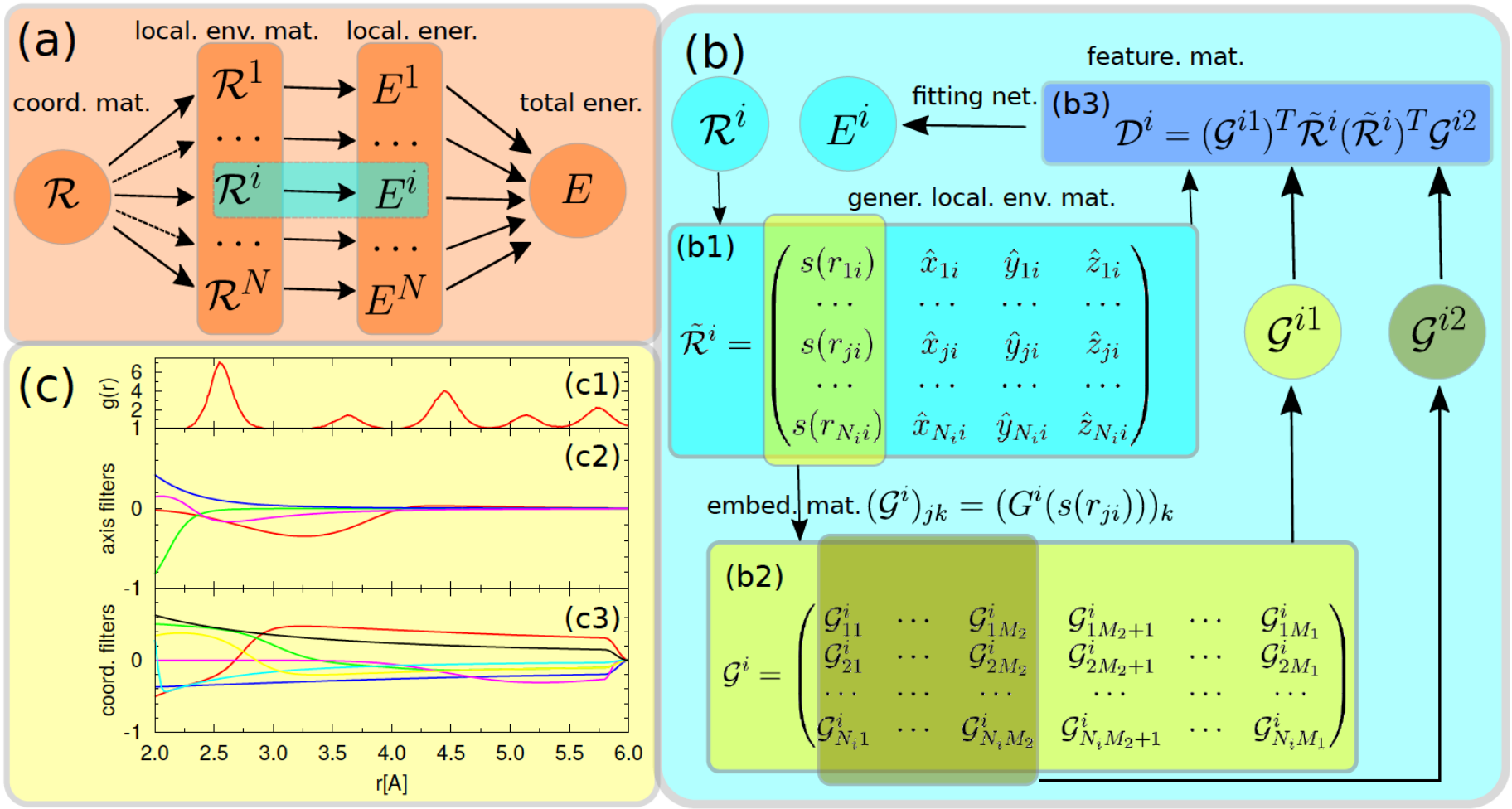
- Permutation:

$$\sum_{j \in \mathcal{N}(i)} g(\boldsymbol{r}_{ji}) \boldsymbol{r}_{ji}.$$

Lemma: A function $f(\boldsymbol{r}_{1i}, ..., \boldsymbol{r}_{ji}, ..., \boldsymbol{r}_{N_i i})$ is invariant to the permutation of instances in $\boldsymbol{r}_{ji}$, if and only if it can be decomposed in the form $\rho(\sum_{j \in \mathcal{N}(i)} g(\boldsymbol{r}_{ji}) \boldsymbol{r}_{ji})$, for suitable transformations g and ρ.

Deep potential: smooth version

The whole sub-network consists of an encoding net $\mathcal{D}^i(\mathcal{R}^i)$ and a fitting net $E^i(\mathcal{D}^i)$.



(Rotation: $\tilde{\mathcal{R}}^i(\tilde{\mathcal{R}}^i)^T$, permutation: $(\mathcal{G}^{i1})^T\tilde{\mathcal{R}}^i$ and $(\tilde{\mathcal{R}}^i)^T\mathcal{G}^{i2}$.)

DeepPot-SE (arxiv: 1805.09003, NIPS 2018), see also Behler and Parrinello, PRL 2007.

DP-GEN: Automatic generation of deep potentials using concurrent learning

The exploration-labeling-training procedure

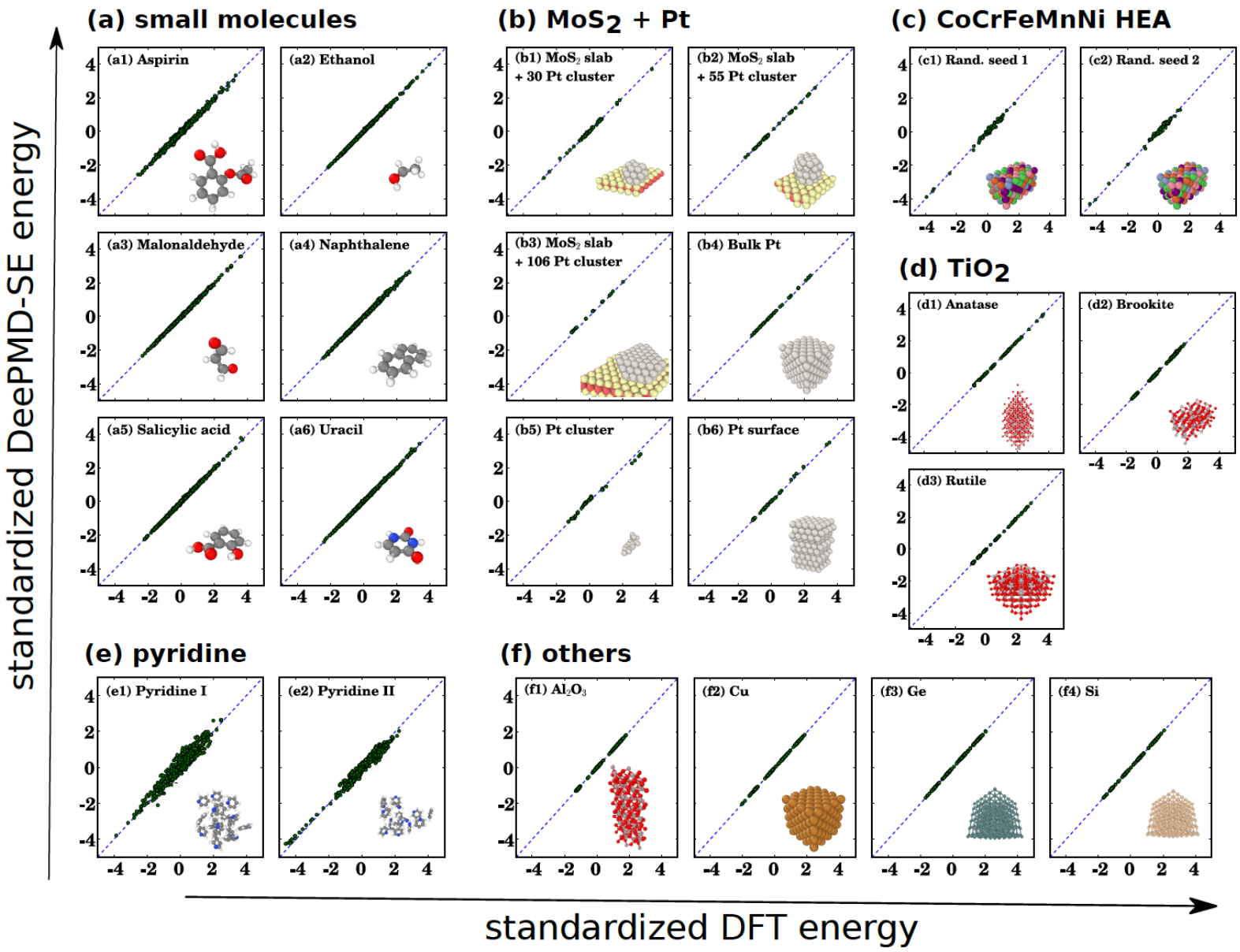
- Exploration:
  - Sample the (T, p) space
  - For each value of (T, p), sample the canonical ensemble (using DPMD).
  - In addition, initialize the exploration with a variety of different initial configurations.
- Labeling: Using DFT (with periodic boundary condition)
- Training: Using "deep potential"

Systems			Al		Mg		Al-Mg alloy	
Type	Lattice	#atom	#Confs	#Data	#Confs	#Data	#Confs	#Data
Bulk	FCC	32	15,174,000	1,326	15,174,000	860	39,266,460	7,313
	HCP	16	15,174,000	908	15,174,000	760	18,999,900	2,461
	Diamond	16	5,058,000	1,026	5,058,000	543	5,451,300	2,607
	SC	8	5,058,000	713	5,058,000	234	2,543,940	667
Surface	FCC (100)	12	3,270,960	728	3,270,960	251	62,203,680	1,131
	FCC (110)	16[a],20[b]	3,270,960	838	3,270,960	353	10,744,2720	2,435
	FCC (111)	12	3,270,960	544	3,270,960	230	62,203,680	1,160
	HCP (0001)	12	3,270,960	39	3,270,960	109	62,203,680	176
	HCP $(10\bar{1}0)$	12	3,270,960	74	3,270,960	167	62,203,680	203
	HCP $(11\bar{2}0)$	16[a],20[b]	3,270,960	293	3,270,960	182	107,442,720	501
sum			60,089,760	6,489	60,089,760	3,689	529,961,760	18,654

