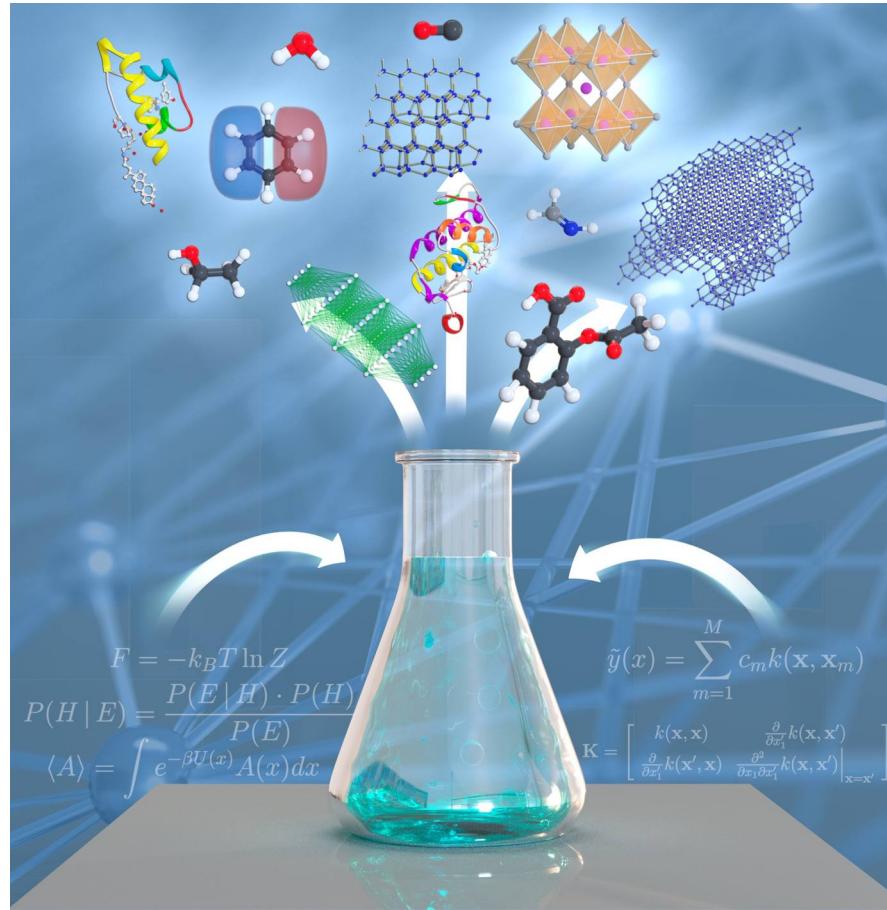


Machine Learning for Molecular Physics

Cecilia Clementi

Freie Universität Berlin
cecilia.clementi@fu-berlin.de



“The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation.”

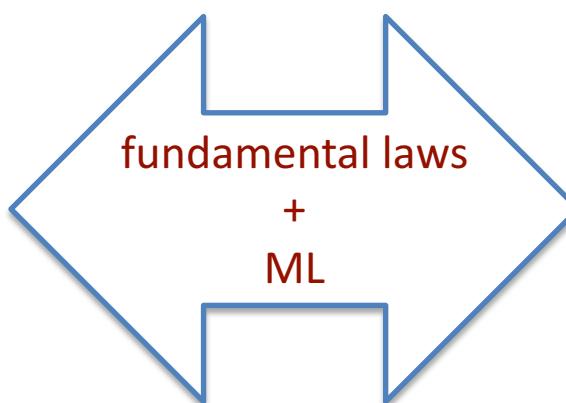
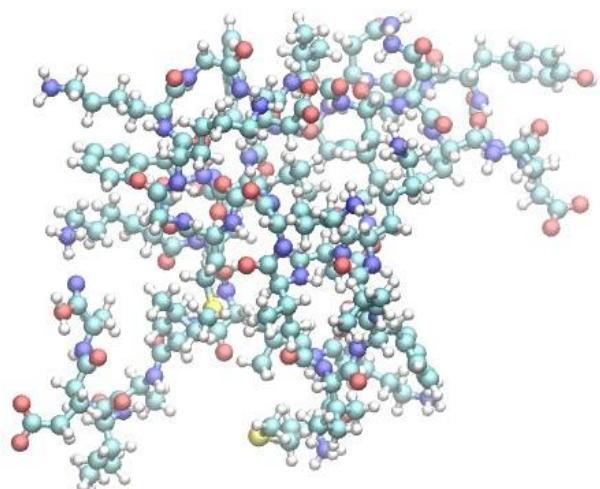
Paul Dirac, 1929

Machine Learning aims at extracting complex patterns and relationships from large data sets, to predict specific properties of the data

How can we use it in Physics and Chemistry?

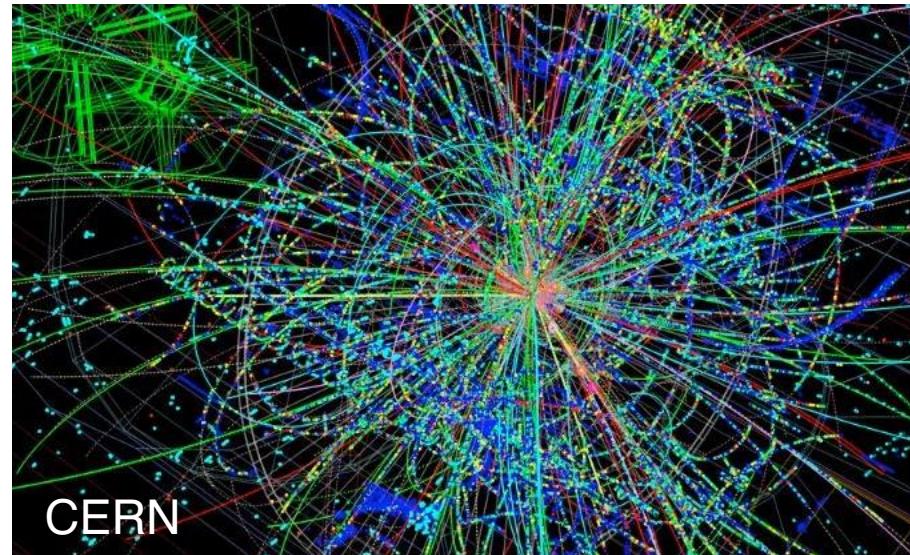


classification in
10 different digits



physical
and
chemical
properties

Looking for patterns in structures and trends in data is one of the defining features of physical sciences



This has been traditionally done using background knowledge from experiments, intuition and physics-based derivations

ML now allows to extend the traditional methods much further and tackle problems that were not accessible before

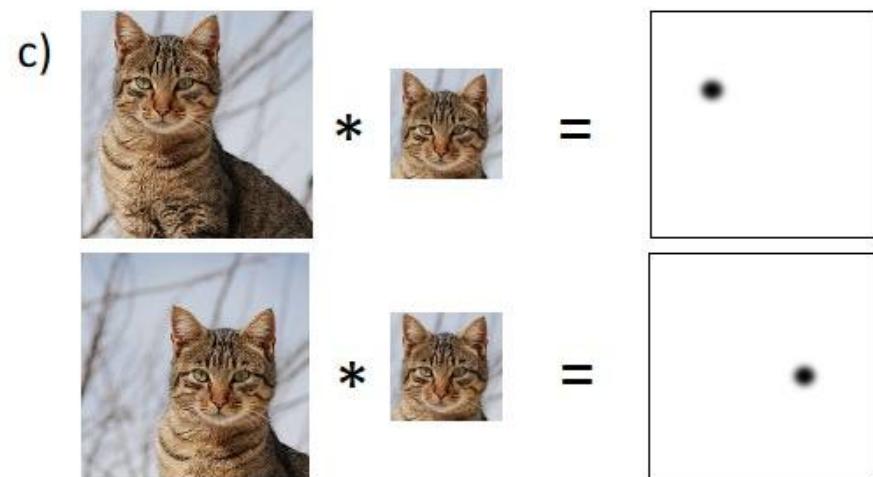
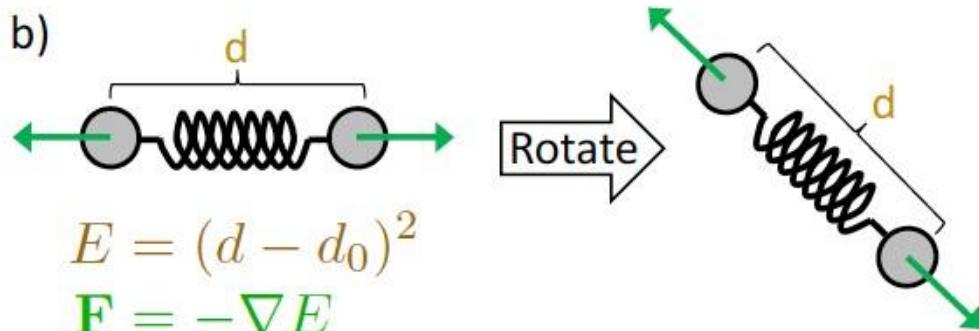
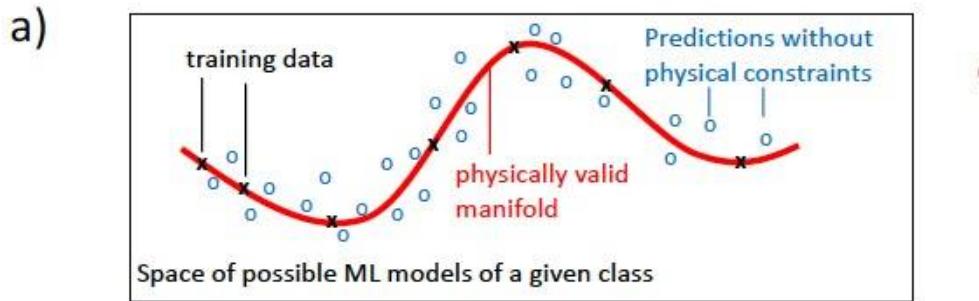
Incorporating Physics into Machine Learning

E.g., when dealing with molecular energies:

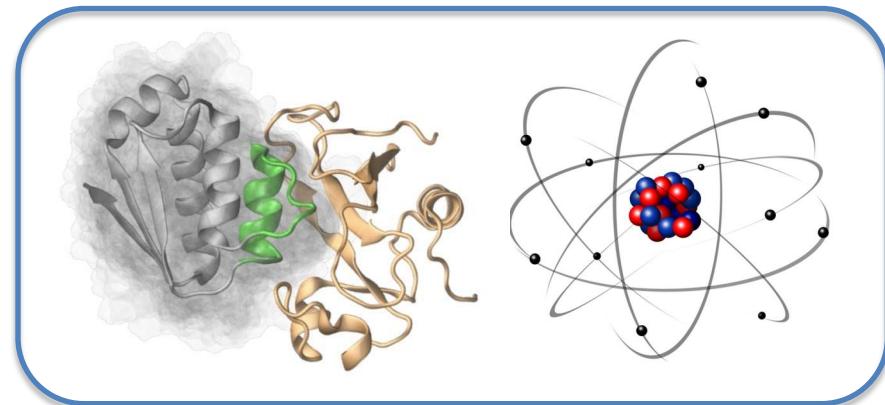
- Energy invariance upon translation and rotation (and forces equivariance)
- Energy conservation
- Indistinguishability of identical particles

Additionally:

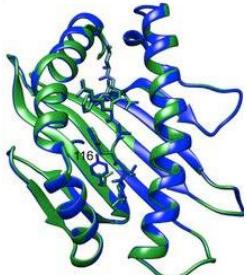
- Probability conservation and stochasticity
- Detailed balance



Molecular Dynamics and Quantum Mechanics



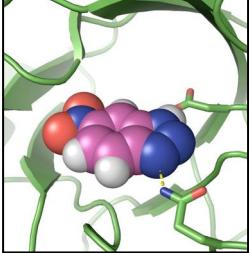
Molecular Dynamics and Quantum Mechanics



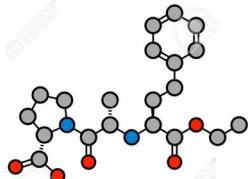
Vaccines



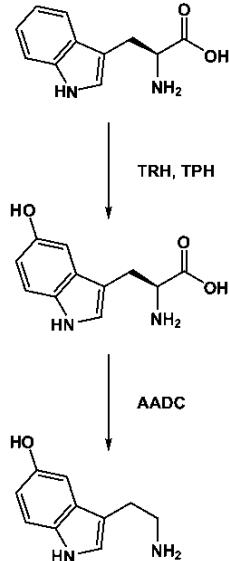
Antibodies



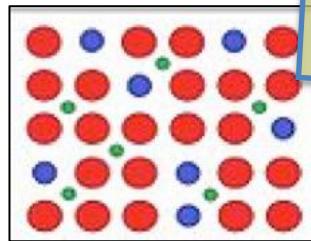
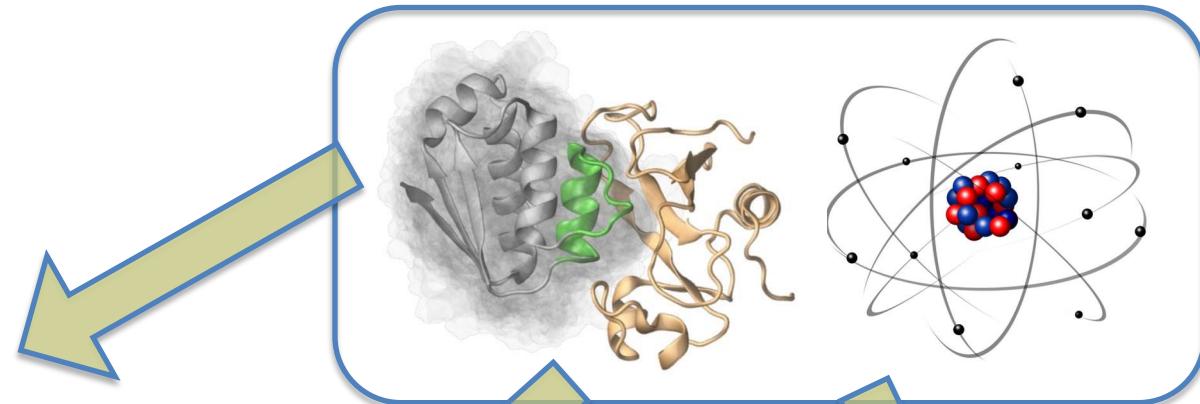
Enzymes