[a]Pure Al
[b]Mg and Al-Mg alloy

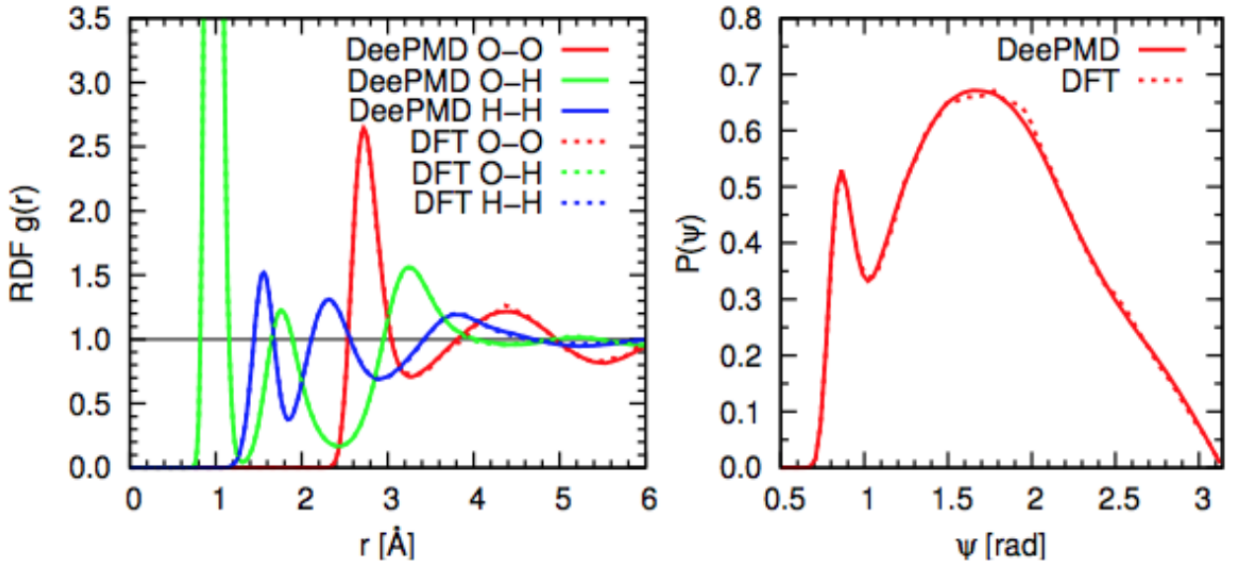
$\sim$0.005% configurations explored by DPMD are selected for labeling.

Case 1: accuracy is comparable to the accuracy of the data

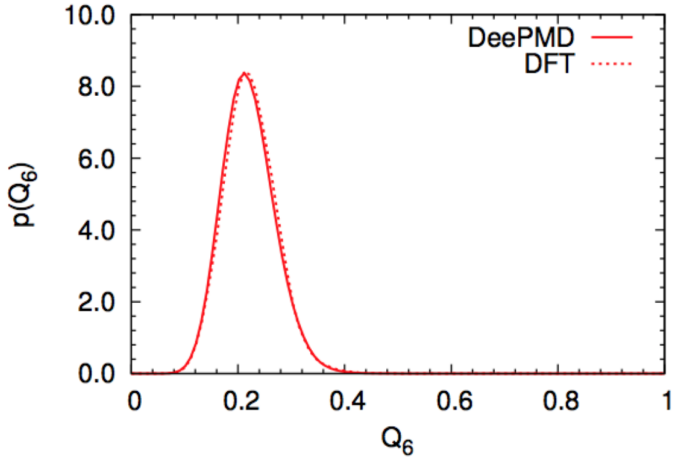


Case 2: structural information of DFT water

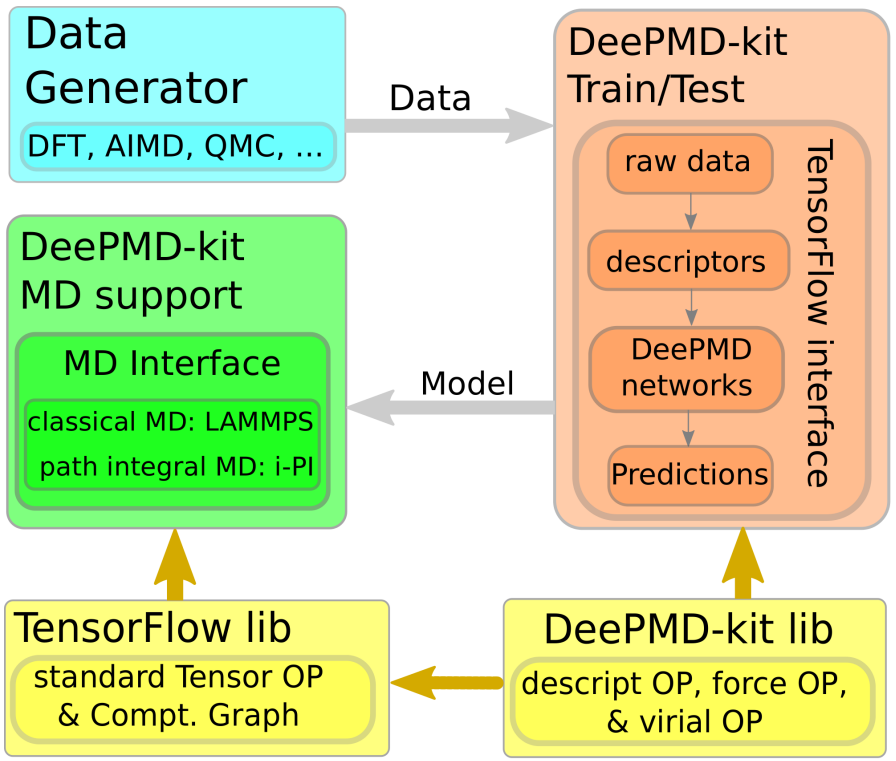
Radial and angular distribution function of liquid water (PI-AIMD):

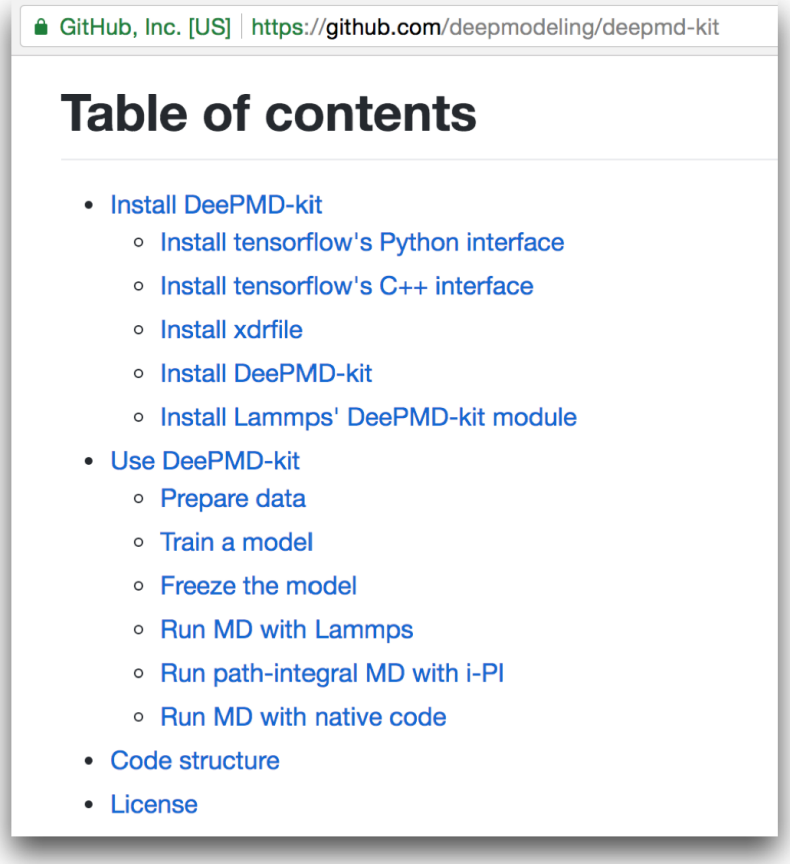


Distribution of the Steinhardt order parameter $\bar{Q}_6$:



Open-source softwares: DeePMD-kit



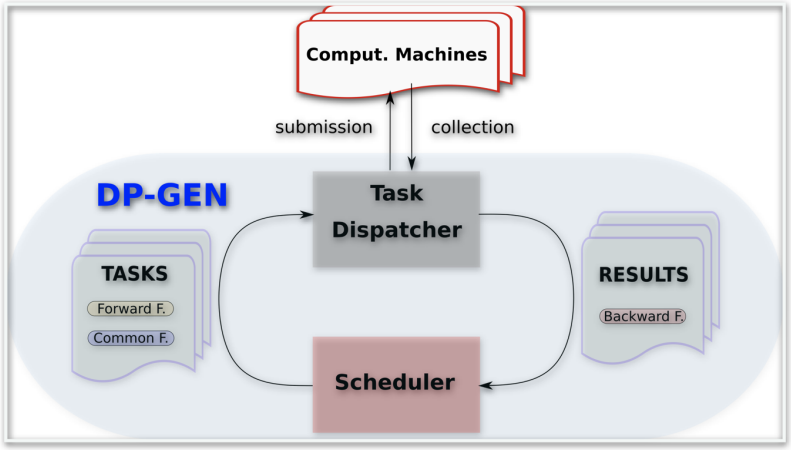


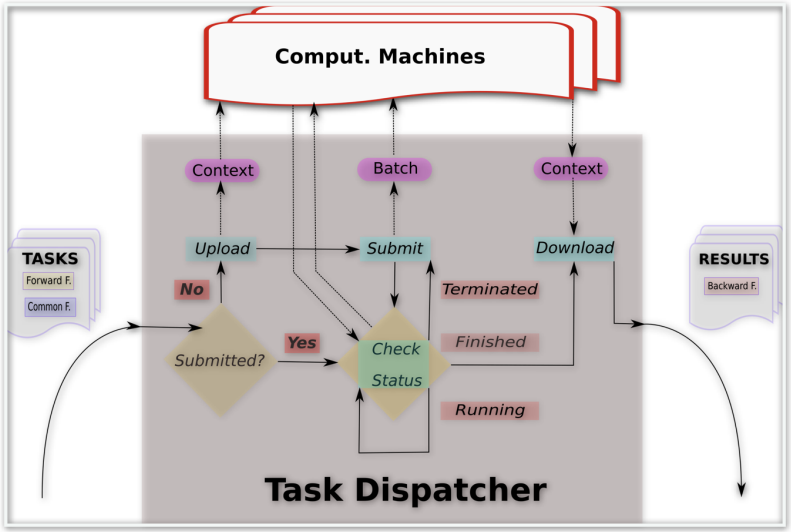
- TensorFlow: efficient network operators
- LAMMPS, i-PI; MPI/GPU support.

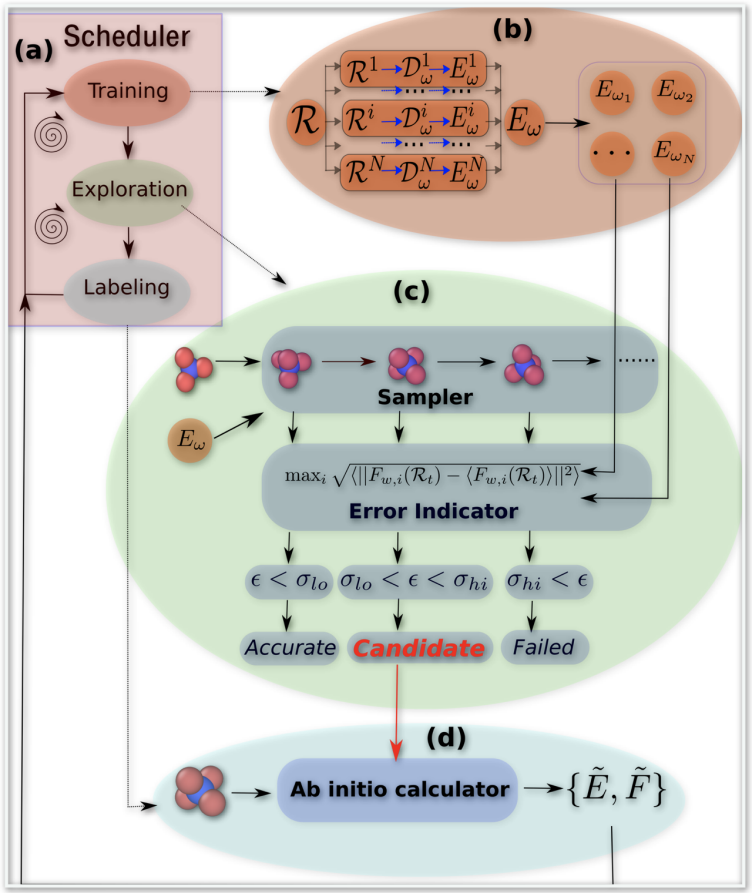
Free download from https://github.com/deepmodeling/deepmd-kit