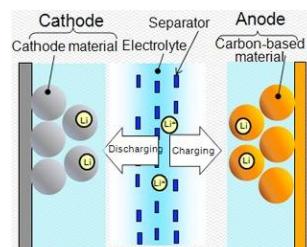
Drugs



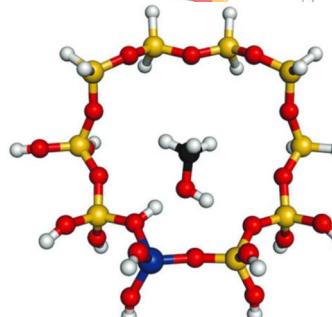
Synthesis



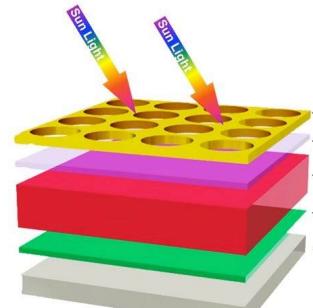
Alloys



Batteries

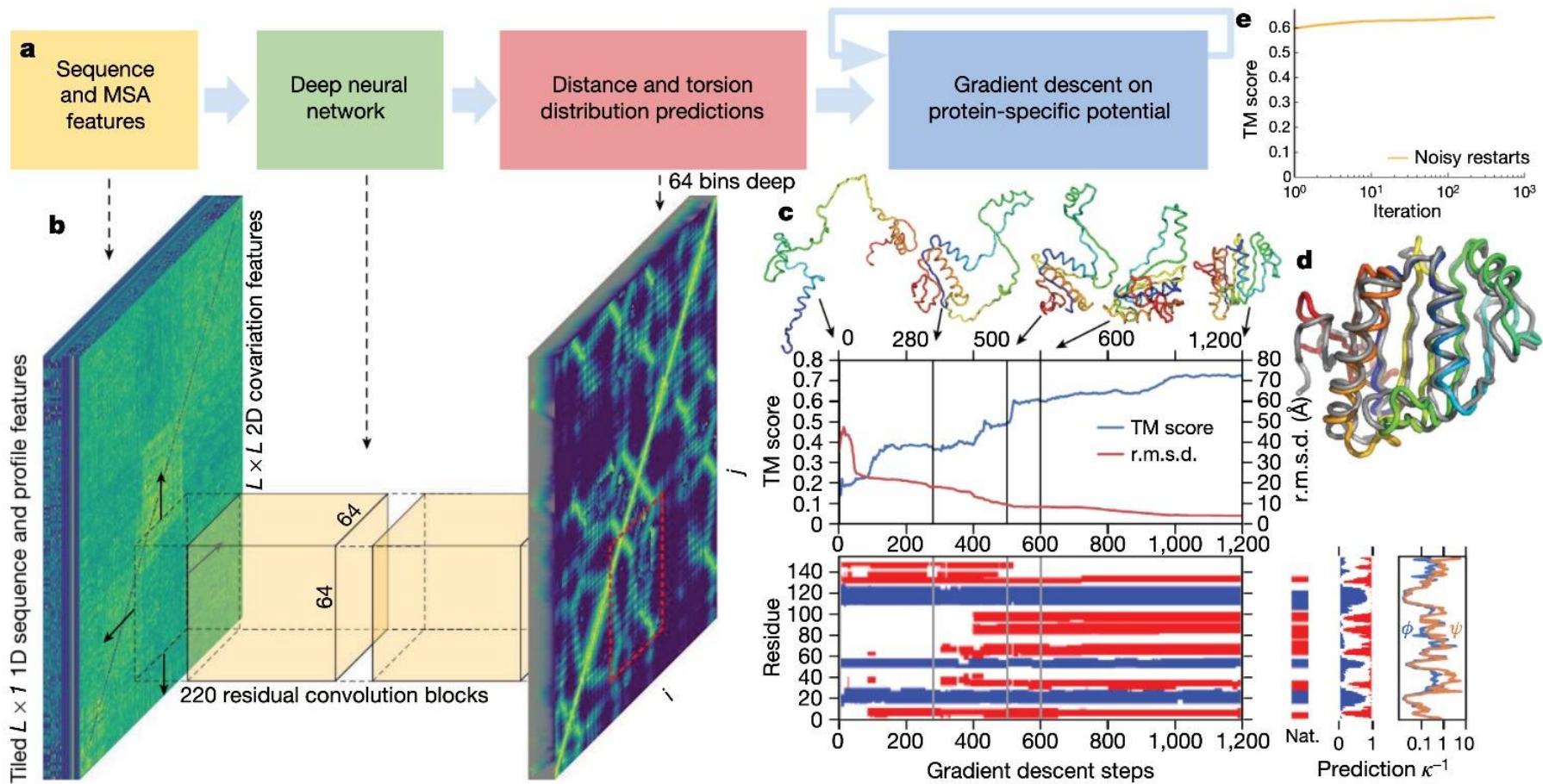


Zeolites/MOFs



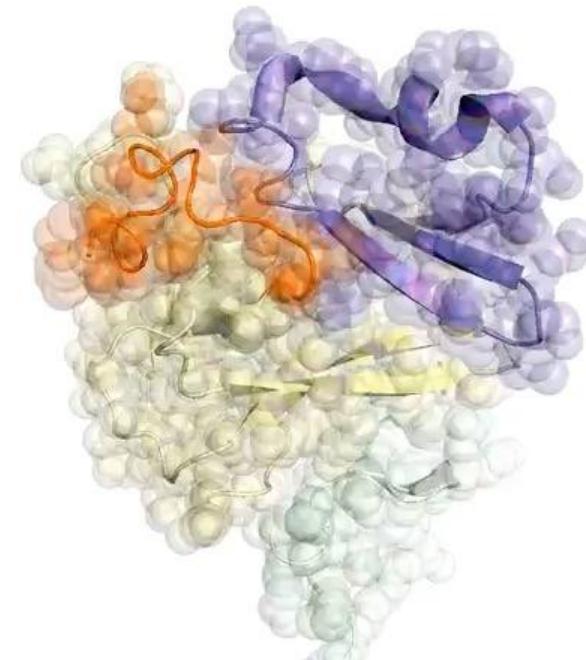
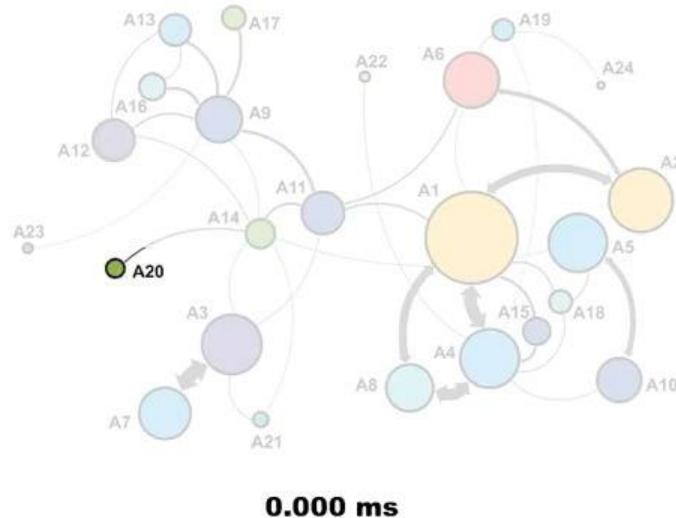
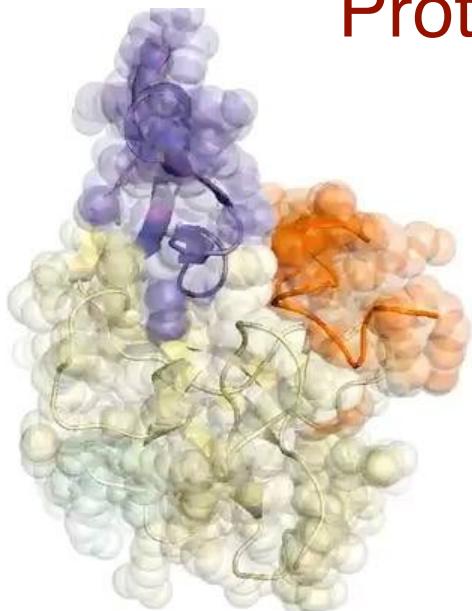
Solar Cells

DeepMind AlphaFold



Senior, A.W., Evans, R., Jumper, J. et al. Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020)

Protein function requires Dynamics



Chen, S., Wiewiora, R.P., Meng, F. et al. The dynamic conformational landscape of the protein methyltransferase SETD8. *eLife* 8, e45403 (2019)

Equilibrium expectation value
(e.g. binding affinity)

$$\langle A \rangle = \int_{\mathbf{x}} \mu(\mathbf{x}) a(\mathbf{x}) d\mathbf{x}$$

Equilibrium density
defined by potential +
thermodynamic constraints

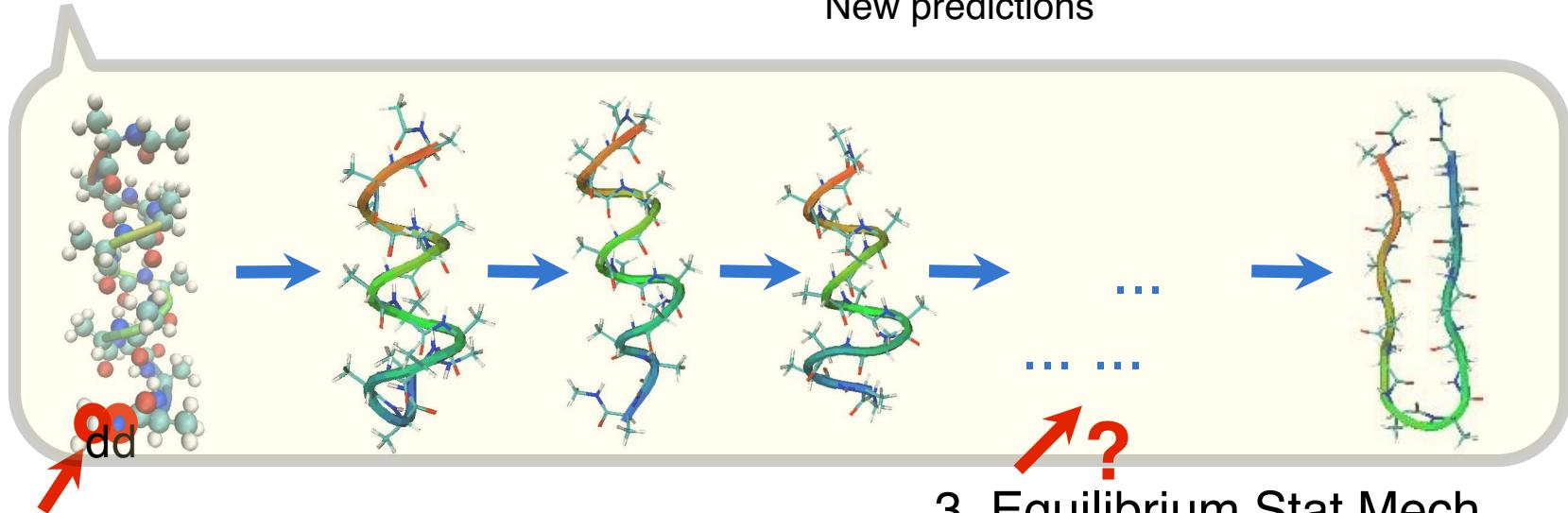
$$\mu(\mathbf{x}) \propto e^{-u(\mathbf{x})}$$

Three aspects of molecular simulations

Trajectories in equilibrium distribution



Observables to compare with the experiment
Mechanism from the dynamics
New predictions



1. Force Field

A set of parameters and equations describing the interactions between atoms



2. Kinetics

How do we understand the molecular mechanisms from the data?

3. Equilibrium Stat Mech

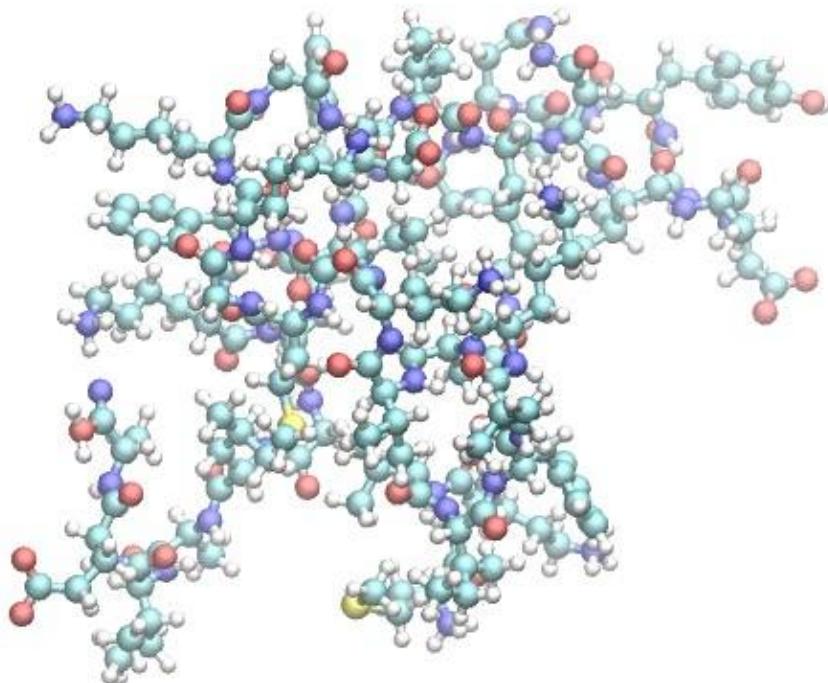
Can the simulations cover the rare events we are interested in?

How we have been doing Molecular Simulation before ML

1. Force fields

$$\mathcal{H}\psi_{\mathbf{x}}(\mathbf{r}, \mathbf{s}) = U(\mathbf{x})\psi_{\mathbf{x}}(\mathbf{r}, \mathbf{s})$$

Born-Oppenheimer approximation



Use (approximate)
quantum mechanics
to calculate energy
in specific configurations
and then fit to
classical potentials

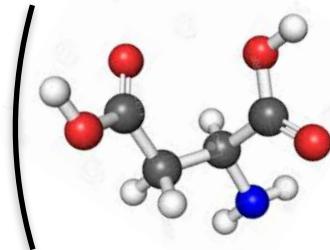
Force fields How can we do it with ML

$$\hat{H}\psi_R(\mathbf{r}_1, \dots, \mathbf{r}_N) = E_R\psi_R(\mathbf{r}_1, \dots, \mathbf{r}_N)$$

1. Highly accurate solution
for small molecules

Quantum Chemistry
(DFT, Coupled-cluster, ...)

Database



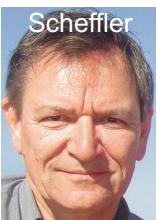
Chemical Elements
Atom coordinates \mathbf{R}
Energy
Forces

$E(\mathbf{R})$
 $-dE(\mathbf{R}) / d\mathbf{R}_i$

, . . .

2. Predict for new molecule
(Elements and atom coordinates \mathbf{R})

Machine learning (supervised)

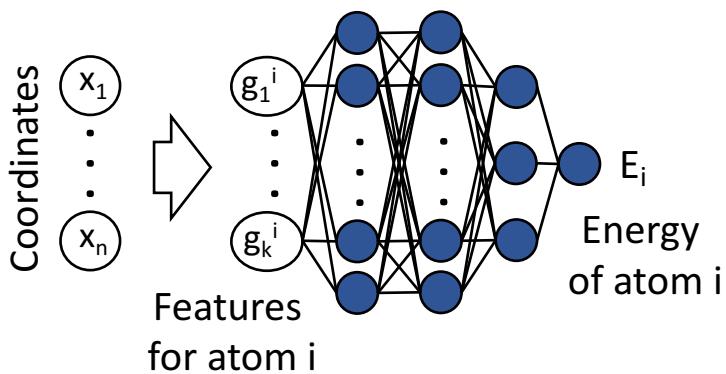


Energy $E(\mathbf{R})$
Forces $-dE(\mathbf{R}) / d\mathbf{R}_i$



Force fields How can we do it with ML

a)

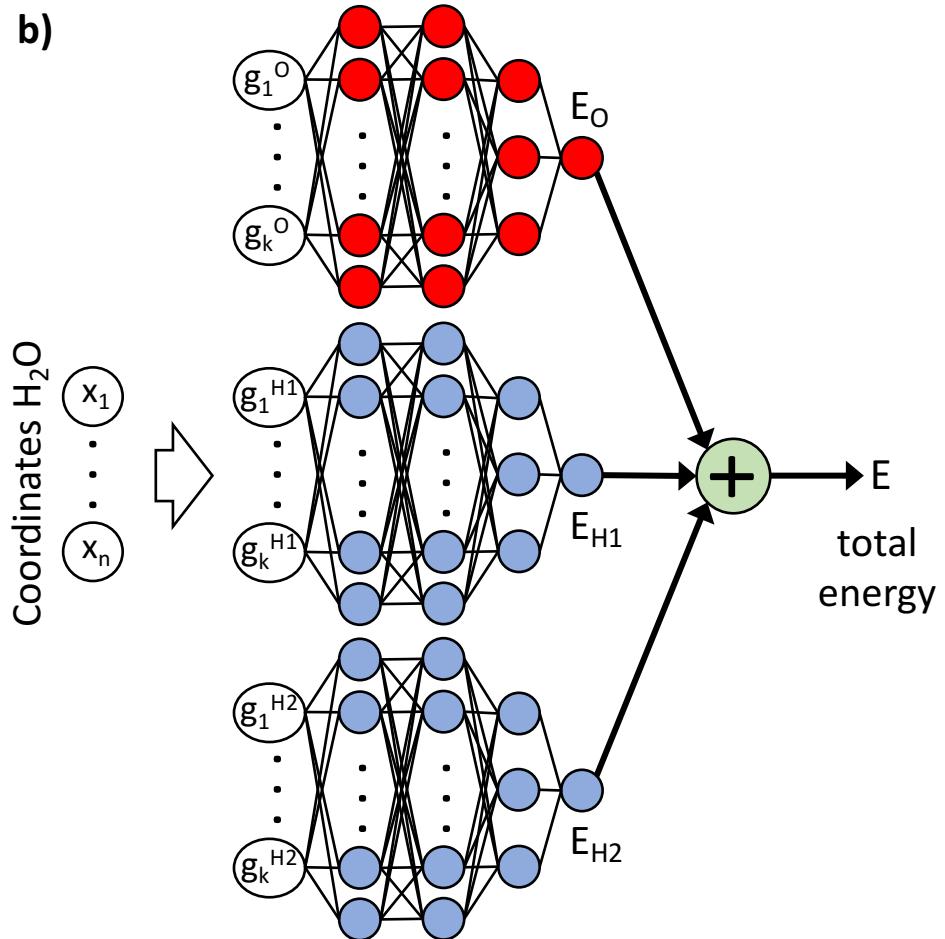


Behler-Parrinello
networks for
quantum mechanical
energies

$$L_{\text{ene}} = \sum (\hat{U}(\mathbf{x}_i, \theta) - U_i)^2$$

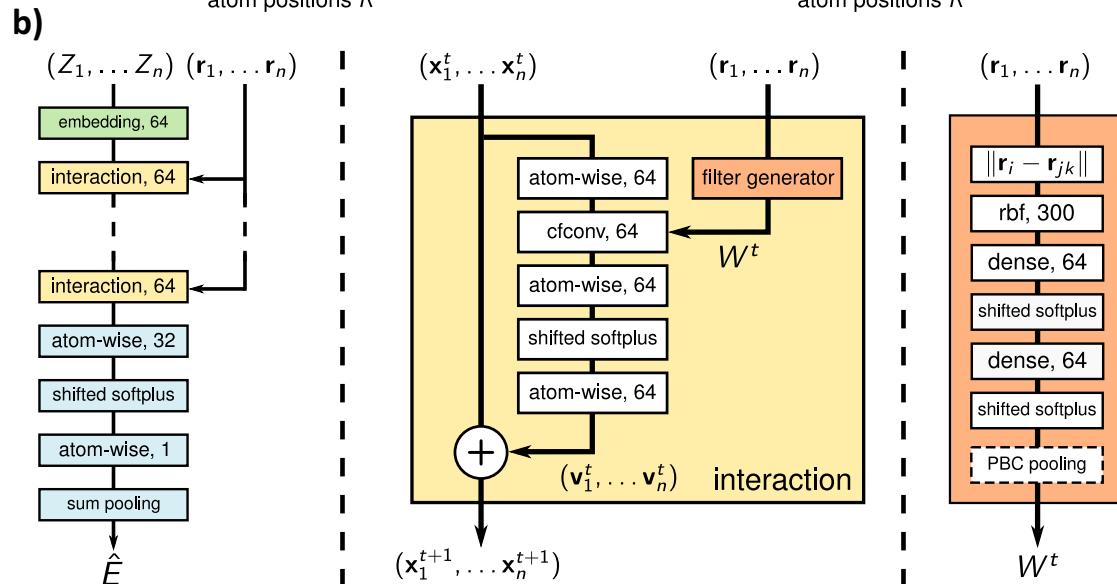
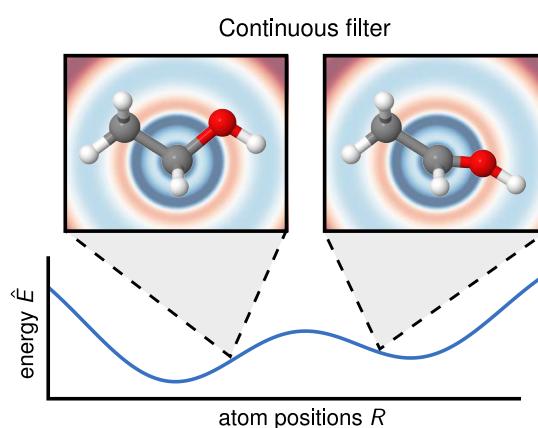
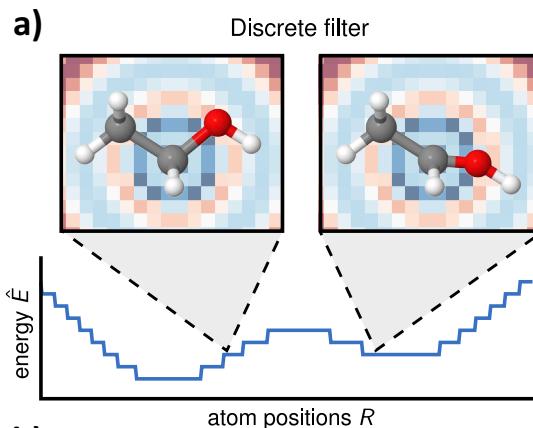
$$L_{\text{force}} = \sum_i \left\| \nabla \hat{U}(\mathbf{x}_i, \theta) + \mathbf{f}_i \right\|^2$$

b)