H. Wang,. et al, .Comp.Phys.Comm., 0010-4655 (2018).

Open-source softwares: DP-GEN

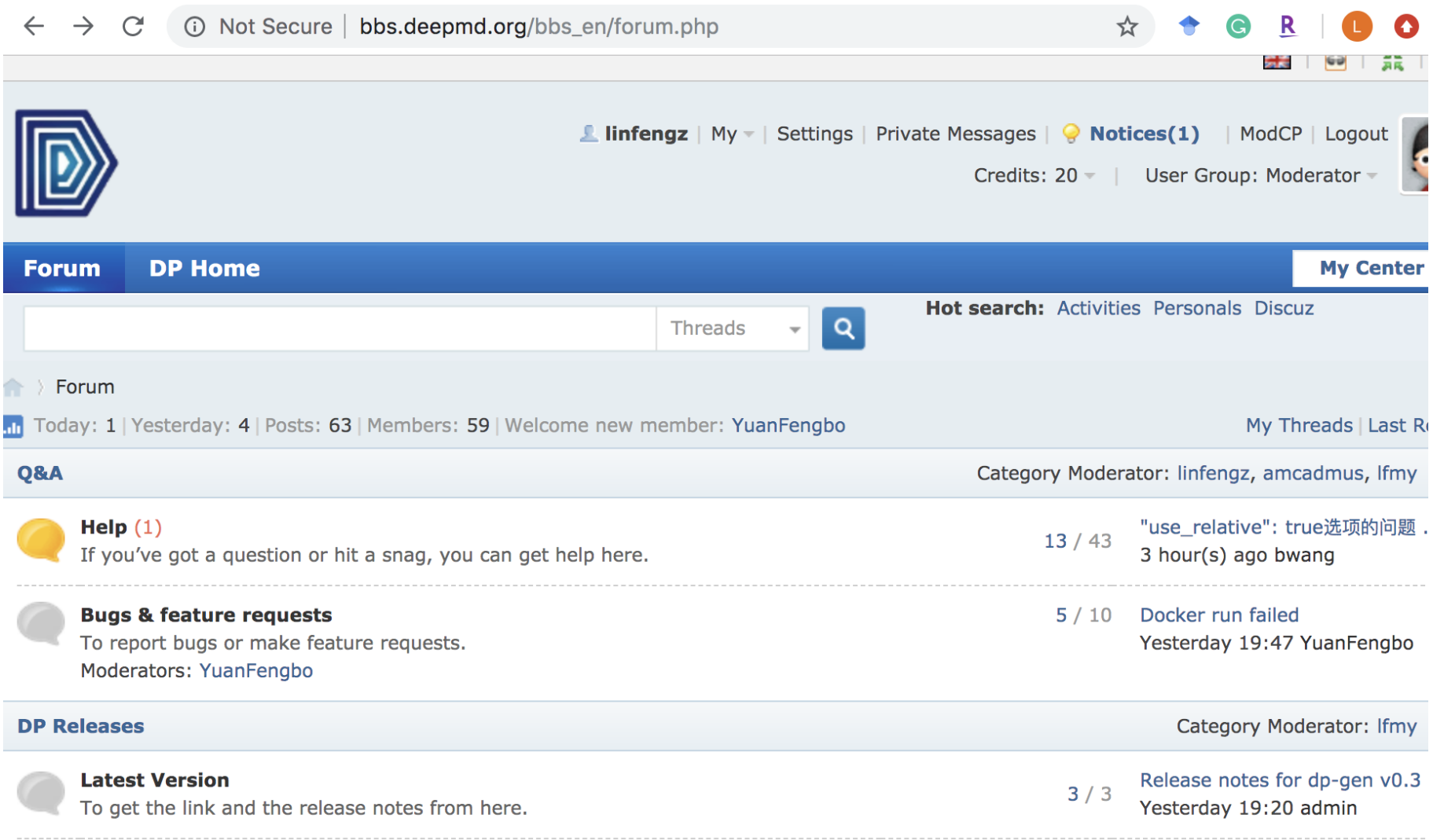






Free download from https://github.com/deepmodeling/dpgen

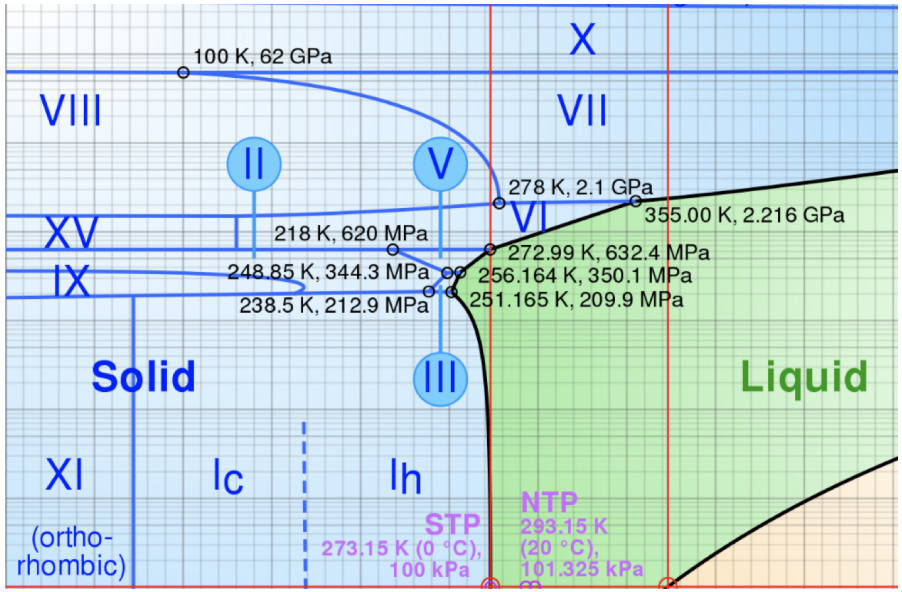
Discussion group



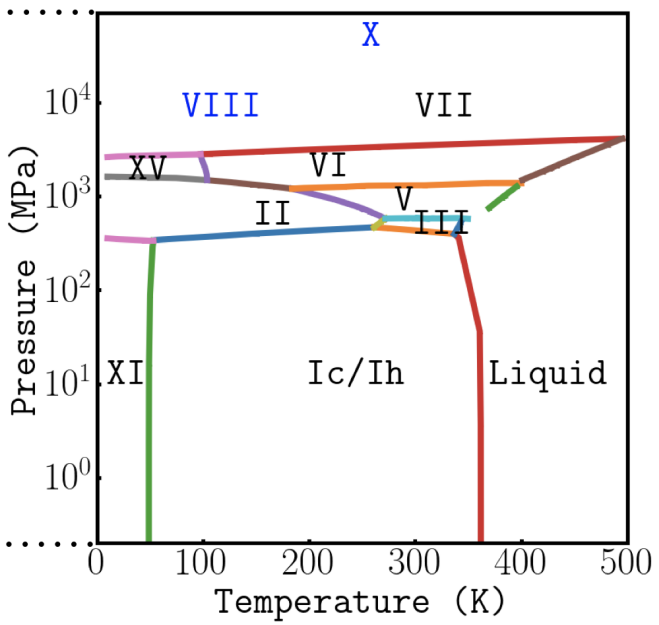
bbs.deepmd.org

Water phase diagram modeled by DP+SCAN

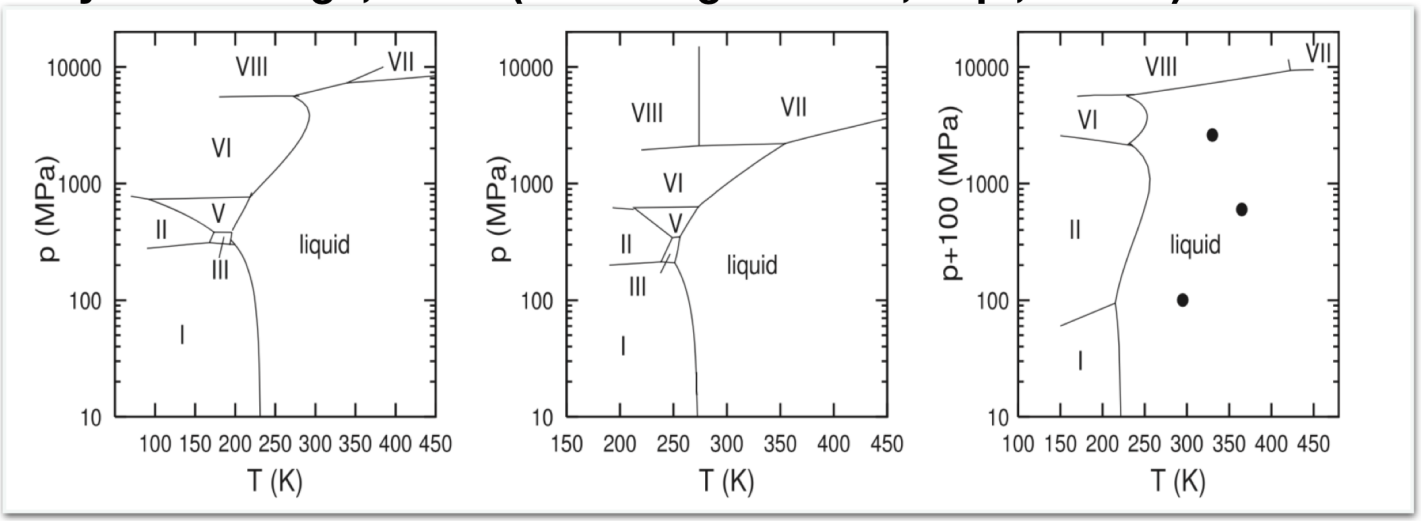
(a) Exp. collected by Martin Chaplin



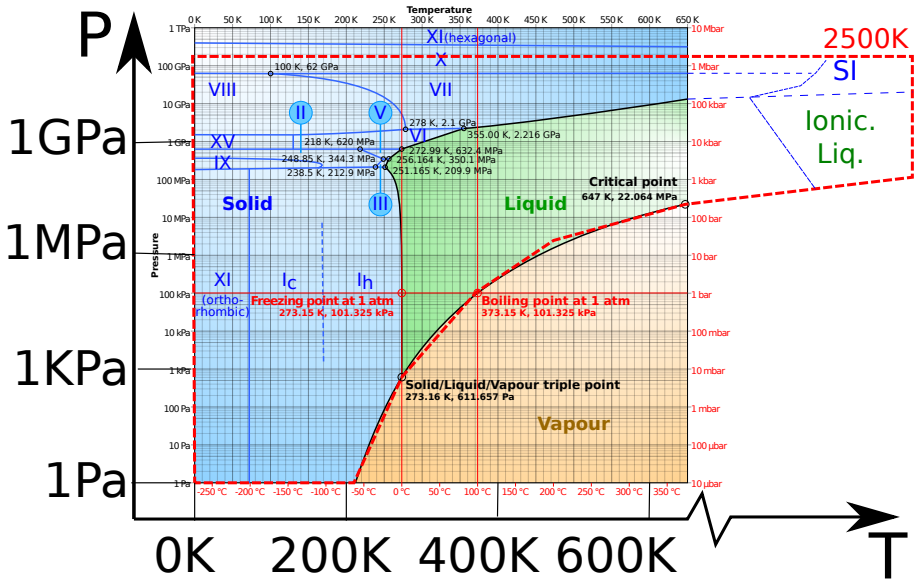
(b) current result

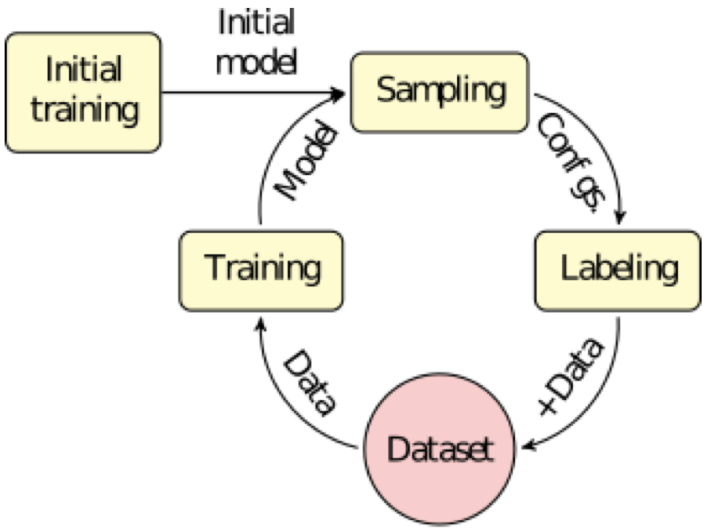


(c) Work by Carlos Vega, et al. (Left to right: TIP4P, Exp., SPC/E)



DP-GEN for water



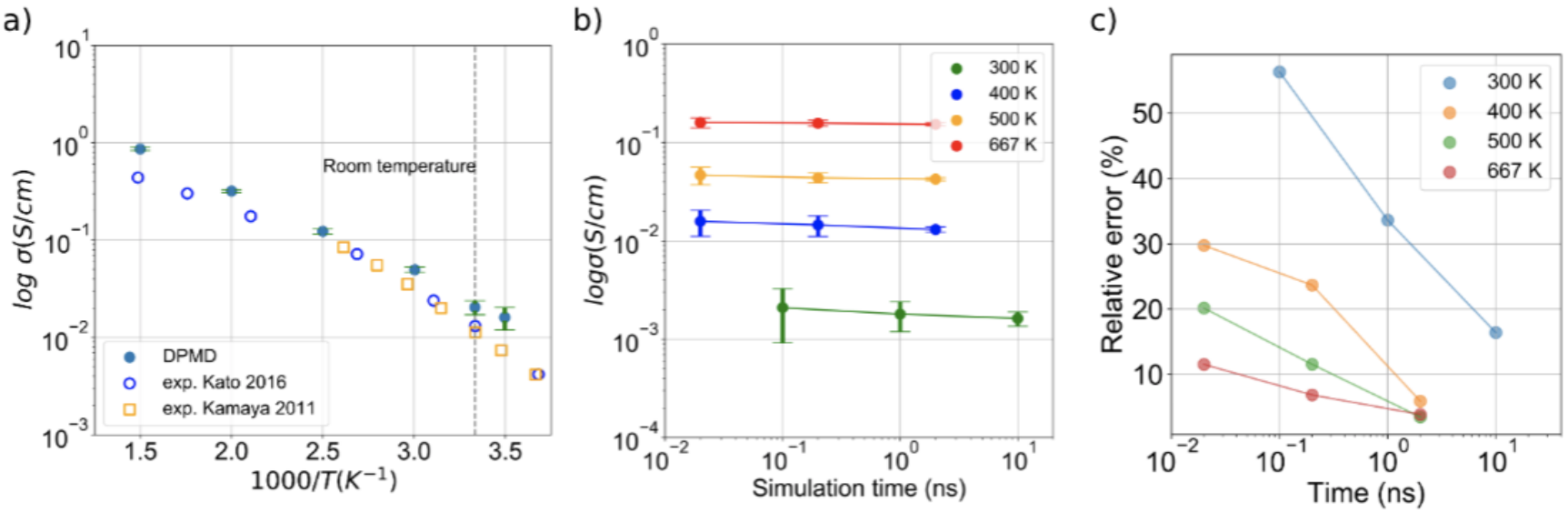


- **Reference model:** DFT at the classical SCAN level;
- **Starting configurations:** relaxed Ice I-XV at $T = 0$ K and equilibrated liquid at $T = 330$ K;
- **Range of thermodynamic conditions:** red dashed box;
- **number of MD snapshots:** DPMD exploration: 1.4 billion, DFT calculation: 32 thousand ($\sim$0.002% of the former).

  Typical AIMD trajectory: 100 thousand snapshots (50-100 *ps*).
- **number of DP-GEN iterations:** 100.

Lithium diffusion in solid-state electrolyte

Ability to handle multi-component systems, here the LiGePS-type systems.



Jun Cheng's group at Xiamen U

1. physical/chemical problems
   - understanding water (phase diagram of water, including reactive regions; phase transition: ice to water, ionic liquid to super-ionic ice; nuclear quantum effect: collective tunneling, isotope effect; reactive event: dissociation and recombination; water surface and water/TiO2 interface; spectra: infra-red; Raman; X-ray Absorption; exotic properties: dielectric constant; density anomaly, etc.
   - physical understanding of different systems that require long-time large-scale simulation with high degrees of model fidelity (high-pressure iron: fractional defect; phase boundary; high-pressure hydrogen: exotic phases)
   - catalysis (Pt cluster on MoS2 surface; CO molecules on gold surface, etc.)
2. materials science problems
   - battery materials (diffusion of lithium in LGePS, LSGeSiPS, etc.; diffusion of Se in Cu2Se alloy)
   - high entropy/high temperature alloy (CoCrFeMnNi alloy; Ni-based alloy)
3. organic chemistry/bio problems
   - crystal structure prediction of molecular crystals;
   - protein-ligand interaction;
   - protein folding.

Outline

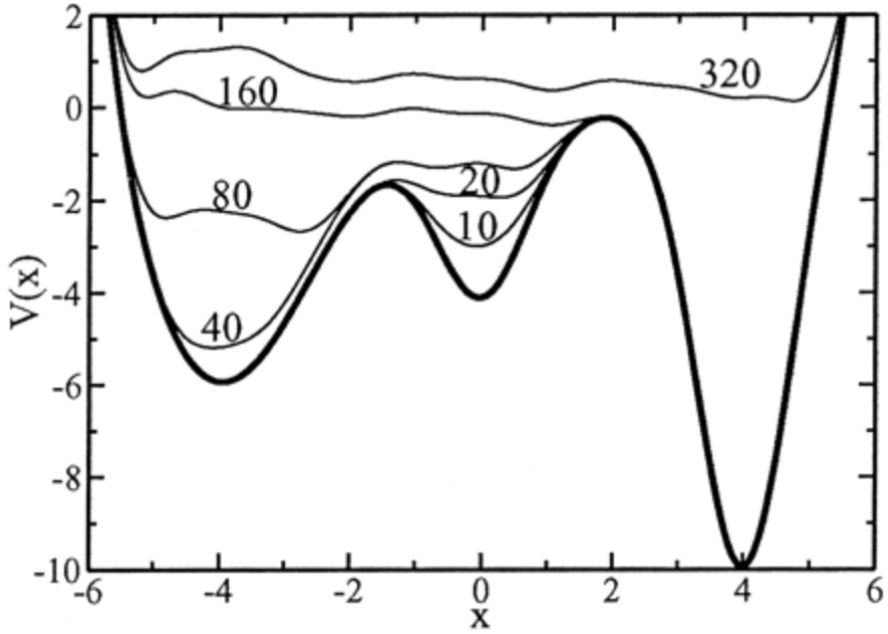
Exploration of very high dimensional configuration spaces

Problems of interest:

- structural optimization (e.g. protein folding)
- free energy calculation
- coarse-grained molecular dynamics

Metadynamics (Laio and Parrinello, 2002)

- free energy is computed by adding up little Gaussians
- free energy is used to help exploration



Very effective when the number of collective variables is small.