Behler & Parrinello, *Phys. Rev. Lett.* 98:146401 (2007)
 Smith, et al. *Chem. Sci.*, 8:3192-3203 (2017)
 Smith, et al. *chemrxiv.6744440.v1* (2018)
 Han et al. *Phys. Rev. Lett.*, 120:14300 (2018)

Force fields How can we do it with ML



Deep Tensor Neural Nets & Schnet

$$\mathbf{x}_i^{t+1} = (X^l * W^l)_i = \sum_j \mathbf{x}_j^l \circ W^l(\mathbf{r}_i - \mathbf{r}_j)$$

$$L_{\text{ene}} = \sum_i (\hat{U}(\mathbf{x}_i, \theta) - U_i)^2$$

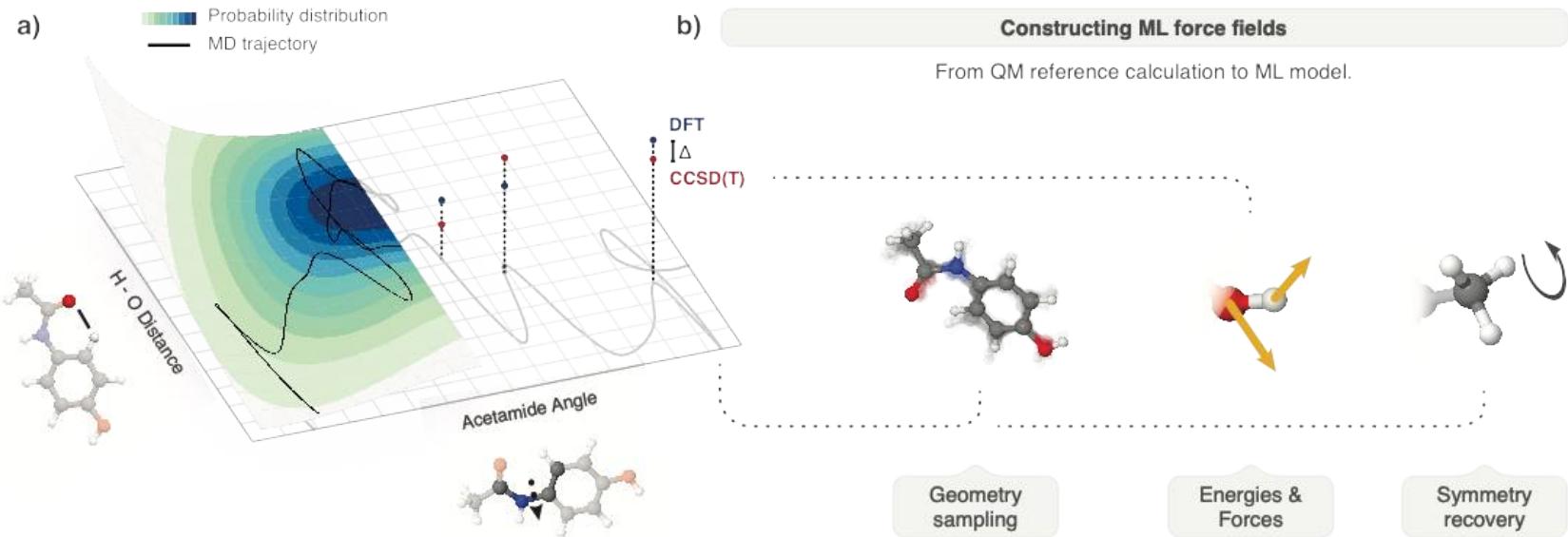
$$L_{\text{force}} = \sum_i \left\| \nabla \hat{U}(\mathbf{x}_i, \theta) + \mathbf{f}_i \right\|^2$$

Schütt, et al. *Nat. Commun.*, 8(1):13890 (2017)

Schütt, et al. *J. Chem. Phys.*, 148(24):241722 (2018)

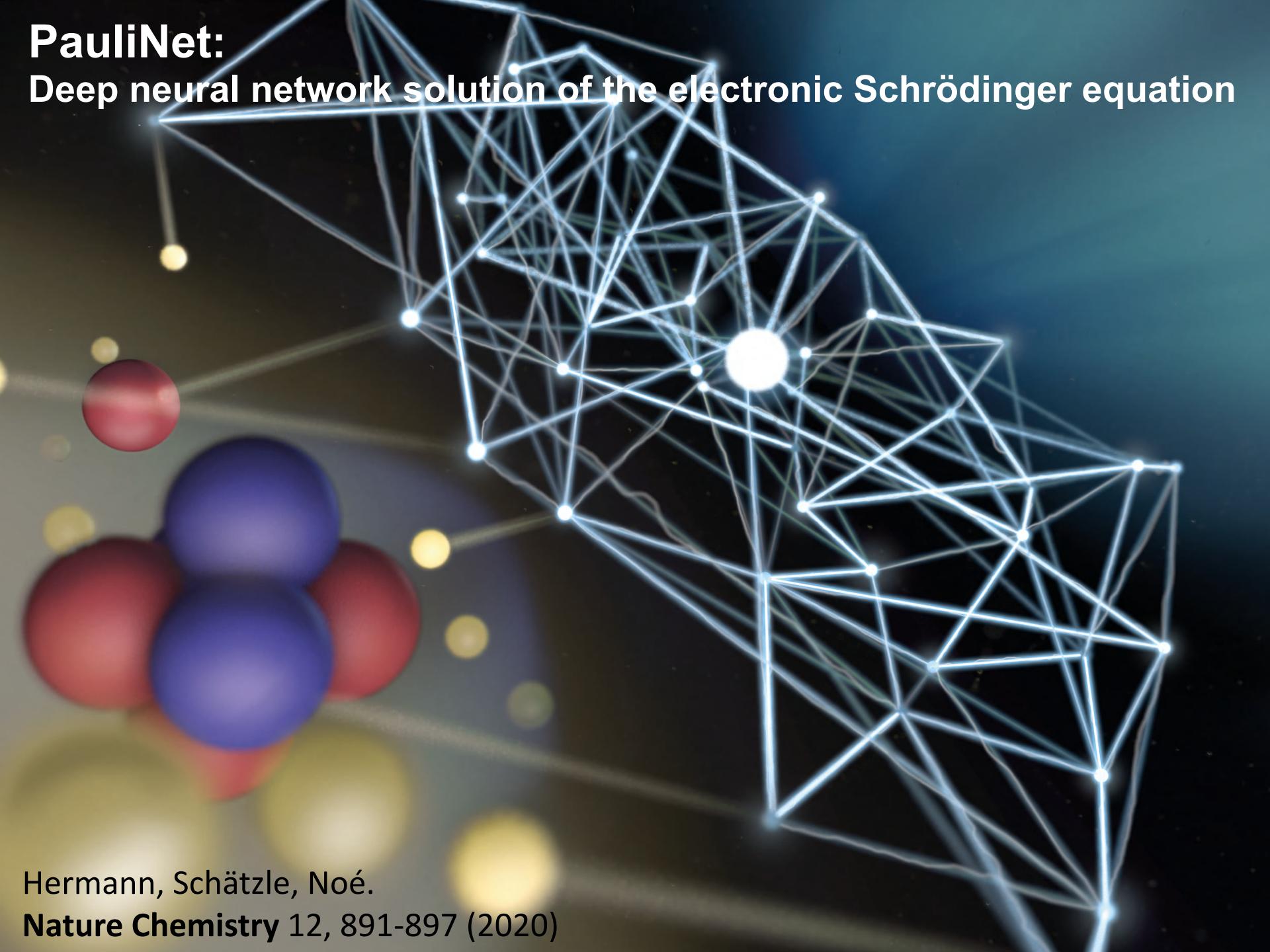
Force fields How can we do it with ML

symmetric Gradient Domain Machine Learning (sGDML)
uses a kernel approach rather than a Neural Network



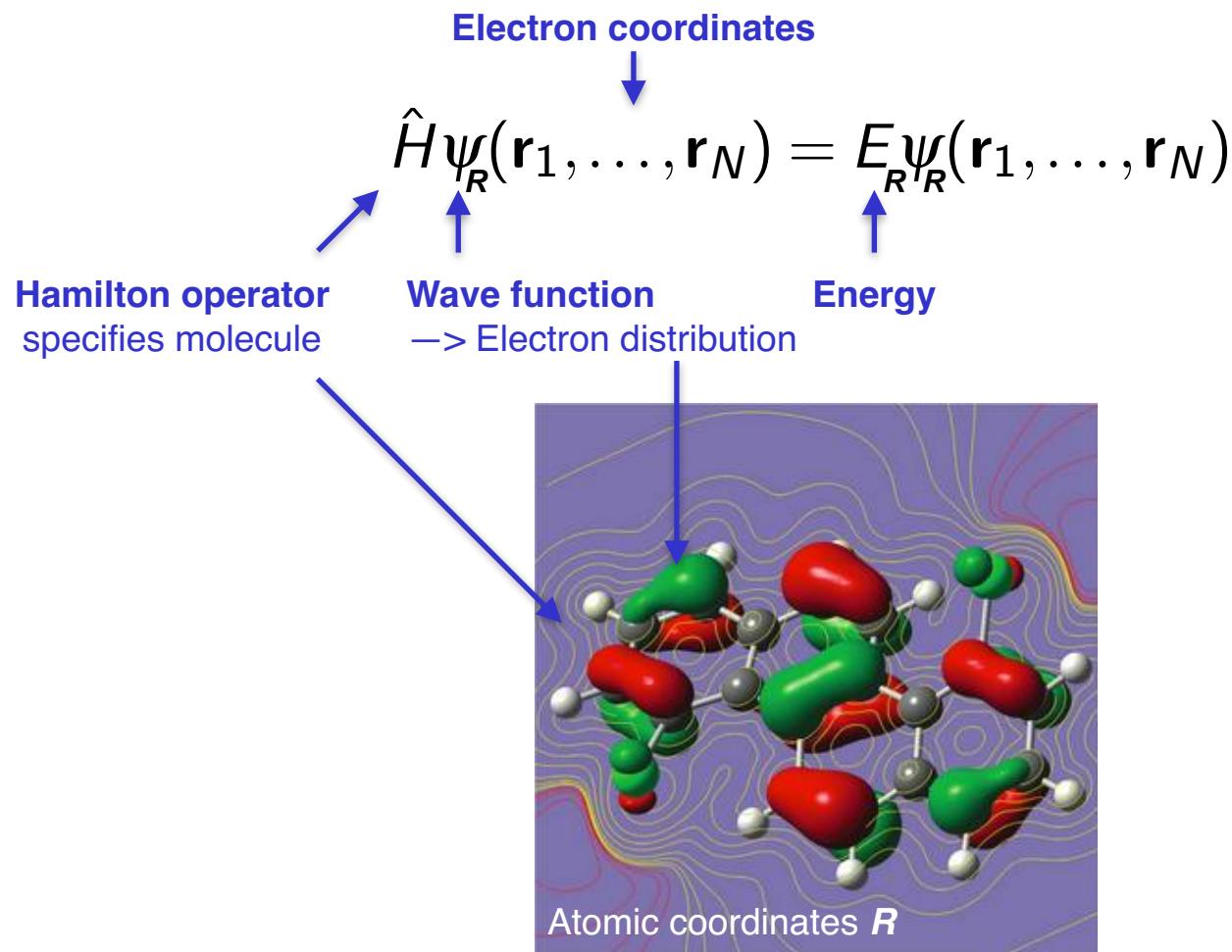
Chmiela et al. *Sci. Adv.* 3(5): e1603015 (2017)
Chmiela et al. *Nat. Commun.*, 9: 3887 (2018)

PauliNet: Deep neural network solution of the electronic Schrödinger equation



Hermann, Schätzle, Noé.
Nature Chemistry 12, 891-897 (2020)

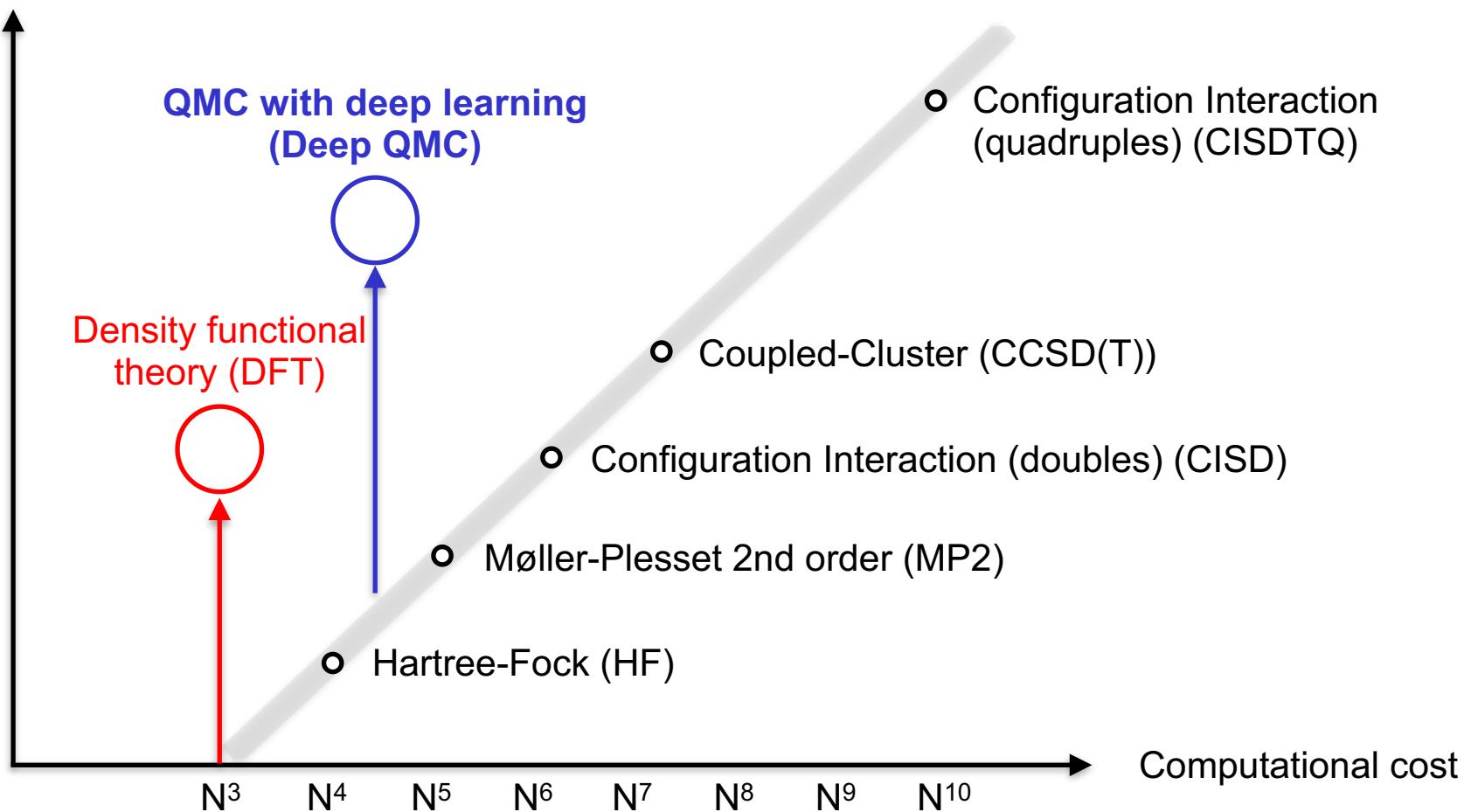
Electronic Schrödinger Equation (time independent, non-relativistic)



Solution of the electronic Schrödinger equation

$$\hat{H}\psi(\mathbf{r}_1, \dots, \mathbf{r}_N) = E\psi(\mathbf{r}_1, \dots, \mathbf{r}_N)$$

Accuracy



Variational approach as a keystone for machine learning

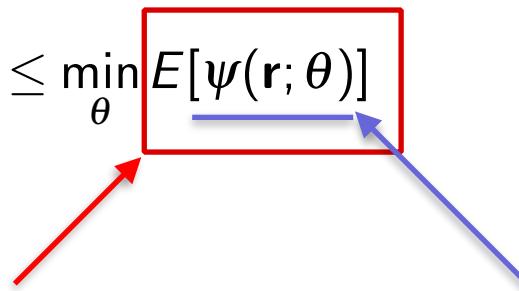
Variational approach

$$E_0 = \min_{\psi} E[\psi] \leq \min_{\theta} E[\psi(\mathbf{r}; \theta)]$$

Loss function

$$E[\psi] = \int d\mathbf{r} \psi(\mathbf{r}) \hat{H} \psi(\mathbf{r})$$

Represent with neural net



- **Variational Approach**

$$E_0 = \min_{\psi} E[\psi] \leq \min_{\theta} E[\psi(\mathbf{r}; \theta)] \quad E[\psi] = \int d\mathbf{r} \psi(\mathbf{r}) \hat{H} \psi(\mathbf{r})$$