Collective variables and free energies

Consider a pre-defined set of collective variables (CV) $\boldsymbol{s}$. Free energy surface (FES) $A(\boldsymbol{s})$:

$$A(\boldsymbol{s}) = -\frac{1}{\beta} \ln p(\boldsymbol{s}), \quad p(\boldsymbol{s}) = \frac{1}{Z} \int e^{-\beta U(\boldsymbol{r})} \delta(\boldsymbol{s}(\boldsymbol{r}) - \boldsymbol{s})\, d\boldsymbol{r},$$

Mean forces

$$\boldsymbol{F}(\boldsymbol{s}) = -\nabla_{\boldsymbol{s}} A(\boldsymbol{s}).$$

Reinforced dynamics

- exploration:
  - biased molecular dynamics:

$$\tilde{\boldsymbol{f}}_i(\boldsymbol{r}) = -\nabla_{\boldsymbol{r}_i} U(\boldsymbol{r}) + \sigma(\boldsymbol{\epsilon}(\boldsymbol{s}(\boldsymbol{r}))) \nabla_{\boldsymbol{r}_i} \boldsymbol{A}(\boldsymbol{s}(\boldsymbol{r}))$$

    σ is an activation function that switches on and off the biasing term
  - decide which ones to label: train an ensemble of networks $\boldsymbol{F}_j, j = 1, \cdots, N$ and compute

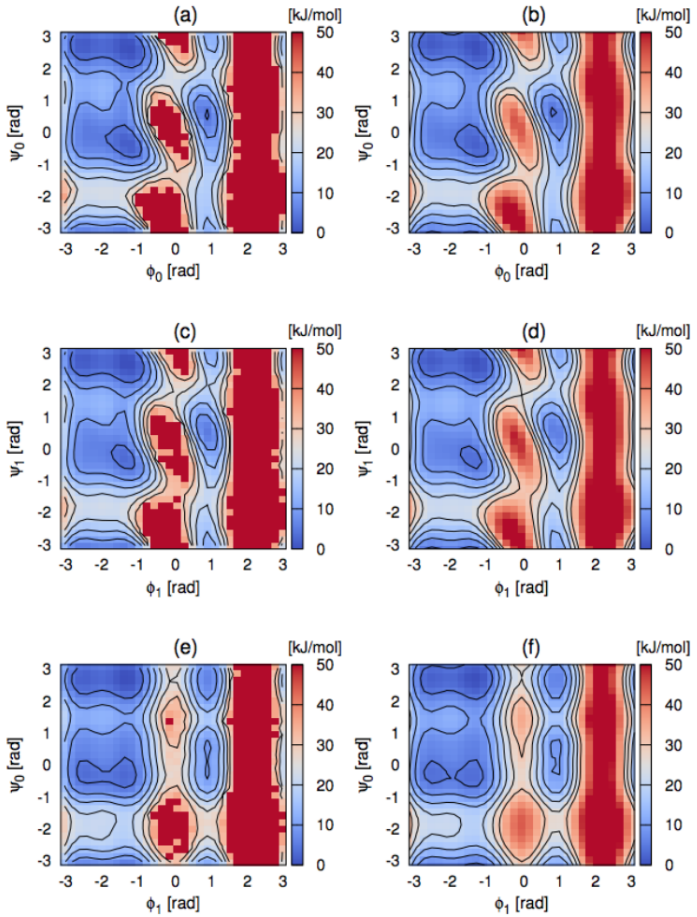
$$\text{variance} = \langle |\boldsymbol{F}_j(\boldsymbol{s}) - \bar{\boldsymbol{F}}(\boldsymbol{s})\|^2 \rangle, \quad \bar{\boldsymbol{F}}(\boldsymbol{s}) = \langle \boldsymbol{F}_j(\boldsymbol{s}) \rangle$$

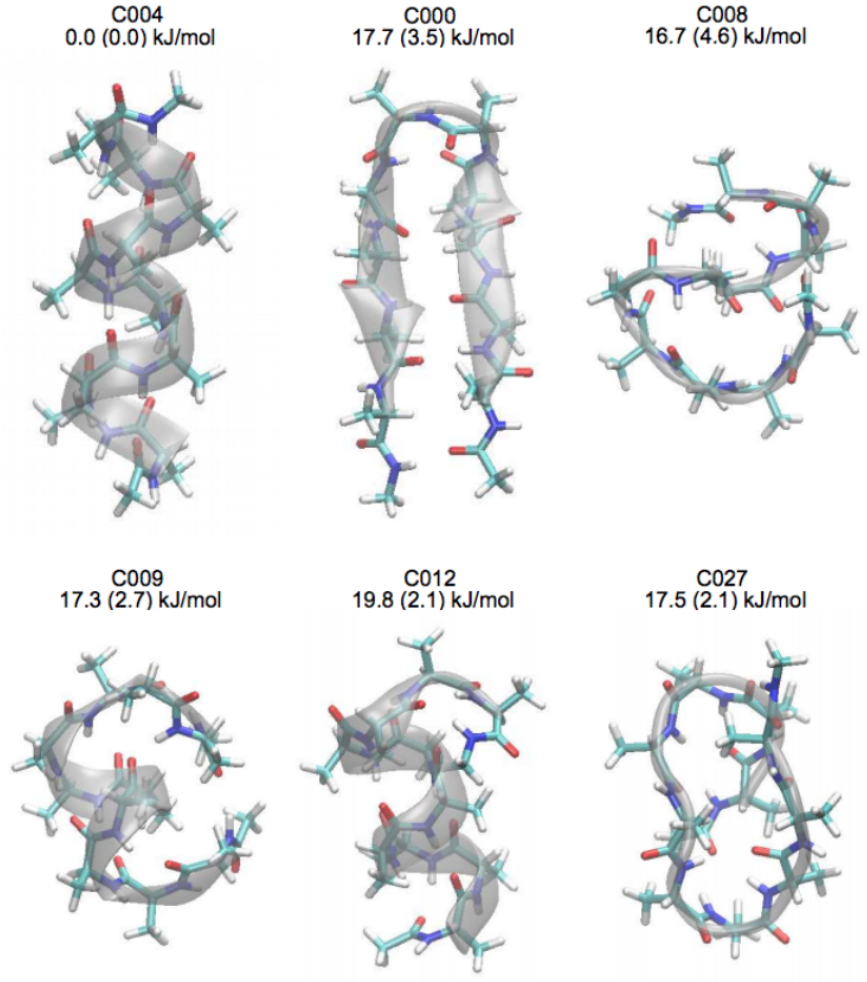
- labeling: compute the mean force (using restrained MD)
- training: deep potential-like coarse-grained model

Can now handle many collective variables

Reinforced dynamics

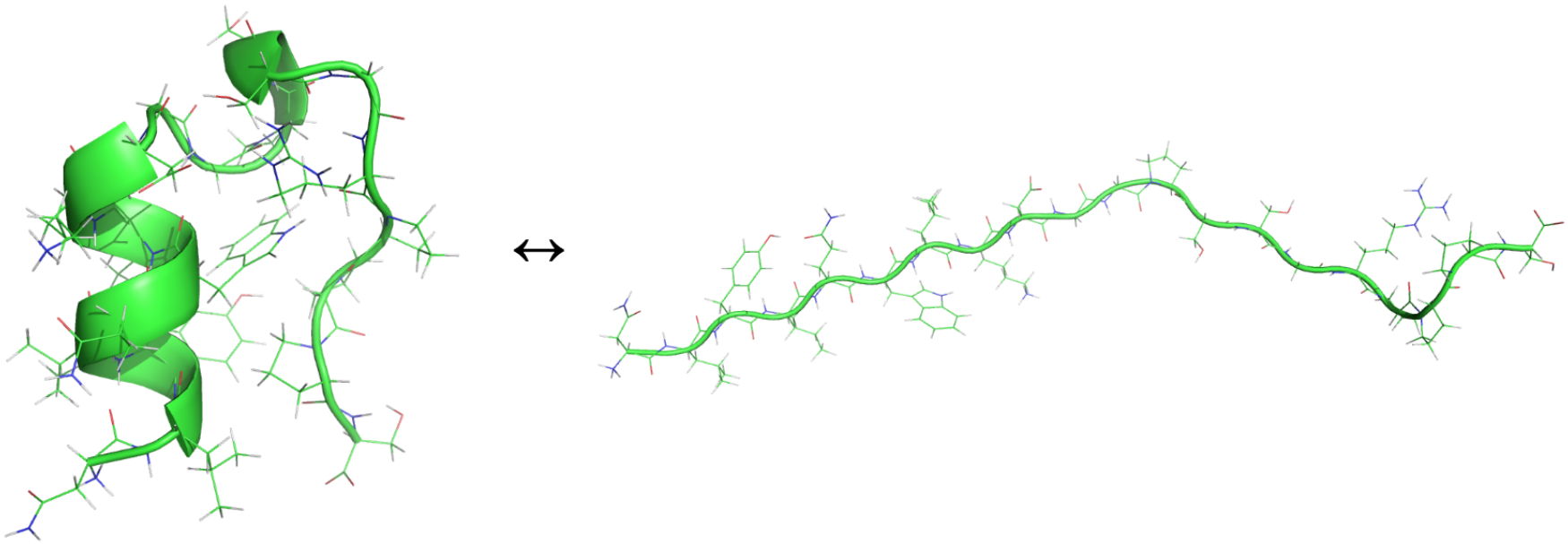
- Left: Tripeptide: brute-force simulation ($\sim$50 μs) $v.s.$ RiD (10 ns biased + 190 ns restrained):
- Right: higher dimensional FES: ala-10 and 20 CVs.