- **VMC:** Variational Quantum Monte Carlo:

$$E[\psi; \theta] = \mathbb{E}_{\mathbf{r} \sim |\psi|^2} [E_{\text{loc}}[\psi](\mathbf{r}; \theta)] \quad E_{\text{loc}}[\psi](\mathbf{r}; \theta) = \hat{H}\psi(\mathbf{r}; \theta)/\psi(\mathbf{r}; \theta)$$

Variational Quantum Monte-Carlo (VMC)

- **Variational Approach**

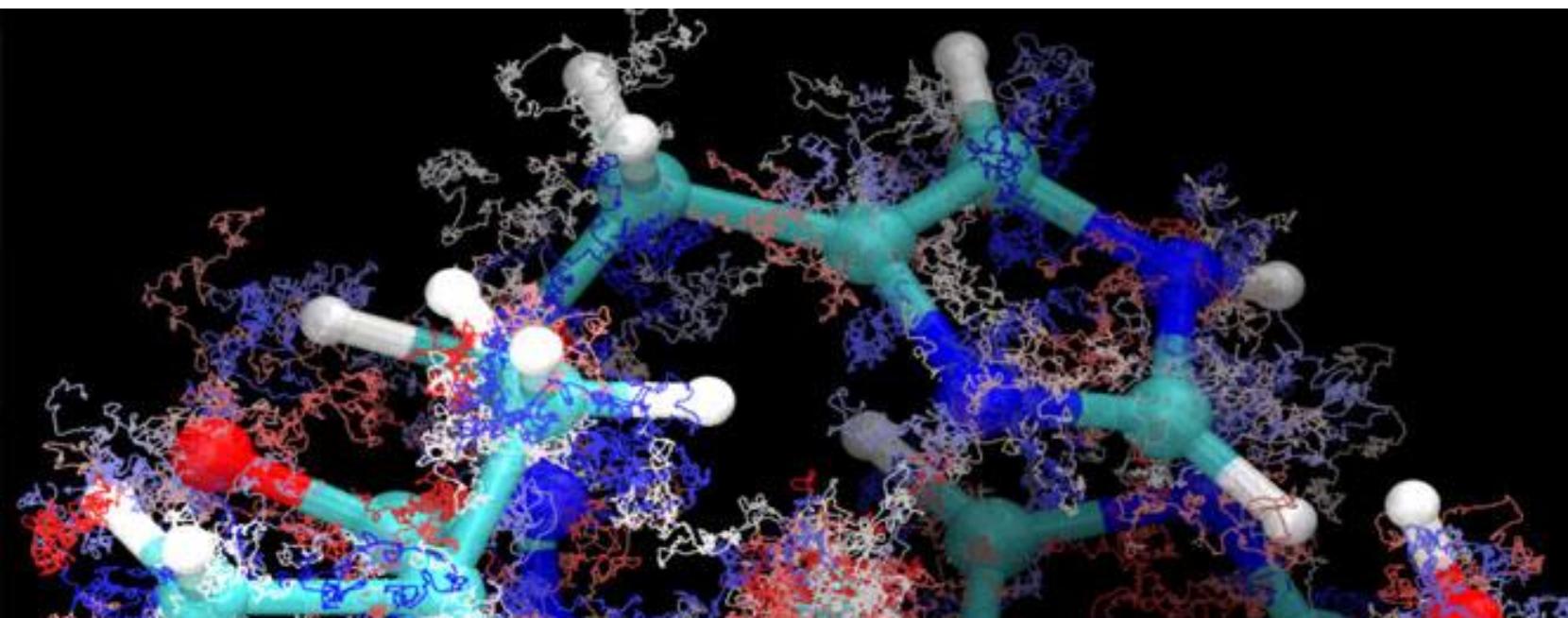
$$E_0 = \min_{\psi} E[\psi] \leq \min_{\theta} E[\psi(\mathbf{r}; \theta)] \quad E[\psi] = \int d\mathbf{r} \psi(\mathbf{r}) \hat{H} \psi(\mathbf{r})$$

- **VMC:** Variational Quantum Monte Carlo:

$$E[\psi; \theta] = \mathbb{E}_{\mathbf{r} \sim |\psi|^2} [E_{\text{loc}}[\psi](\mathbf{r}; \theta)] \quad E_{\text{loc}}[\psi](\mathbf{r}; \theta) = \hat{H}\psi(\mathbf{r}; \theta)/\psi(\mathbf{r}; \theta)$$



Sample Minibatches

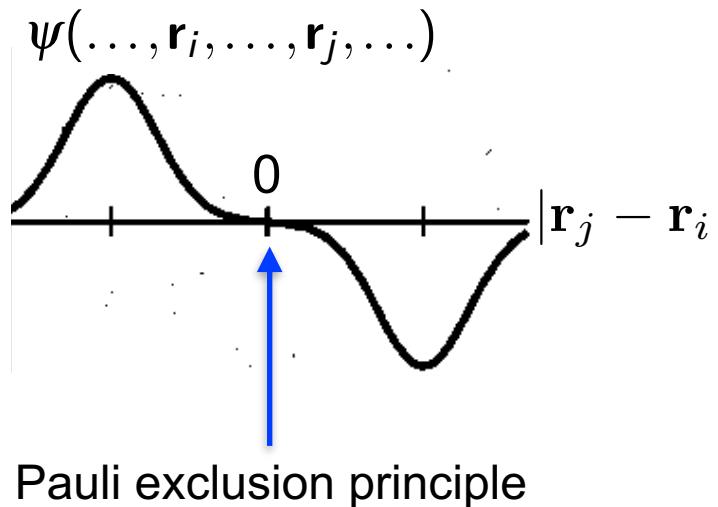


$$\hat{H}\psi(\mathbf{r}_1, \dots, \mathbf{r}_N) = E\psi(\mathbf{r}_1, \dots, \mathbf{r}_N)$$

Electron spins $s_i \in \{\uparrow, \downarrow\}$.

ψ is antisymmetric wrt. exchange of electrons of equal spin

$$\psi(\dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots) = -\psi(\dots, \mathbf{r}_j, \dots, \mathbf{r}_i, \dots)$$



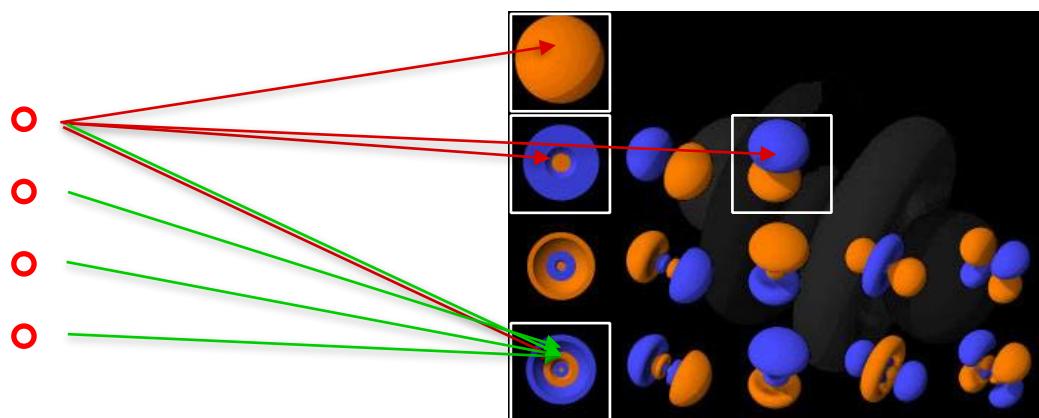
PauliNet Wave Function — Antisymmetry

$$\psi(\mathbf{r}; \theta) = \det[\varphi_\mu(\mathbf{r}_i^\uparrow) f_{\mu i}(\mathbf{r}; \theta)] \det[\varphi_\mu(\mathbf{r}_i^\downarrow) f_{\mu i}(\mathbf{r}; \theta)] e^{\gamma(\mathbf{r}) + J(\mathbf{r}; \theta)}$$

↑
antisymmetric part

Standard approach: Slater Determinants of single-electron functions

N electrons → Choose N functions (orbitals) → Pairwise combinations



$$\begin{vmatrix} \varphi_1(\mathbf{r}_1^\uparrow) & \varphi_2(\mathbf{r}_1^\uparrow) & \cdots & \varphi_N(\mathbf{r}_1^\uparrow) \\ \varphi_1(\mathbf{r}_2^\uparrow) & \varphi_2(\mathbf{r}_2^\uparrow) & \cdots & \varphi_N(\mathbf{r}_2^\uparrow) \\ \vdots & \vdots & & \vdots \\ \varphi_1(\mathbf{r}_N^\uparrow) & \varphi_2(\mathbf{r}_N^\uparrow) & \cdots & \varphi_N(\mathbf{r}_N^\uparrow) \end{vmatrix}$$

↓ Determinant
Antisymmetric wavefunction

- Single determinant is not very expressive
- Many Slater determinants needed for high accuracy (CI)

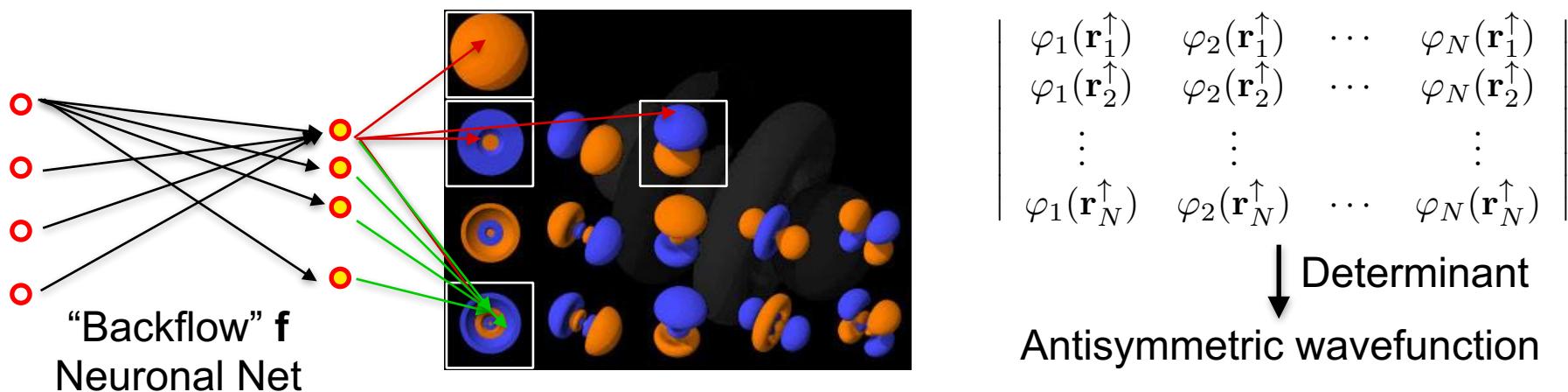
PauliNet Wave Function — Antisymmetry

$$\psi(\mathbf{r}; \theta) = \det[\varphi_\mu(\mathbf{r}_i^\uparrow) f_{\mu i}(\mathbf{r}; \theta)] \det[\varphi_\mu(\mathbf{r}_i^\downarrow) f_{\mu i}(\mathbf{r}; \theta)] e^{\gamma(\mathbf{r}) + J(\mathbf{r}; \theta)}$$

↑
antisymmetric part

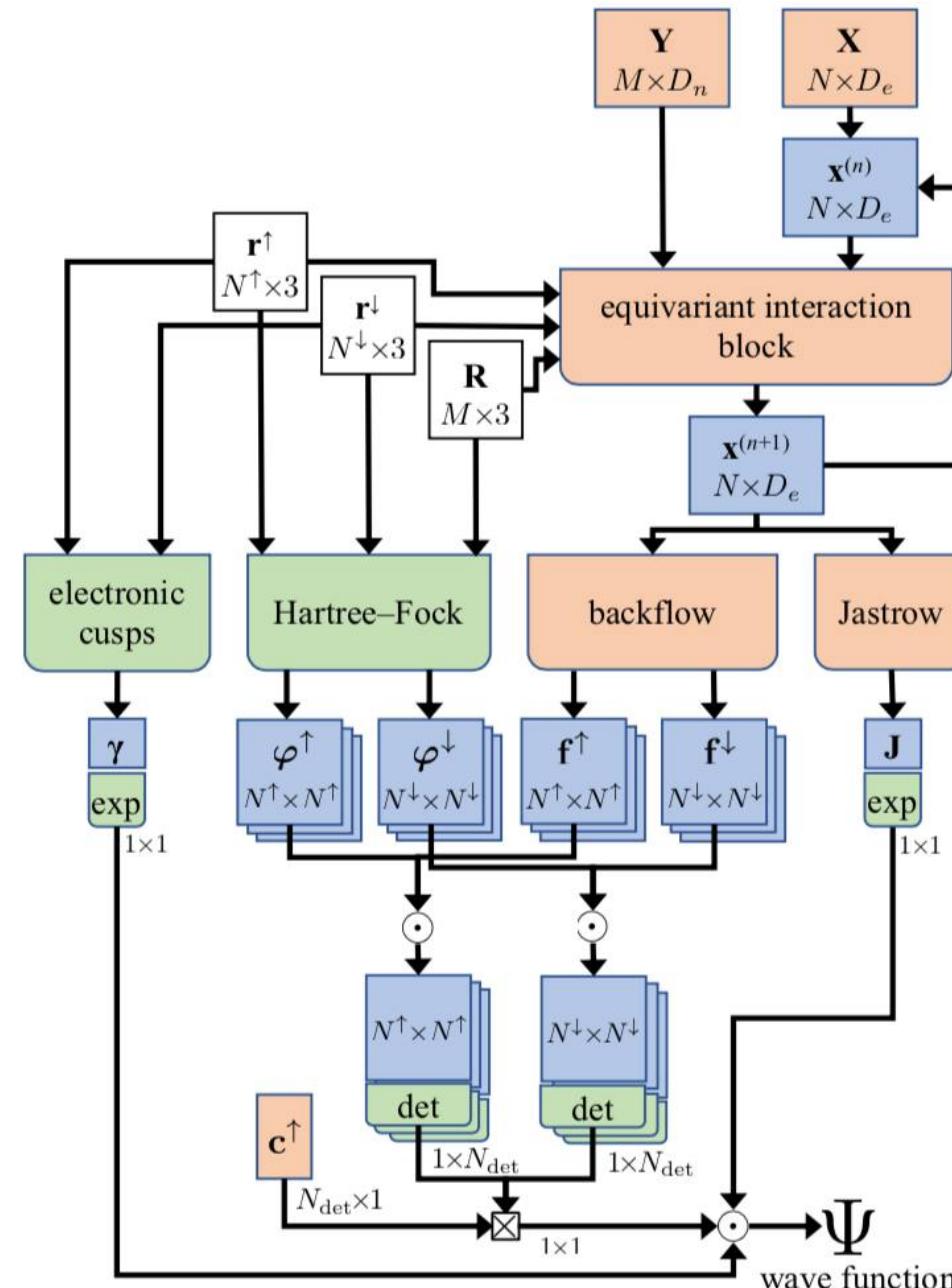
Backflow: Determinants of many-electron functions

N electrons → Choose N functions (orbitals) → Pairwise combinations

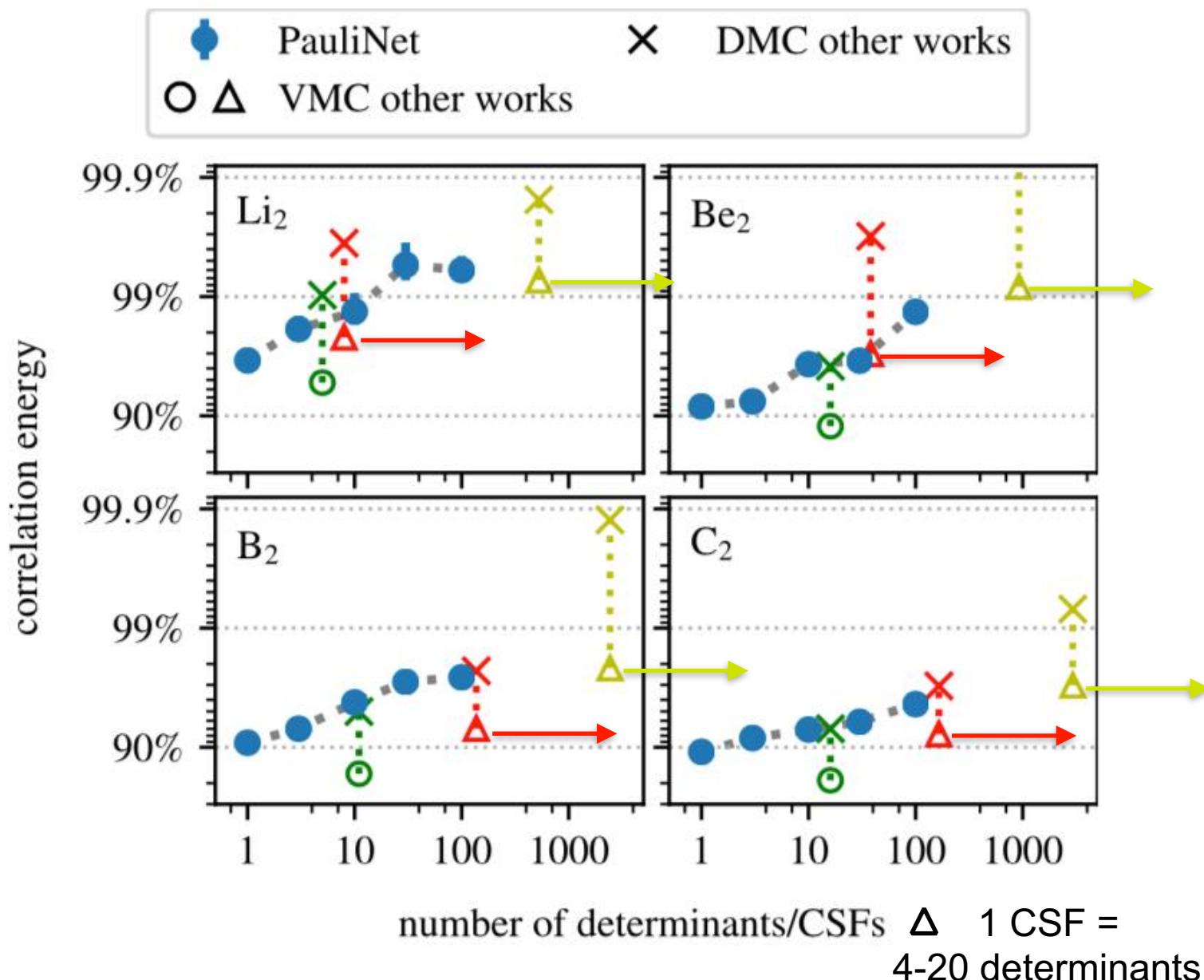


- Much more expressive
- Few Slater determinants necessary for high accuracy

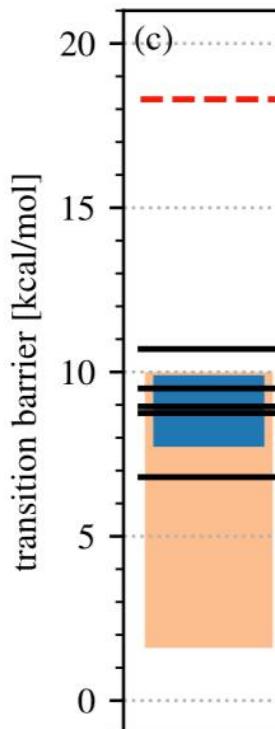
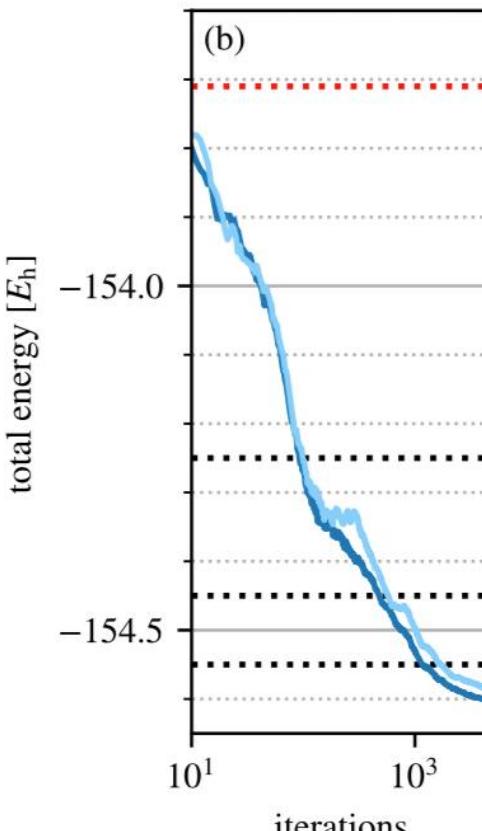
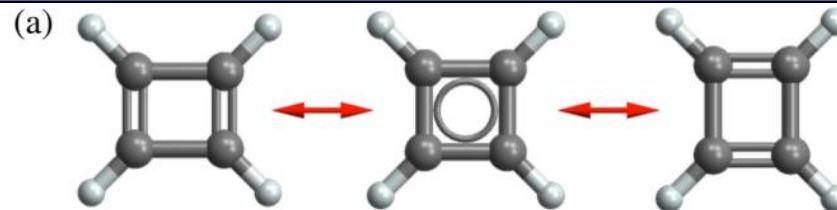
| | |
|--|--------------------------------|
| N | Number of electrons |
| N^\uparrow | Number of spin-up electrons |
| N^\downarrow | Number of spin-down electrons |
| M | Number of nuclei |
| D_e | Embedding dimension, electrons |
| D_n | Embedding dimension, nuclei |
| Neural Network (trainable function) | trainable array |
| Fixed function | Input Array |
| | Hidden Array |



PauliNet: Diatomic Molecules

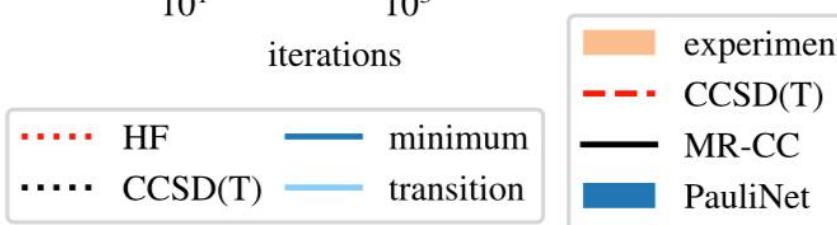


PauliNet: cyclobutadien



10 determinants
ca. 40 GPU hours

Lyakh et al,
Chem. Rev 112:182
(2012)



$\sim O(N^7)$
 $\sim O(N^4)$

Force fields

Open challenges

A useful ML force field needs to be:

- i) ACCURATE (energy error < 0.2-0.3 kcal/mol/functional group)
- ii) CHEAP
- iii) SCALABLE
- iv) TRANSFERABLE

Such universal model does not exist yet

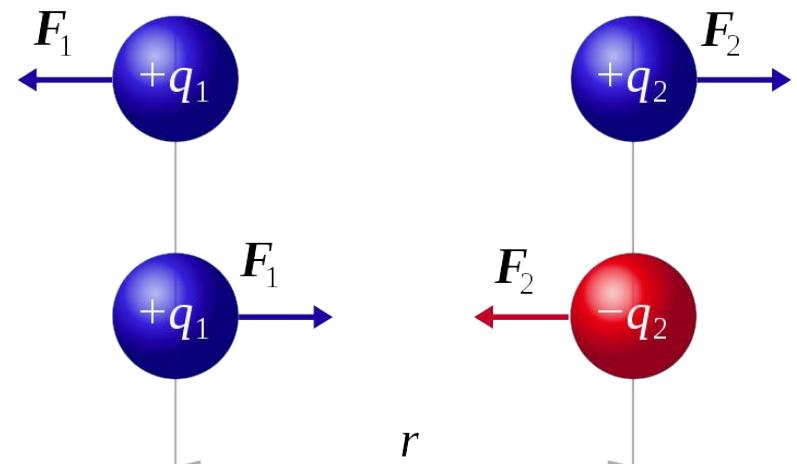
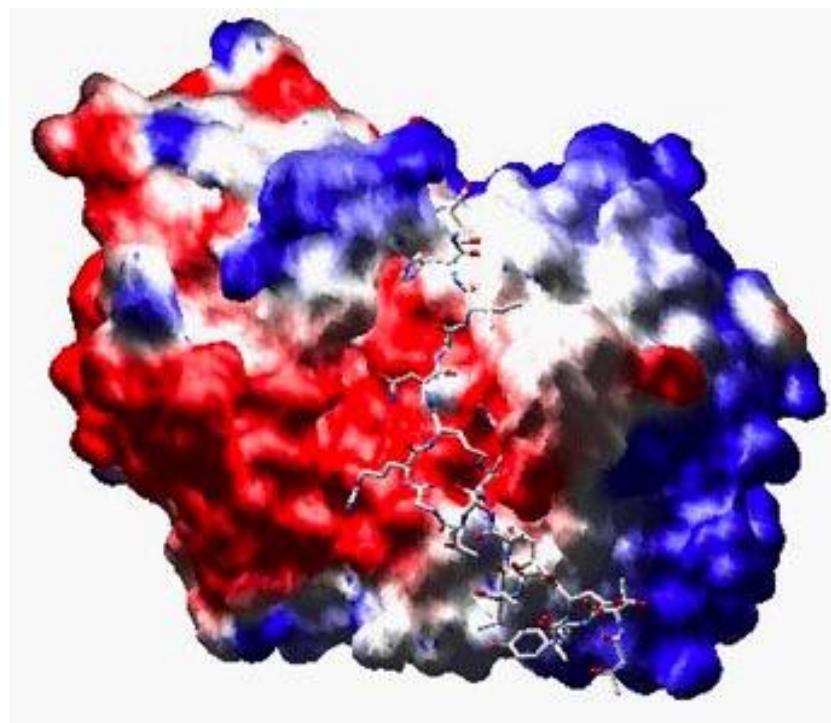
Force fields

Open challenges

How to include long-ranged interactions?

Combine short-ranged ML models with
long-ranged electrostatics methods (e.g., Ewald)?

Long-range character of van der Waals dispersion?



$$U_c(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r_{ij}}$$

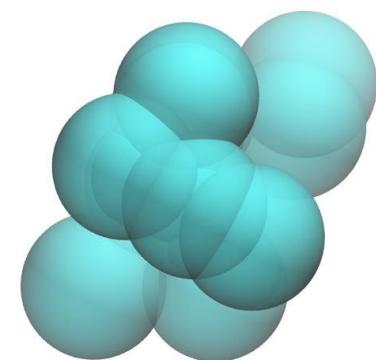
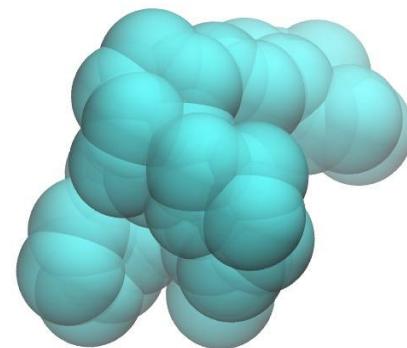
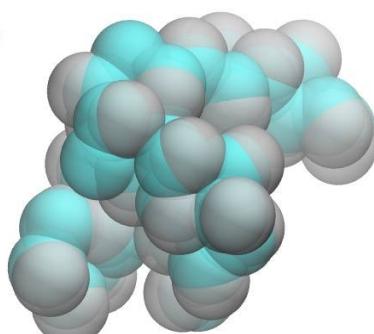
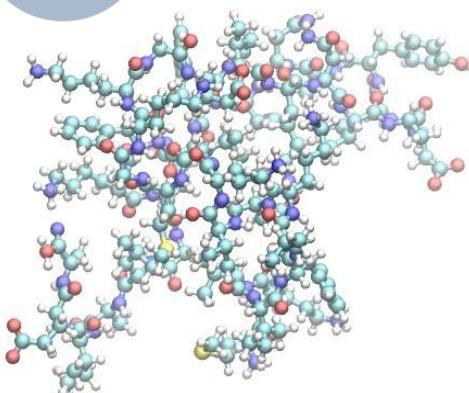
Force fields

Open challenges

Can we push this further and
design force-fields at coarse-grained resolution?

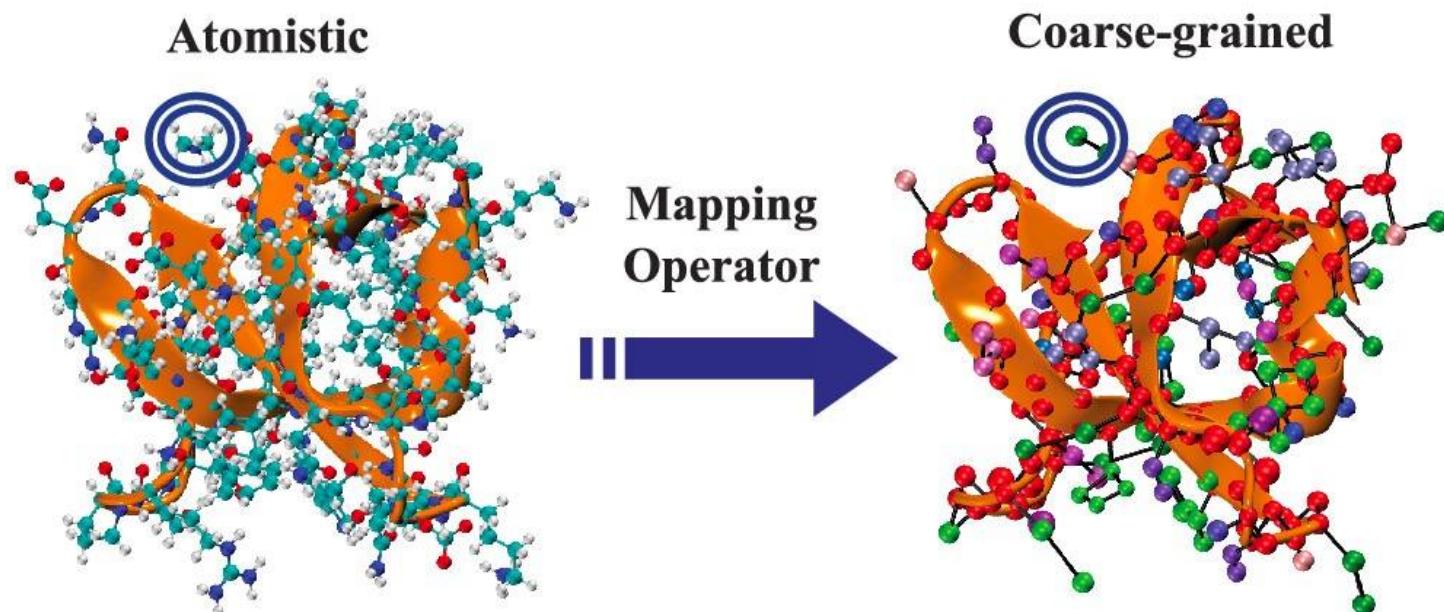
\mathbb{R}^3

coarse-graining in structural space

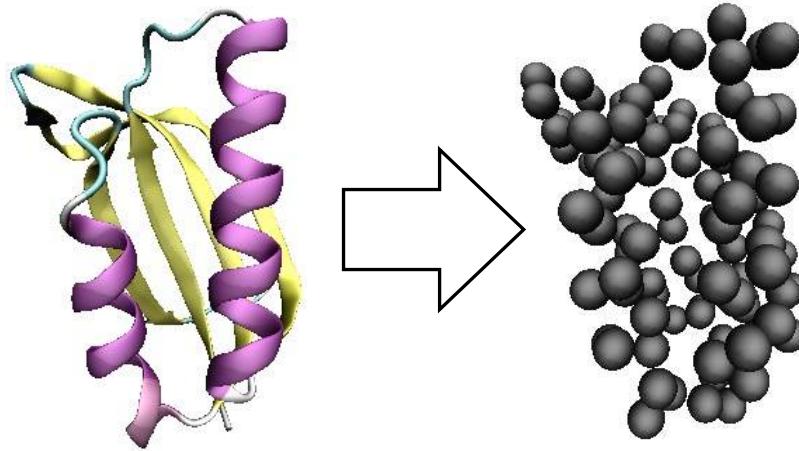


Coarse-grained mapping

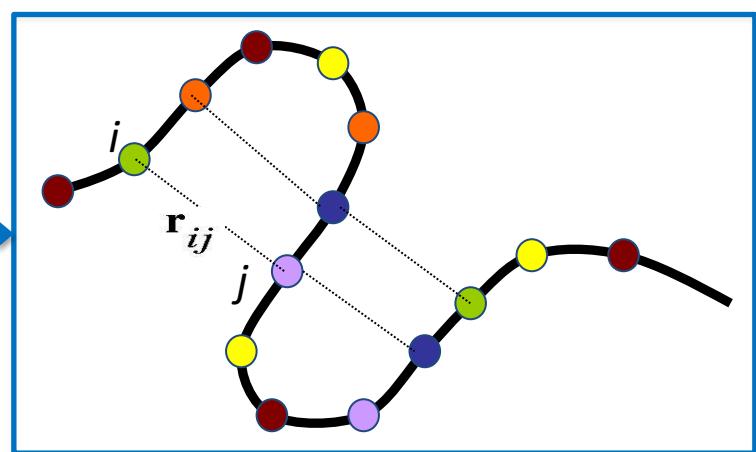
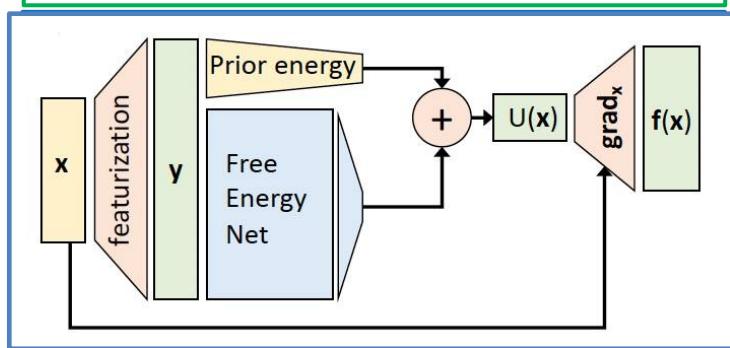
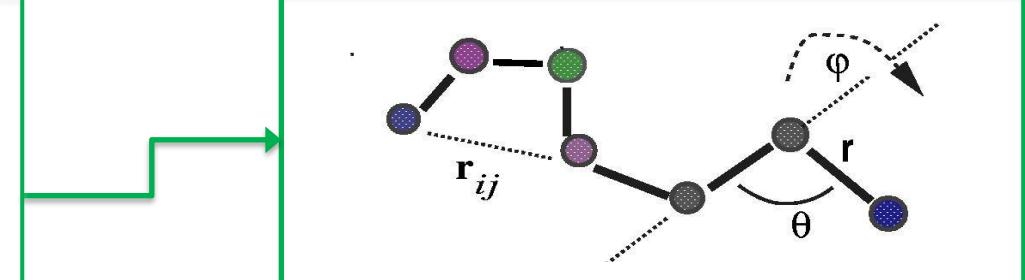
| | | | |
|--------------------------|--|---------------------|-----------------|
| atomistic representation | $\mathbf{r} \in \mathbb{R}^{3N}$ | atomistic potential | $V(\mathbf{r})$ |
| CG representation | $\mathbf{x} = \xi(\mathbf{r}) \in \mathbb{R}^{3n}$ | CG potential | $U(\mathbf{x})$ |