Reinforced dynamics

More recently: Trp-cage folded (20 amino acids, 38 CVs)



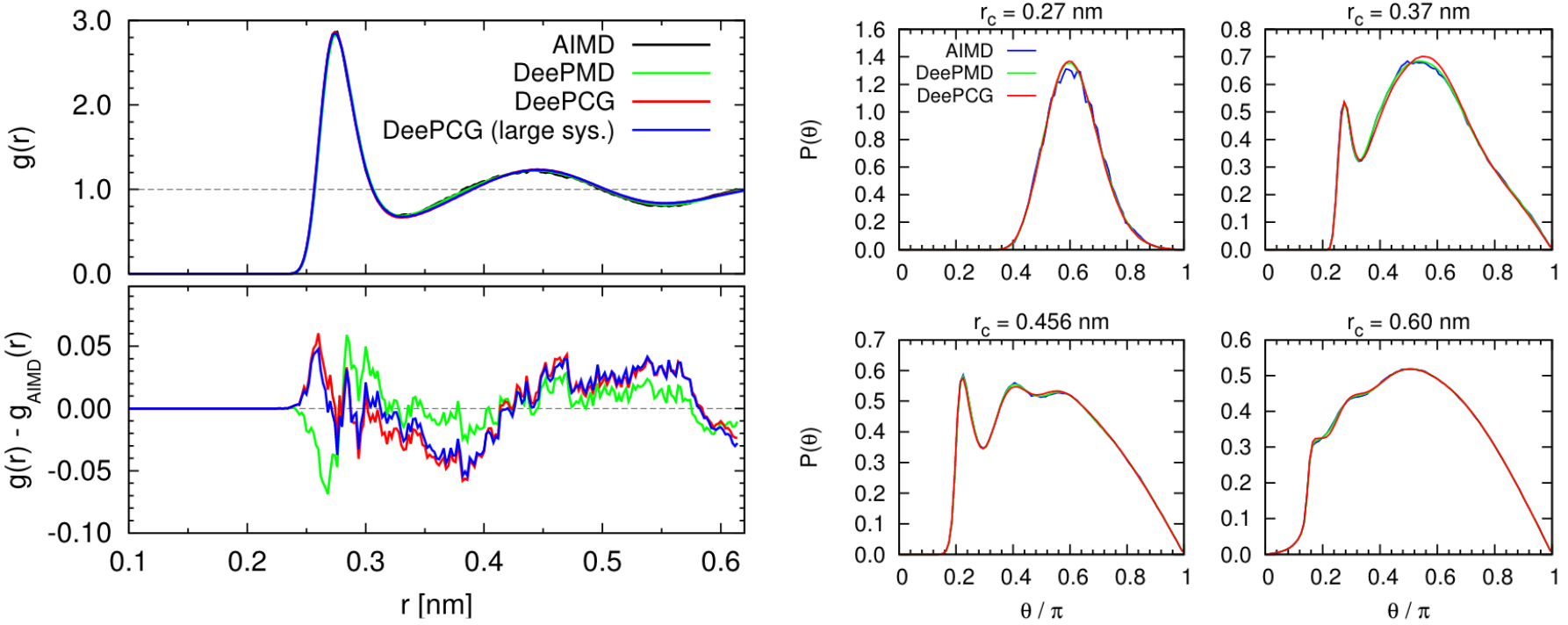
Now we are folding more.

Coarse graining (CG)

Given CG variables $\boldsymbol{s}$. Free energy surface (FES), or CG potential, $A(\boldsymbol{s})$:

$$A(\boldsymbol{s}) = -\frac{1}{\beta}\ln p(\boldsymbol{s}), \quad p(\boldsymbol{s}) = \frac{1}{Z}\int e^{-\beta U(\boldsymbol{r})}\delta(\boldsymbol{s}(\boldsymbol{r}) - \boldsymbol{s})\, d\boldsymbol{r}.$$

Radial distribution functions (left) and angular distribution functions (right):

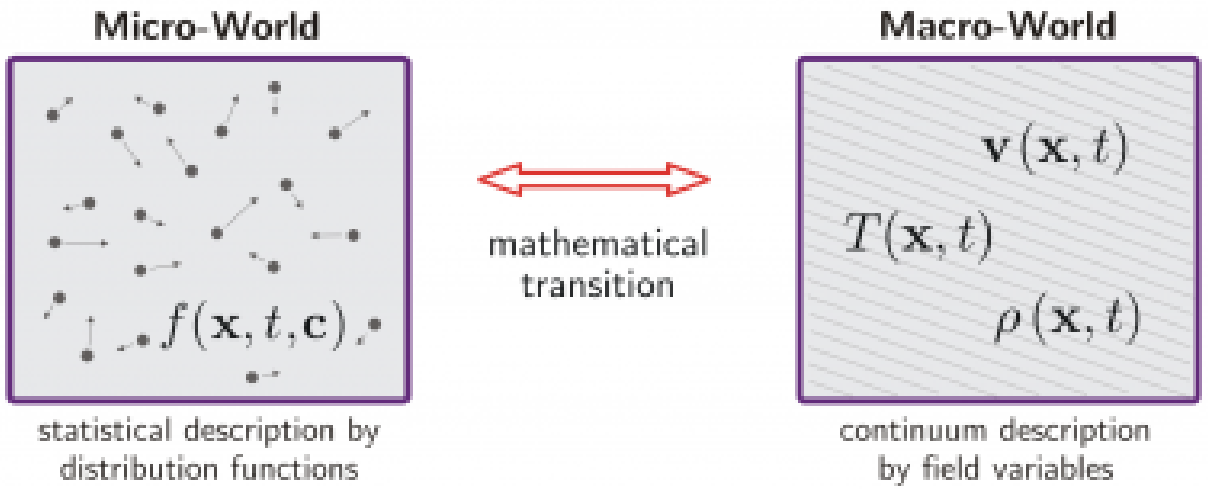


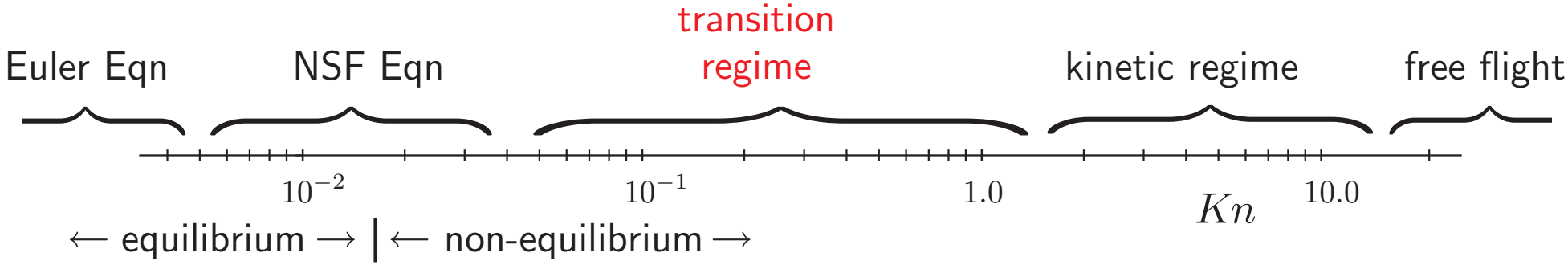
(J. Chem. Phys. 2018, 149(3): 034101.)

Outline

Modeling gas dynamics

$$Kn = \frac{\text{mean free path}}{\text{macroscopic length}}$$





Boltzmann Equation

One-particle density function $f(\boldsymbol{x}, \boldsymbol{v}, t)$

$$\partial_t f + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f = \frac{1}{\varepsilon} Q(f), \quad \boldsymbol{v} \in \mathbb{R}^3, \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^3,$$

$\varepsilon =$ Knudsen number and Q is the collision operator.

Macroscopic state variables: ρ, $\boldsymbol{u}$ and T (density, bulk velocity and temperature)

$$\rho = \int f \, \mathrm{d}\boldsymbol{v}, \quad \boldsymbol{u} = \frac{1}{\rho} \int f \boldsymbol{v} \, \mathrm{d}\boldsymbol{v}, \quad T = \frac{1}{3\rho} \int f |\boldsymbol{v} - \boldsymbol{u}|^2 \, \mathrm{d}\boldsymbol{v}.$$

When $\varepsilon \ll 1$, Boltzmann can be approximated by Euler:

$$\partial_t \boldsymbol{U} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F}(\boldsymbol{U}) = 0,$$

with $p = \rho T$, $E = \frac{1}{2}\rho \boldsymbol{u}^2 + \frac{3}{2}\rho T$,

$$\boldsymbol{U} = (\rho, \rho \boldsymbol{u}, E)^T$$

$$\boldsymbol{F}(\boldsymbol{U}) = (\rho \boldsymbol{u}, \rho \boldsymbol{u} \otimes \boldsymbol{u} + pI, (E + p)\boldsymbol{u})^T$$

Conventional Moment Method

Proceed in 3 steps:

1. Start with the choice of a finite-dimensional linear subspace of functions of $\boldsymbol{v}$ (usually to be polynomials, e.g., Hermite polynomials).
2. Expand $f(\boldsymbol{x}, \boldsymbol{v}, t)$ using these functions as bases and take the coefficients as moments (including macroscopic variables ρ, $\boldsymbol{u}$, T, etc.).
3. Finally close the system with simplified assumptions, e.g., truncating moments of higher orders

$$\begin{cases} \partial_t \boldsymbol{U} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F}(\boldsymbol{U}, \boldsymbol{W}) = 0, \\ \partial_t \boldsymbol{W} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{G}(\boldsymbol{U}, \boldsymbol{W}) = \dfrac{1}{\varepsilon} \mathbb{R}(\boldsymbol{U}, \boldsymbol{W}). \end{cases}$$

For instance, in Grad 13-moment system, $(\boldsymbol{U}, \boldsymbol{W})$ is constructed based on the moments of the bases $\{1, \boldsymbol{v}, (\boldsymbol{v} - \boldsymbol{u}) \otimes (\boldsymbol{v} - \boldsymbol{u}), |\boldsymbol{v} - \boldsymbol{u}|^2 (\boldsymbol{v} - \boldsymbol{u})\}$.

Machine learning-based moment method

Objective: construct an uniformly accurate (generalized) moment model using machine learning.