Coarse-grained molecular model

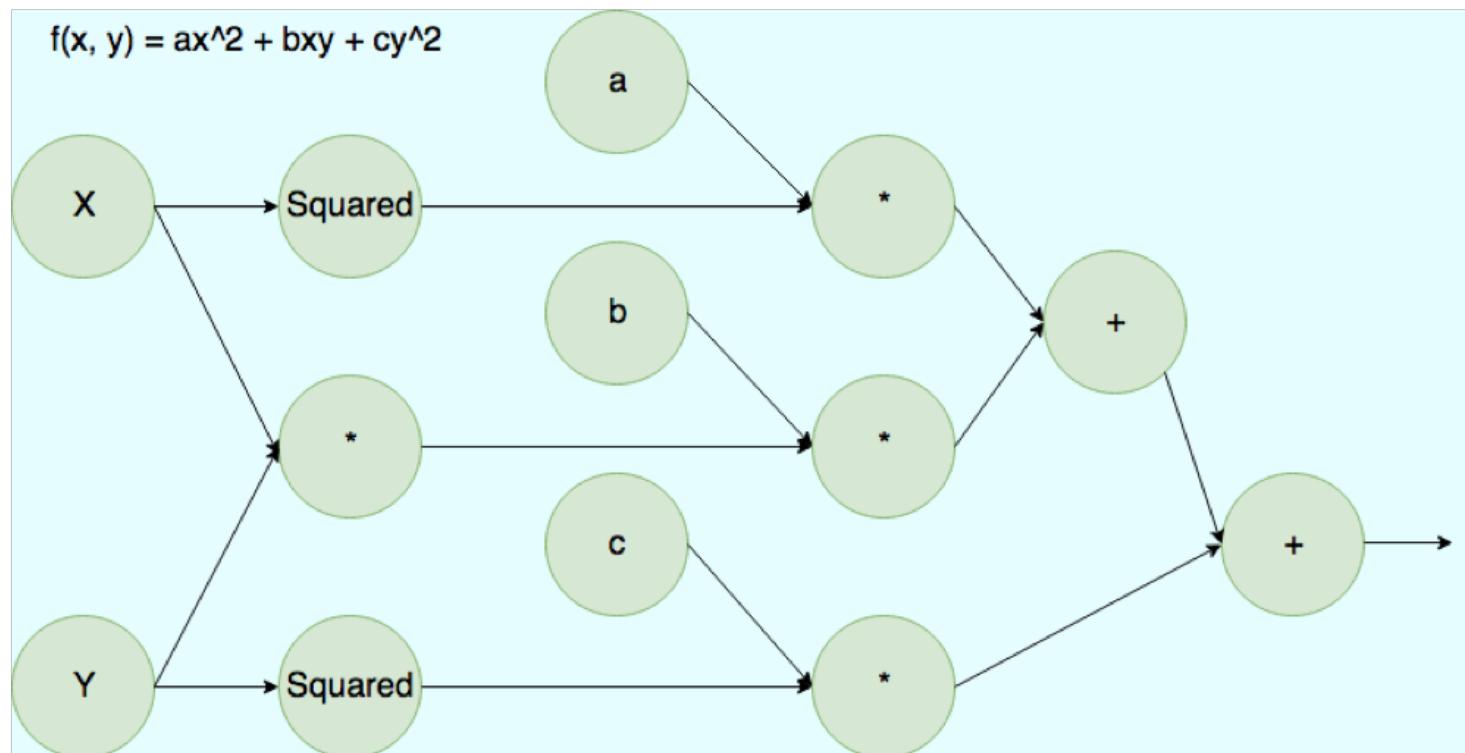


$$H = \sum_{\text{bonds}} K_r (r - r_0)^2 + \sum_{\text{angles}} K_\theta (\theta - \theta_0)^2 + \sum_{\text{dihedrals}} K_d (1 + \cos(n(\phi - \phi_0)))$$



Why Deep Learning?

- In quantitative sciences we have **computations** and **parameters**.
- Deep learning is an approach to **program any computation** and **fit the parameters** in an **automatical way**
- **Main approach:** end-to-end learning of differentiable computational models
 - Represent computation by a computational graph
 - Backpropagate error with respect to learning goal from output to input
 - Change parameters to minimize error.



Coarse-graining with thermodynamic consistency

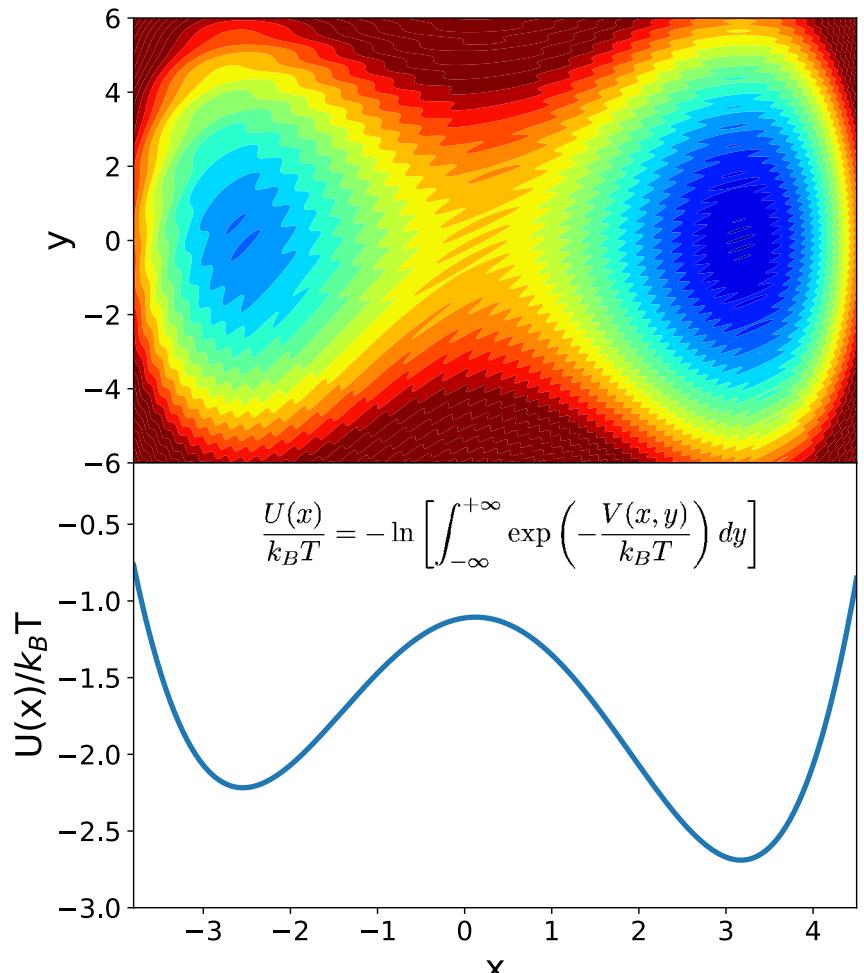
| | | | |
|--------------------------|--|---------------------|-----------------|
| atomistic representation | $\mathbf{r} \in \mathbb{R}^{3N}$ | atomistic potential | $V(\mathbf{r})$ |
| CG representation | $\mathbf{x} = \xi(\mathbf{r}) \in \mathbb{R}^{3n}$ | CG potential | $U(\mathbf{x})$ |

Thermodynamic consistency

$$U(\mathbf{x}) = -k_B T \ln p^{CG}(\mathbf{x}) + \text{const}$$

$$p^{CG}(\mathbf{x}) = \frac{\int \exp\left(-\frac{V(\mathbf{r})}{k_B T}\right) \delta(x - \xi(\mathbf{r})) d\mathbf{r}}{\int \exp\left(-\frac{V(\mathbf{r})}{k_B T}\right) d\mathbf{r}}$$

GOAL:
Optimize the parameters of
a CG model $U(\mathbf{x}; \boldsymbol{\theta})$
to satisfy this consistency
as best as possible



Force matching

Enforcing the thermodynamic consistency is equivalent to minimize the force matching error:

$$\chi^2(\boldsymbol{\theta}) = \left\langle \|\xi(\mathbf{F}(\mathbf{r})) + \nabla U(\xi(\mathbf{r}); \boldsymbol{\theta})\|^2 \right\rangle_{\mathbf{r}}$$

instantaneous atomistic forces

The force matching error is always > 0 and can be decomposed in two parts:

$$\chi^2(\boldsymbol{\theta}) = \text{PMF error}(\boldsymbol{\theta}) + \text{Noise}$$

The PMF error is what should be as close to zero as possible

$$\text{PMF error}(\boldsymbol{\theta}) = \langle \|\mathbf{f}(\xi(\mathbf{r})) + \nabla U(\xi(\mathbf{r}); \boldsymbol{\theta})\|^2 \rangle_{\mathbf{r}}$$

$$\text{Noise} = \langle \|\xi(\mathbf{F}(\mathbf{r})) - \mathbf{f}(\xi(\mathbf{r}))\| \rangle_{\mathbf{r}}$$

The noise term is determined solely by the coarse-graining mapping

$$\mathbf{f}(\mathbf{x}) = \langle \xi(\mathbf{F}(\mathbf{r})) \rangle_{\mathbf{r}|\mathbf{x}}$$

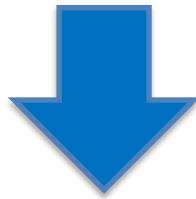
mean force

the gradient of the CG potential should be as close as possible to the “mean force”

Coarse-graining as a machine learning problem

Enforcing the thermodynamic consistency is equivalent to minimize the force matching error:

$$\chi^2(\boldsymbol{\theta}) = \left\langle \|\xi(\mathbf{F}(\mathbf{r})) + \nabla U(\xi(\mathbf{r}); \boldsymbol{\theta})\|^2 \right\rangle_{\mathbf{r}}$$



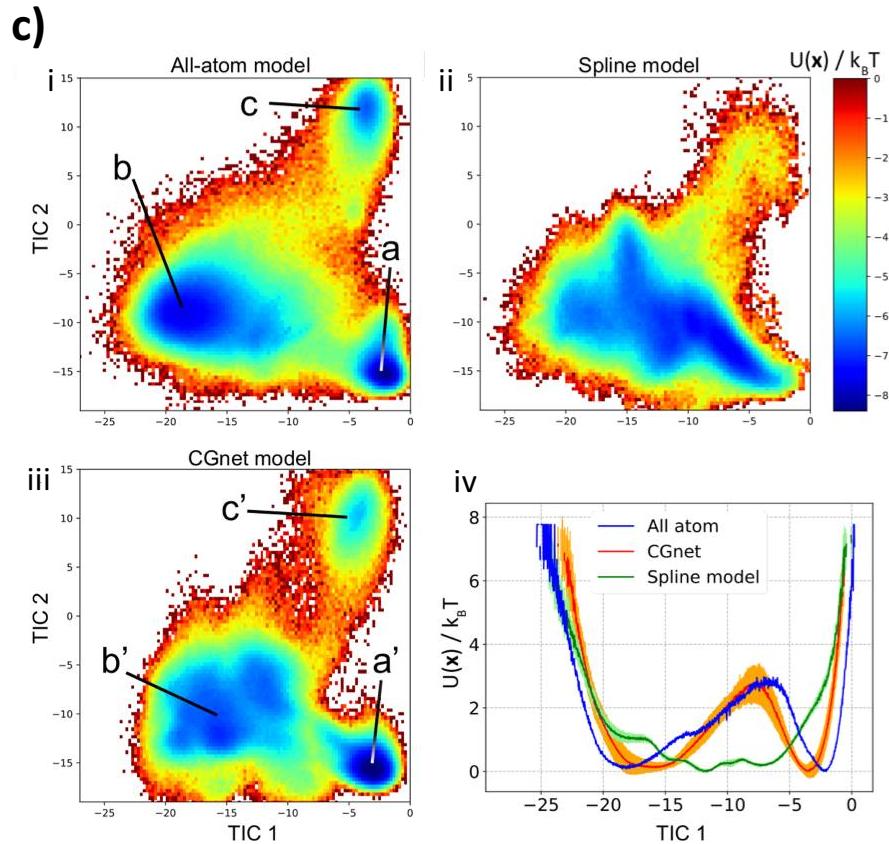
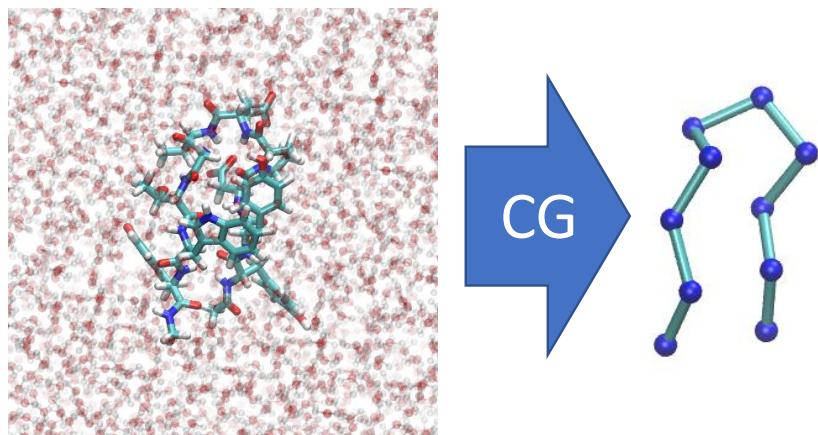
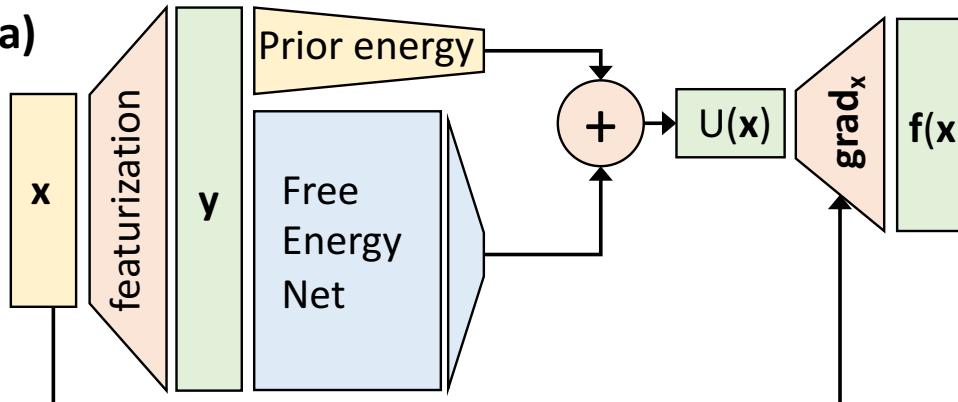
Define a loss function to minimize over a Boltzmann-distributed sample:

$$\begin{aligned} L(\boldsymbol{\theta}; \mathbf{R}) &= \frac{1}{M} \sum_{i=1}^M \|\xi(\mathbf{F}(\mathbf{r}_i)) + \nabla U(\xi(\mathbf{r}_i); \boldsymbol{\theta})\|^2 \\ &= \|\xi(\mathbf{F}(\mathbf{R})) + \nabla U(\xi(\mathbf{R}); \boldsymbol{\theta})\|_F^2 \end{aligned}$$

Force fields

Open challenges

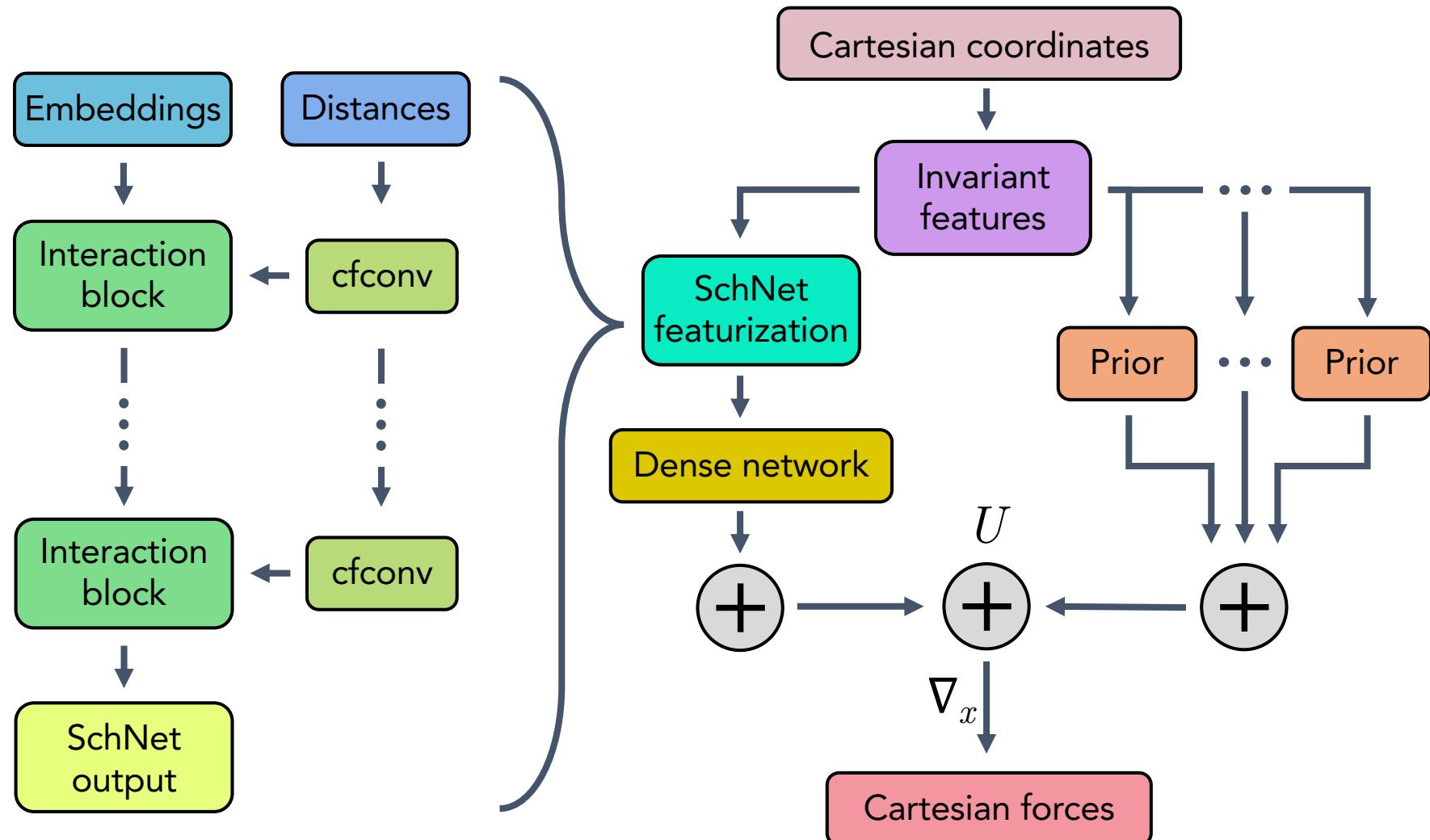
CGnets



Wang et al. *ACS Cent. Sci.* 5:755-767 (2019)

Zhang et al. *J. Chem. Phys.*, 149:034101 (2018)

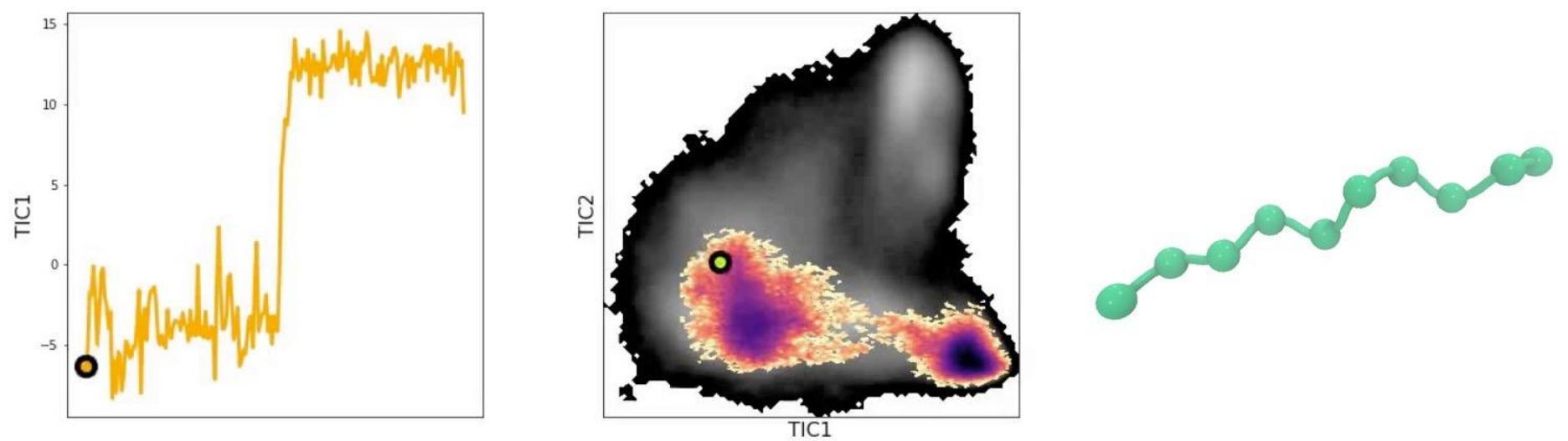
Towards a transferable model: Replacing structural features with graph neural networks



SchNet : K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, K.-R. Müller,
J. Chem. Phys. 148, 241722 (2018).

CGSchNet : B. E. Husic, N. E. Charron, ..., G. de Fabritiis, F. Noé, C. Clementi
J. Chem. Phys. 153 (19), 194101 (2020).

Coarse-graining of Chignolin folding/unfolding



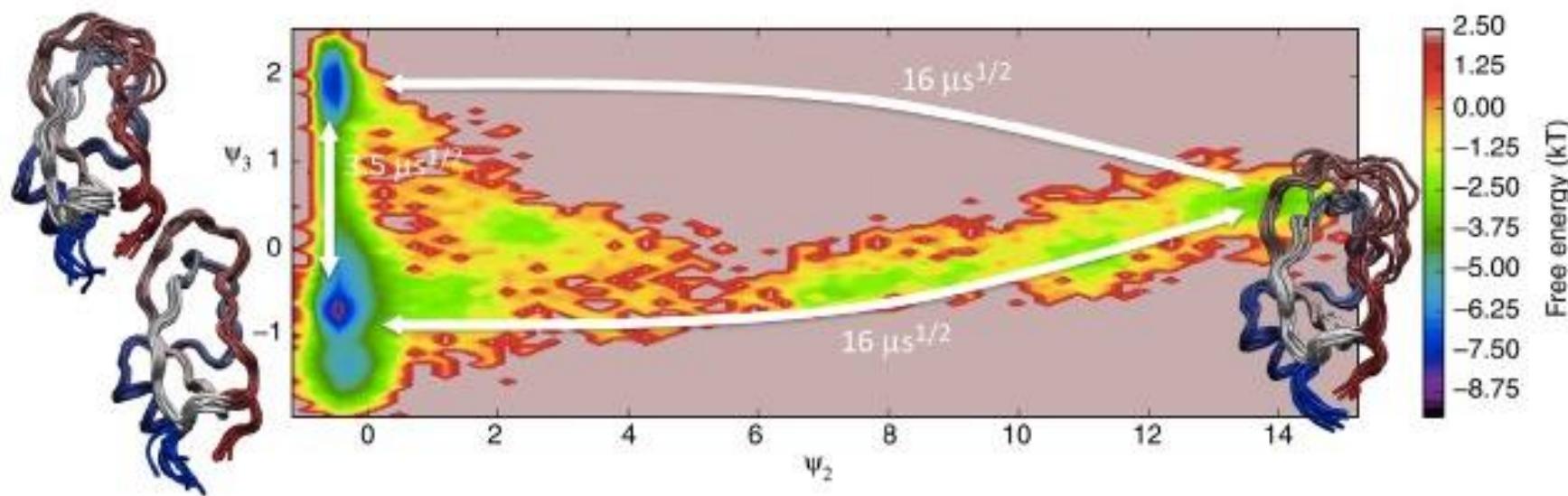
How we have been doing Molecular Simulation before ML

2. Kinetics

$$p(\mathbf{x}_{t+\tau}) = \int p(\mathbf{x}_{t+\tau} \mid \mathbf{x}_t; \tau) p(\mathbf{x}_t) d\mathbf{x}_t$$
$$\approx \sum_{k=1}^n \sigma_k^* \langle p(\mathbf{x}_t) \mid \phi(\mathbf{x}_t) \rangle \psi(\mathbf{x}_{t+\tau}).$$

Spectral Decomposition of Markov Operator

Markov State Models
Variational Principle
Diffusion Maps



Spectral decomposition

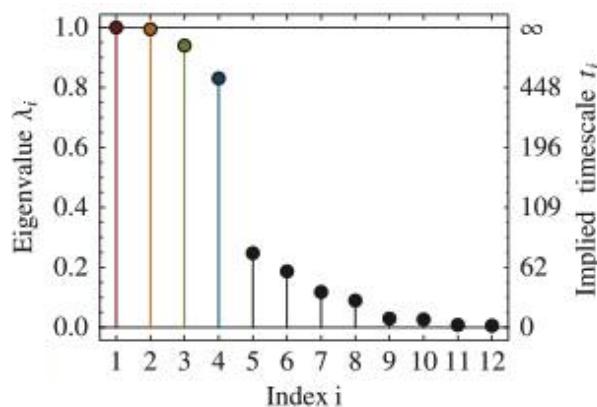
Long-time action of the MD propagator is described by its large eigenvalues + eigenfunctions

Schütte et al: J. Comput. Phys. (1999), Prinz et al.: J. Chem. Phys. 134, p174105 (2011)

Backward propagator

$$\rho_\tau = \mathcal{T}(\tau)\rho_0$$

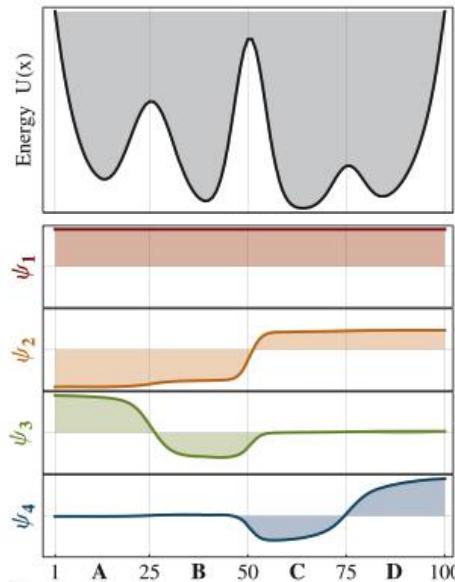
Eigenvalues / timescales κ_i^{-1}



Spectral decomposition

$$\rho_\tau = \sum_{i=1}^{\infty} e^{-\tau \kappa_i} \langle \psi_i | \rho_0 \rangle \psi_i$$

Slow processes:



Variational Approach to approximating Markov Processes

Noé, Nüske, SIAM MMS (2013)
Wu, Noé, arXiv:1707.04659

Variational Approach for Markov Processes (VAMP)

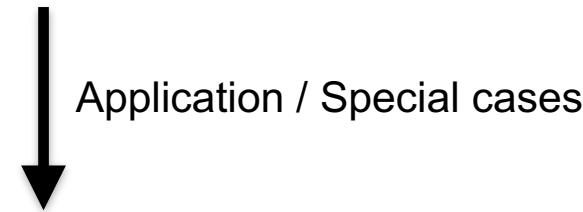
The first m eigenfunctions ψ_1, \dots, ψ_m are the solution to the problem

$$\begin{aligned} & \max_{f_1, \dots, f_m} \sum_{i=1}^m \mathbb{E} [f_i(\mathbf{x}_t) f_i(\mathbf{x}_{t+\tau})] \\ \text{s.t. } & \mathbb{E} [f_i(\mathbf{x}_t)^2] = 1 \\ & \mathbb{E} [f_i(\mathbf{x}_t) f_j(\mathbf{x}_{t+\tau})] = 0, \text{ for } i \neq j \end{aligned} \tag{1}$$

and the maximum value is the sum of $\lambda_1, \dots, \lambda_m$

Noé and Nüske, **MMS** 11, 635-655 (2013)

Nüske et al, **JCTC** 10, 1739-1752 (2014)



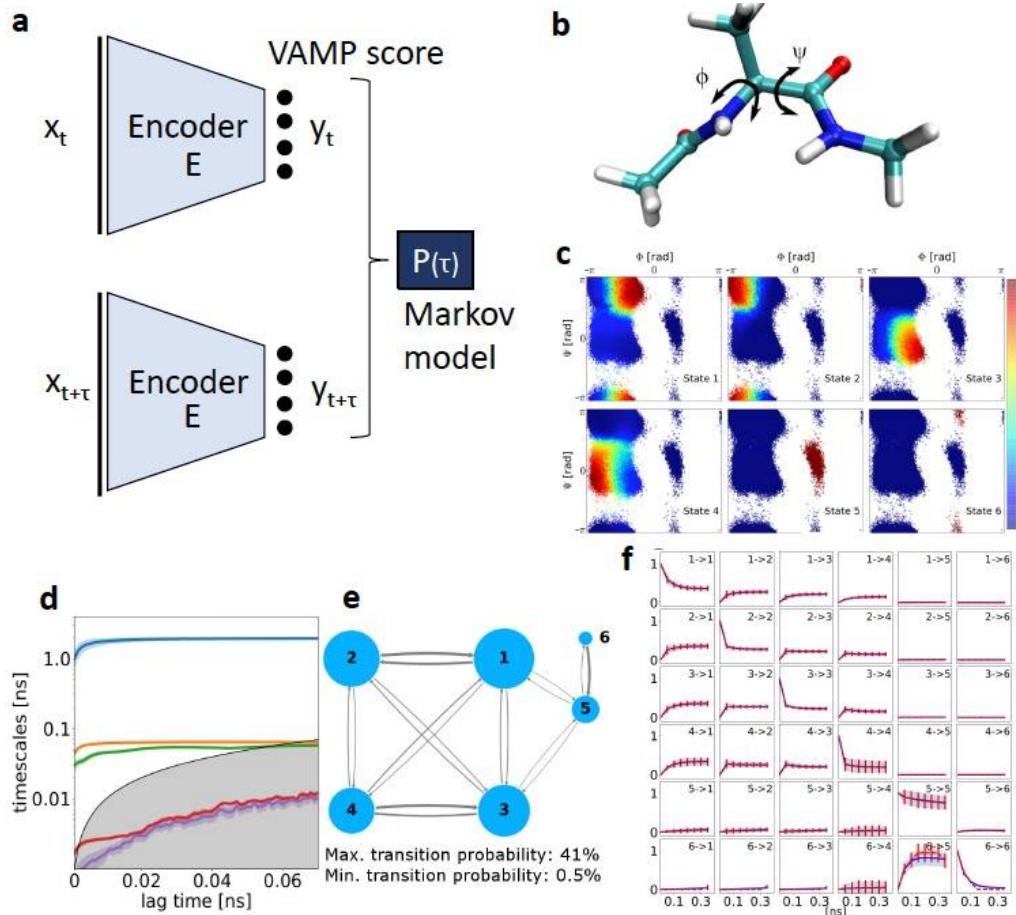
- | | |
|---|--|
| <ul style="list-style-type: none">• TICA• kernel TICA• MSMs | <p>Under additional assumptions</p> <ul style="list-style-type: none">• SGOOP• Diffusion Maps |
|---|--|

Kinetics

How can we do it with ML

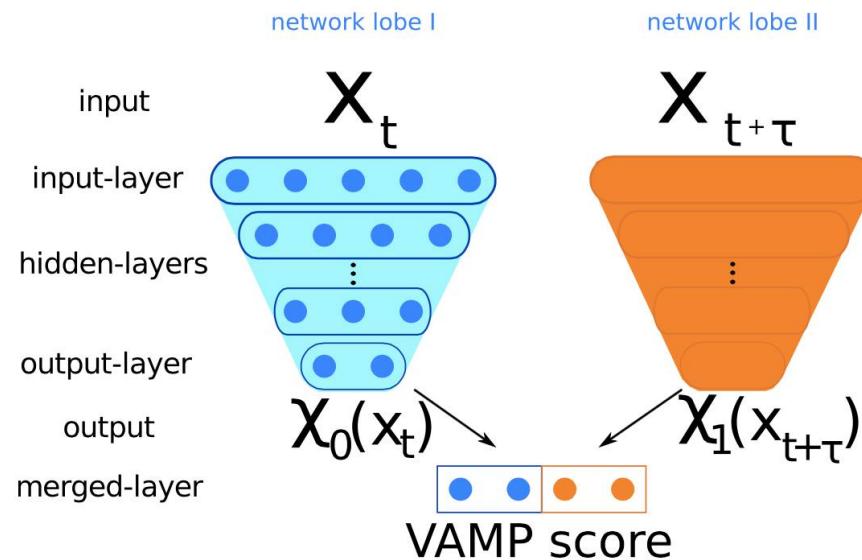
$$\mathbf{C}_{00} = \frac{1}{T} \sum_{t=1}^{T-\tau} \mathbf{y}_t \mathbf{y}_t^\top \quad \mathbf{C}_{0\tau} = \frac{1}{T} \sum_{t=1}^{T-\tau} \mathbf{y}_t \mathbf{y}_{t+\tau}^\top \quad \mathbf{C}_{\tau\tau} = \frac{1}{T} \sum_{t=1}^{T-\tau} \mathbf{y}_{t+\tau} \mathbf{y}_{t+\tau}^\top$$

$$\mathcal{L}_{\text{VAMP-2}}(\{\mathbf{y}_t^0, \mathbf{y}_t^\tau\}) = - \left\| (\mathbf{C}_{00})^{-\frac{1}{2}} \mathbf{C}_{0\tau} (\mathbf{C}_{\tau\tau})^{-\frac{1}{2}} \right\|_F^2$$



VAMPnets

Mardt et al. *Nat. Commun.* 9:5 (2018)
 Chen et al. *arXiv:1902.03336* (2019)



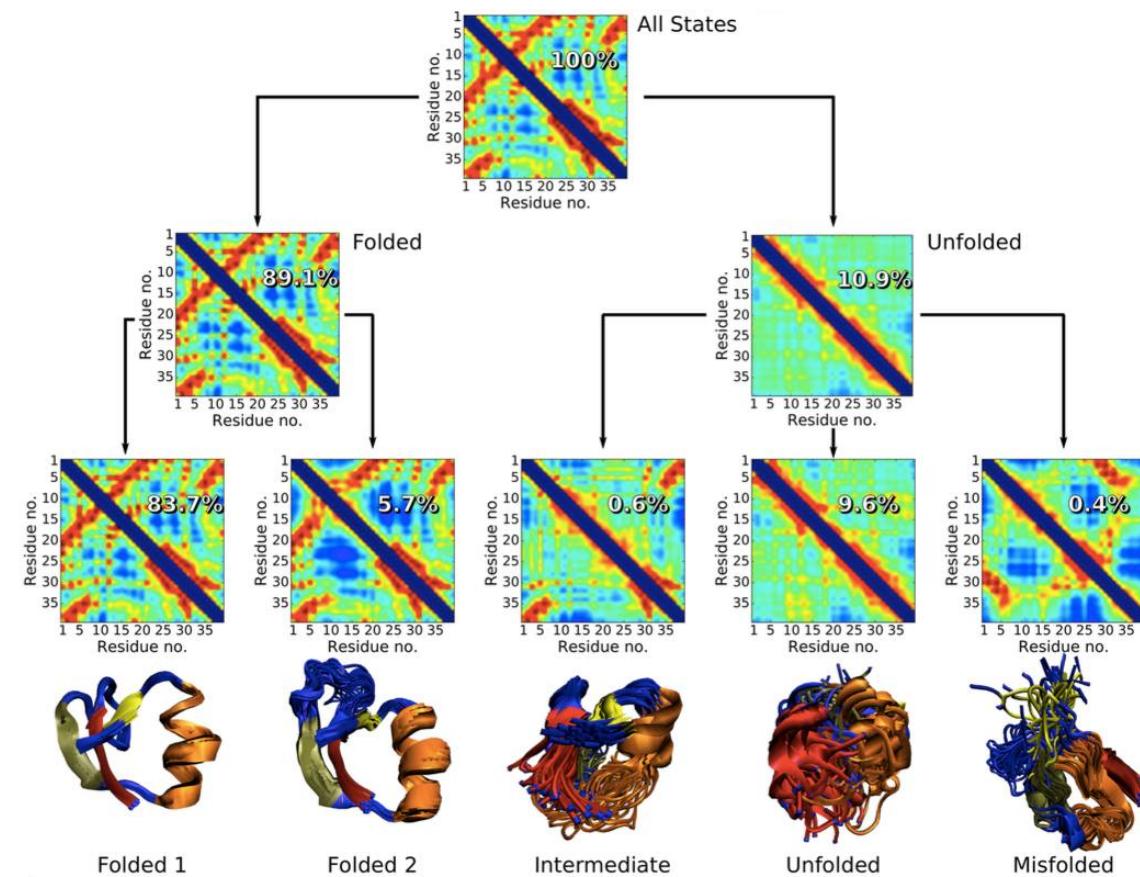
VAMP variational principle (subspace version)