1: Learn the Moments through Autoencoder
Find an encoder Ψ that maps $f(\cdot, \boldsymbol{v})$ to generalized moments $\boldsymbol{W} \in \mathbb{R}^M$ and a decoder Φ that recovers the original f from $\boldsymbol{U}, \boldsymbol{W}$

$$\boldsymbol{W} = \Psi(f) = \int \mathbf{w} f \, \mathrm{d}\boldsymbol{v}, \quad \Phi(\boldsymbol{U}, \boldsymbol{W})(\boldsymbol{v}) = h(\boldsymbol{v}; \boldsymbol{U}, \boldsymbol{W}).$$

The goal is essentially to find optimal $\mathbf{w}$ and h parametrized by neural networks through minimizing

$$\mathop{\mathbb{E}}_{f \sim \mathcal{D}} \|f - \Phi(\Psi(f))\|^2 + \lambda_\eta (\eta(f) - h_\eta(\boldsymbol{U}, \boldsymbol{W}))^2.$$

$\eta(f)$ denotes entropy.

2: Learn the Fluxes and Source Terms in the PDE

Recall the general conservative form of the moment system

$$\begin{cases} \partial_t \boldsymbol{U} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon) = 0, \\ \partial_t \boldsymbol{W} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{G}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon) = \mathbb{R}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon). \end{cases}$$

Rewrite it into (variance reduction)

$$\begin{cases} \partial_t \boldsymbol{U} + \nabla_{\boldsymbol{x}} \cdot [\boldsymbol{F}_0(\boldsymbol{U}) + \tilde{\boldsymbol{F}}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon)] = 0, \\ \partial_t \boldsymbol{W} + \nabla_{\boldsymbol{x}} \cdot [\boldsymbol{G}_0(\boldsymbol{U}) + \tilde{\boldsymbol{G}}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon)] = \mathbb{R}(\boldsymbol{U}, \boldsymbol{W}; \varepsilon). \end{cases}$$

$\boldsymbol{F}_0(\boldsymbol{U}), \boldsymbol{G}_0(\boldsymbol{U})$ are the fluxes of the moments $\boldsymbol{U}, \boldsymbol{W}$ under the Maxwellian distribution.

Our goal is to obtain ML models for $\tilde{\boldsymbol{F}}, \tilde{\boldsymbol{G}}, \mathbb{R}$ from the original kinetic equation.

Issues: (1) physical symmetries (e.g. Galilean invariance); (2) data generation (active learning algorithm); (3) locality vs. non-locality of the model

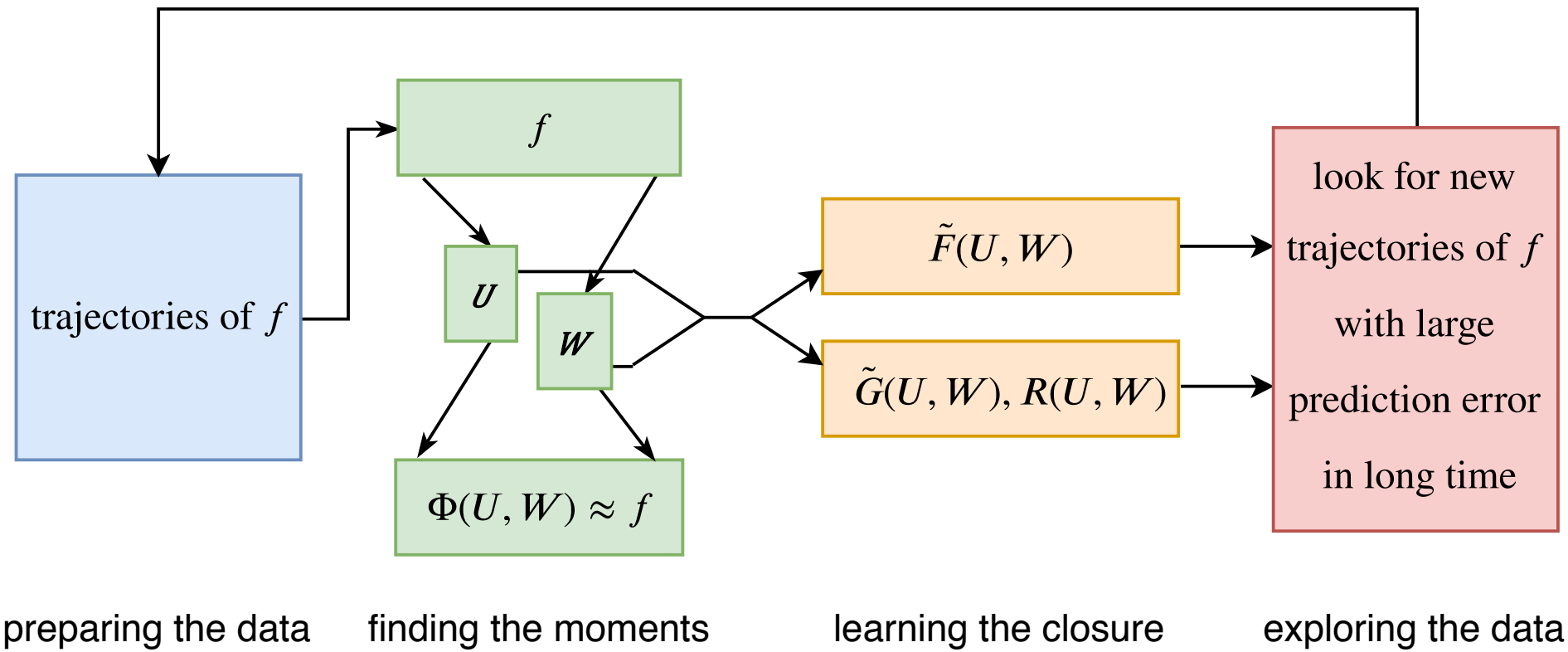


Figure: Schematic diagram of the machine learning-based moment method

- exploration: random initial conditions made up of waves and discontinuities
- labeling: solving kinetic equation (Boltzmann equation for Maxwell molecules)
- training: Galilean invariance

Galilean Invariant Moments

Galilean invariance of the Boltzmann equation:

$$f'(\boldsymbol{x}, \boldsymbol{u}, t) = f(\boldsymbol{x} - t\boldsymbol{u}', \boldsymbol{v} - \boldsymbol{u}', t).$$

Moments:

$$\boldsymbol{W}_{\mathsf{Gal}} = \Psi(f) = \int f(\boldsymbol{v}) \mathbf{w}\left(\frac{\boldsymbol{v} - \boldsymbol{u}}{\sqrt{T}}\right) \mathrm{d}\boldsymbol{v}.$$

Closure:

$$\partial_t \int_{\mathbb{R}^D} f(\boldsymbol{v}) \mathbf{w}\left(\frac{\boldsymbol{v} - \boldsymbol{u}_j}{\sqrt{T_j}}\right) \mathrm{d}\boldsymbol{v} + \nabla_{\boldsymbol{x}} \cdot \int_{\mathbb{R}^D} f(\boldsymbol{v}) \mathbf{w}\left(\frac{\boldsymbol{v} - \boldsymbol{u}_j}{\sqrt{T_j}}\right) \boldsymbol{v}^{\mathrm{T}} \mathrm{d}\boldsymbol{v} = \int_{\mathbb{R}^D} \frac{1}{\varepsilon} Q(f) \mathbf{w}\left(\frac{\boldsymbol{v} - \boldsymbol{u}_j}{\sqrt{T_j}}\right) \mathrm{d}$$

$$\partial_t \boldsymbol{W}_{\mathsf{Gal}} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{G}_{\mathsf{Gal}}(\boldsymbol{U}, \boldsymbol{W}_{\mathsf{Gal}}; \boldsymbol{U}_j) = \frac{1}{\varepsilon} \mathbb{R}_{\mathsf{Gal}}(\boldsymbol{U}, \boldsymbol{W}_{\mathsf{Gal}}).$$

The data efficiency is better than the previous one since it learns the dynamical system more intrinsically.

$\varepsilon \sim$ Log10-Uniform$(-3, 1)$, constant across the domain; initial profiles consist of a combination of a few sin waves and shocks.

Size of dataset array: $200 \times 100 \times 48 \times 48 \times 100$. Specify $\boldsymbol{W} \in \mathbb{R}^9$.

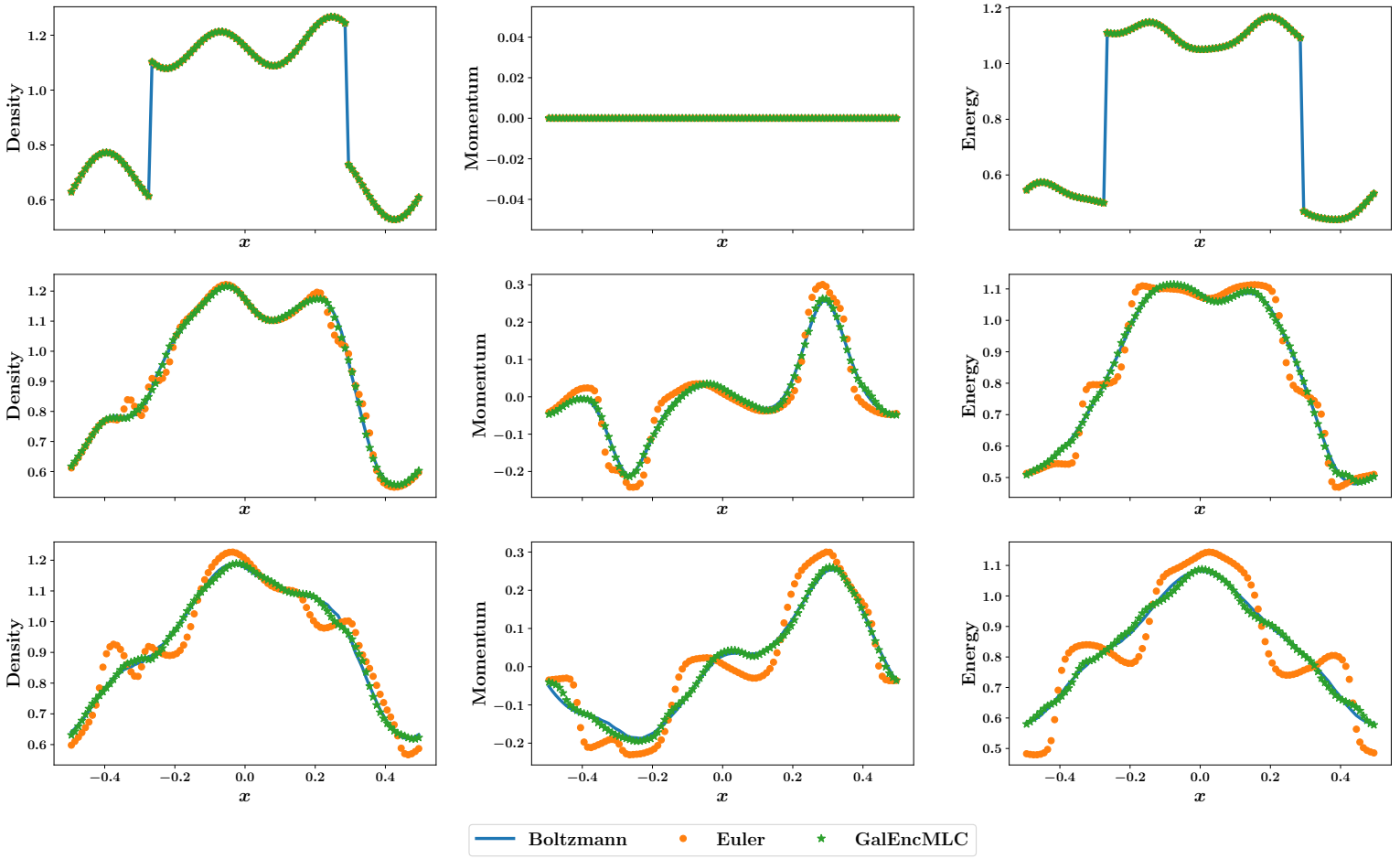


Figure: Sample profiles of $\rho, \rho\boldsymbol{u}, E$ (from left to right) at $t = 0, 0.05, 0.1$ (from top to bottom), $\varepsilon = 8.10$

ε varies from 10^{-3} to 10 in the domain; initial profiles are the same as before

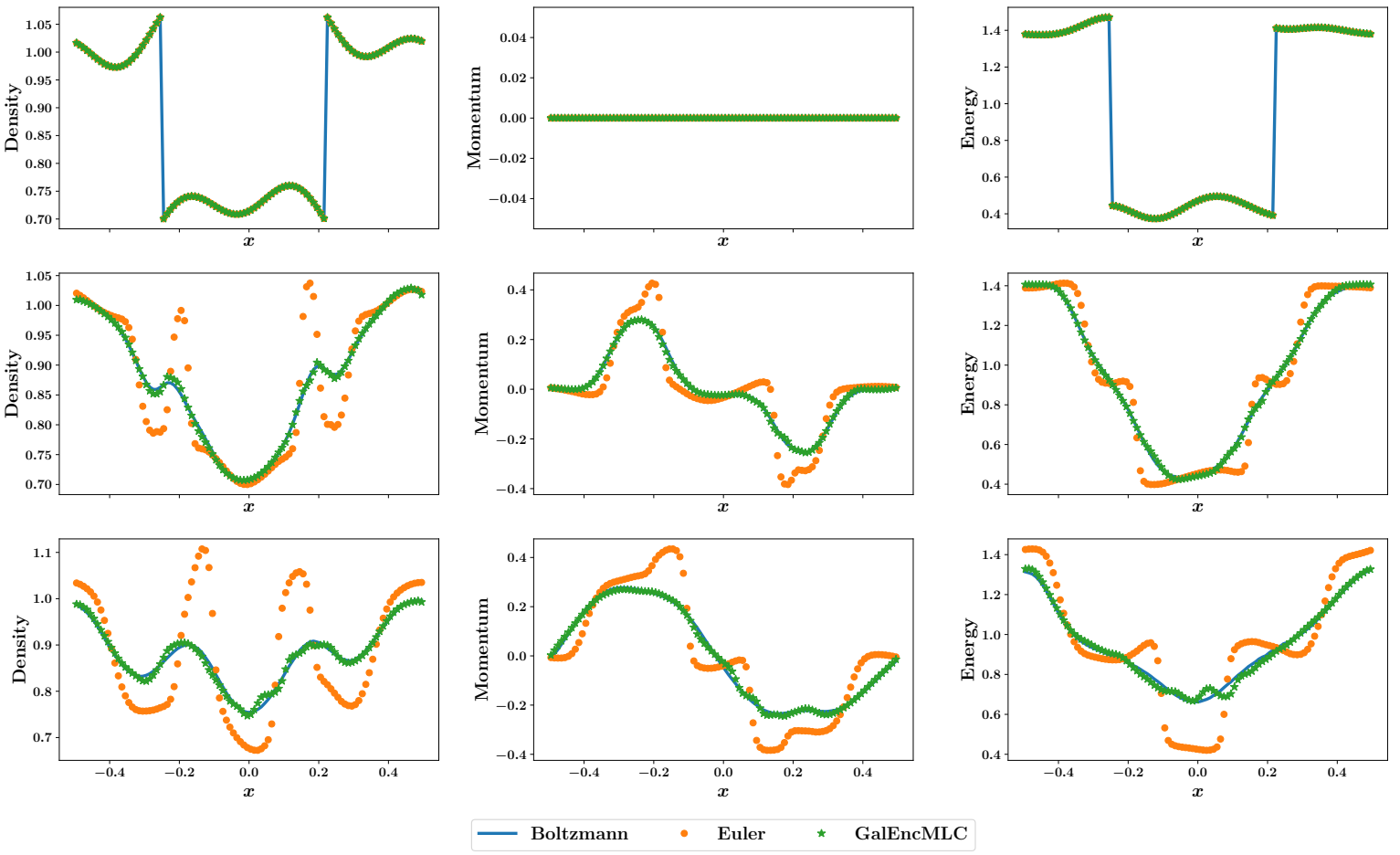
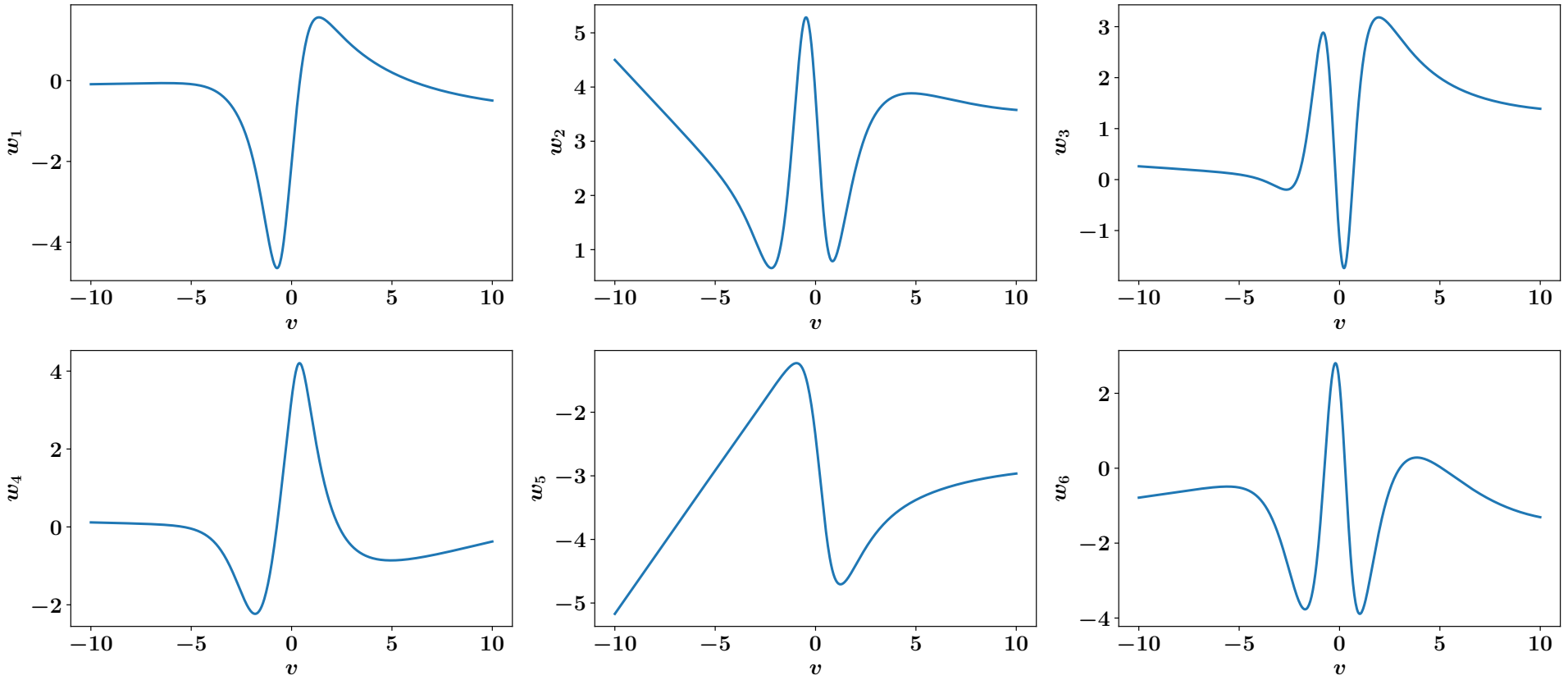


Figure: Profiles of $\rho, \rho\boldsymbol{u}, E$ (from left to right) at $t = 0, 0.05, 0.1$ (from top to bottom)

Numerical results

Learned functions $\mathbf{w}(v)$ as generalized moments



Other possible applications of concurrent learning

- solving PDEs
- model-based reinforcement learning
- control

Essentially any time we have a code (or a simulator) to generate the states and the data.

Outline

Concluding remarks

- Concurrent learning is a very powerful tool for multi-scale modeling.
  - DPMD allows us to perform MD simulation of large systems with quantum accuracy
  - RiD allows us to compute free energy function with many collective variables
  - ML based moment closure allows to obtain hydrodynamic models for the Boltzmann equation for Maxwell molecules that are uniformly accurate for a wide range of Knudsen numbers
- These are models, not just algorithms (sequential multi-scale modeling paradigm)
- The methodologies are quite general
- It is important to take into account symmetries and other physical constraints
- New paradigm for multi-scale models: analogy to Euler's equations for complex gases

MSML 2020

A new annual conference:
Mathematical and Scientific Machine Learning (MSML)

First meeting:

- Program Chairman: Jianfeng Lu (Duke) and Rachel Ward (Univ Texas/Austin)
- Time: July 15-17, 2020
- Location: Princeton
- Submission deadline: November 30, 2019
- website: http://msml-conf.org