For any two sets of linearly independent functions $\chi_0(\mathbf{x}) = (\chi_{01}(\mathbf{x}), \dots, \chi_{0n}(\mathbf{x}))$ and $\chi_1(\mathbf{x}) = (\chi_{11}(\mathbf{x}), \dots, \chi_{1n}(\mathbf{x}))$, let us call

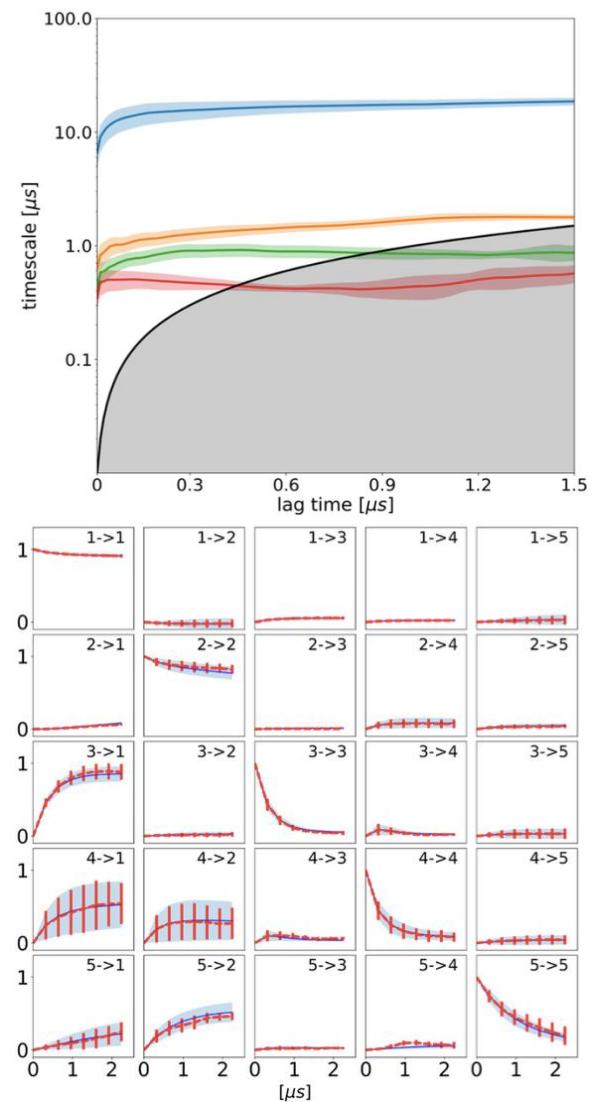
$$\hat{R}_2[\chi_0, \chi_1] = \left\| \mathbf{C}_{00}^{-\frac{1}{2}} \mathbf{C}_{01} \mathbf{C}_{11}^{-\frac{1}{2}} \right\|_F^2$$

their **VAMP-2 score**, where \mathbf{C}_{00} , \mathbf{C}_{01} , \mathbf{C}_{11} are the feature correlation matrices as defined earlier and $\|\cdot\|_F$ indicates the Frobenius norm. The maximum value of the VAMP-2 score is achieved when the top n left and right Koopman singular functions belong to $\text{span}(\chi_0)$ and $\text{span}(\chi_1)$, respectively.

NTL9 Protein folding



Validation



Mardt, Pasquali, Wu, Noé **Nature Communications** (2018)

Kinetics

Open challenges

Transferability of kinetic models?

Quantum Kinetics?

Kinetics of coarse-grained models? (E.g. Spectral matching)

Kinetic Consistency:

first few eigenvalues of original dynamics are preserved:

$$\mathcal{L}^\xi \psi_i^\xi = -\kappa_i^\xi \psi_i^\xi$$

$$\kappa_i \approx \kappa_i^\xi$$

Crommelin & Vanden-Eijnden, Commun. Math. Sci. 4, 651 (2006)

Crommelin & Vanden-Eijnden, SIAM Multiscale Model. Simul. 9, 1588 (2011)

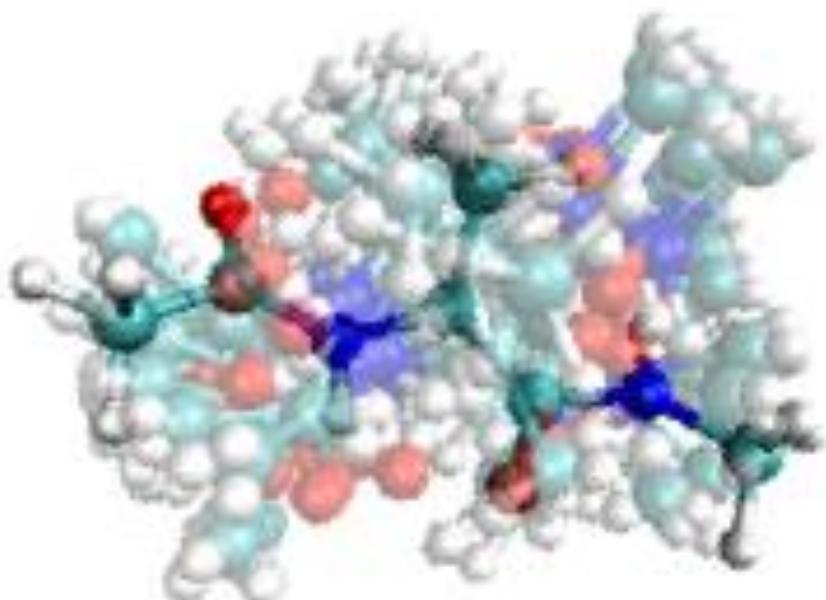
Nüske et al. J. Chem. Phys. 151:044116 (2019)

How we have been doing Molecular Simulation before ML

3. Equilibrium Stat Mech

$$\mathbb{E}[A] = \int A(\mathbf{x}) \mu(\mathbf{x}) d\mathbf{x}$$

Computation of equilibrium observables



If energy function
is available,
generate configurations
according to
Boltzmann distribution
and integrate

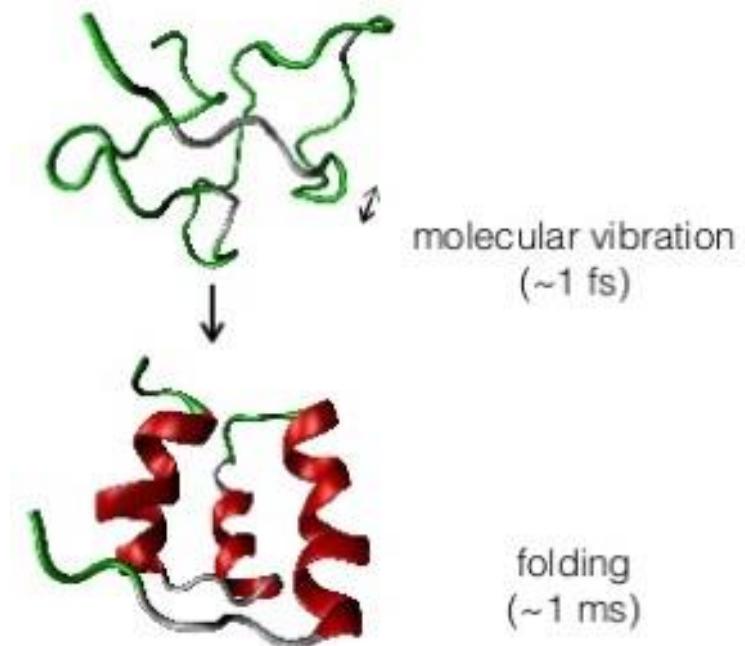
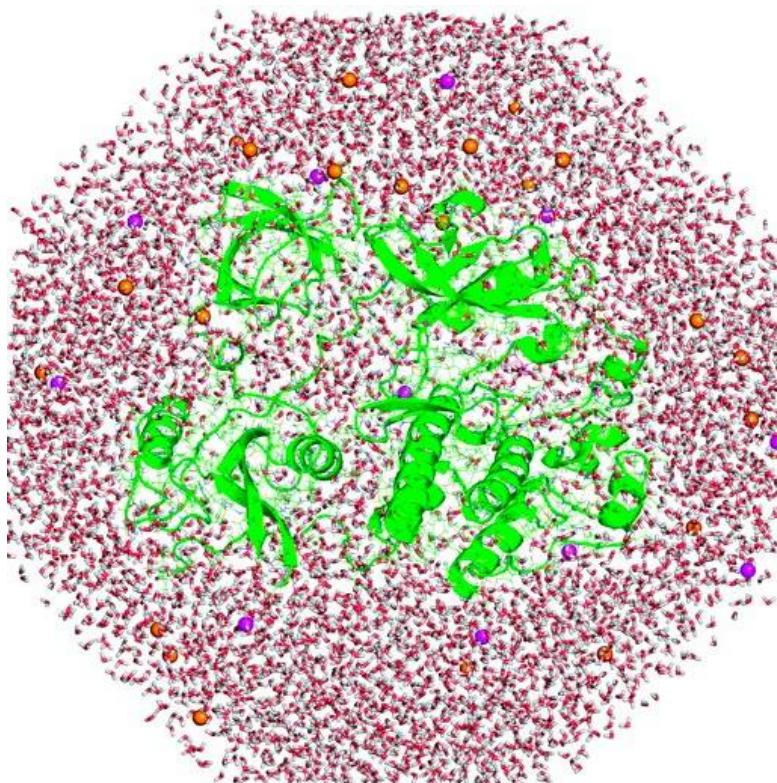
$$\mu(\mathbf{x}) \propto e^{-u(\mathbf{x})}$$

How we have been doing Molecular Simulation before ML

3. Equilibrium Stat Mech

$$\mu(\mathbf{x}) \propto e^{-u(\mathbf{x})}$$

The equilibrium distribution is sampled by generating MD (or MC) trajectories



Rare events sampling problem

Protein-protein association



2 milliseconds molecular dynamics
= 20,000 GPU days

500 Gigajoule



Combustion of a Saturn V rocket
& transport of 50 ton payload to lunar orbit

1500 Gigajoule

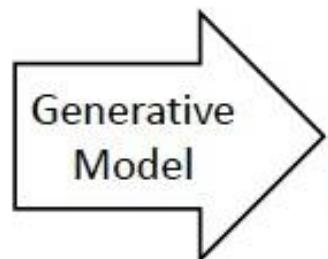


Equilibrium Stat Mech How can we do it with ML

Generative networks?



Random Noise

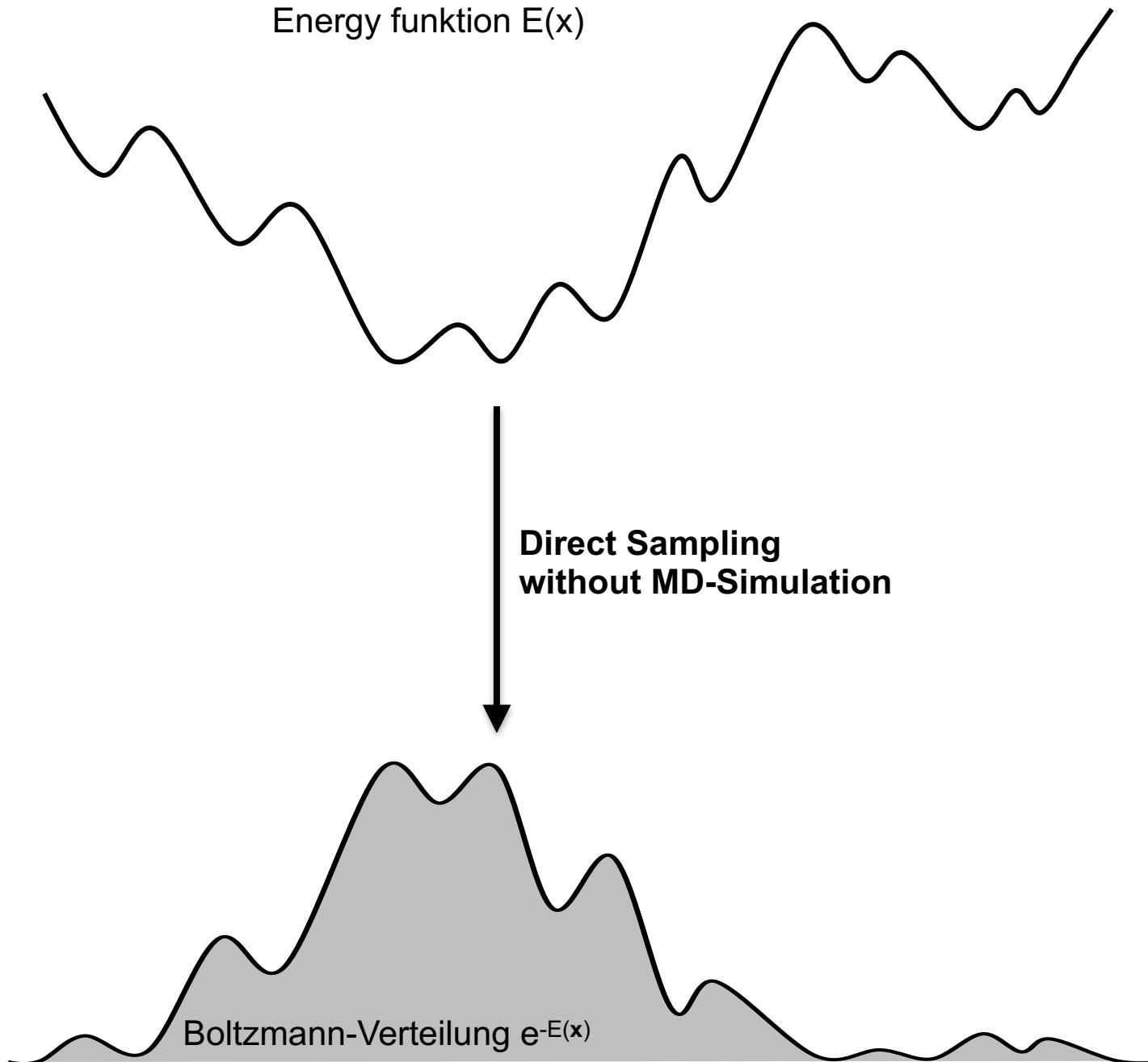


Generative
Model



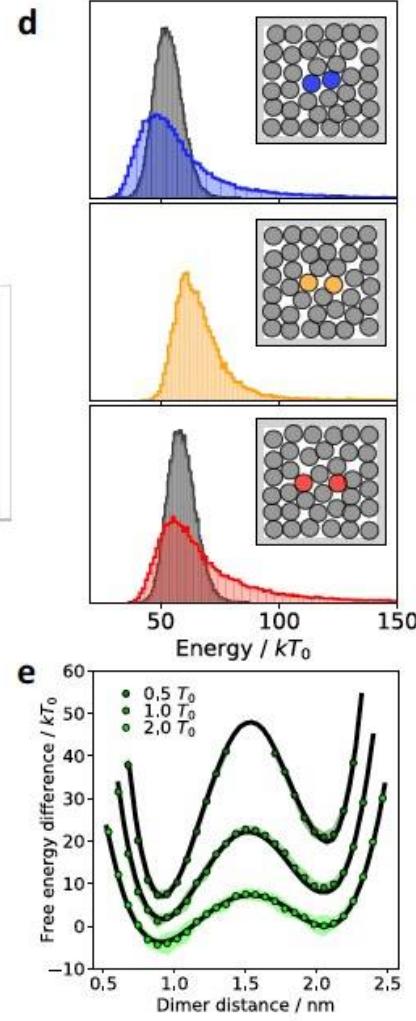
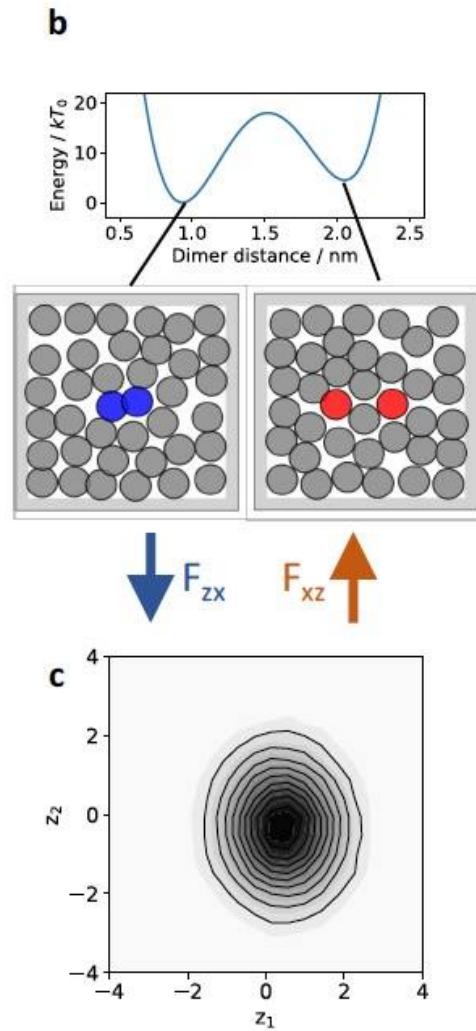
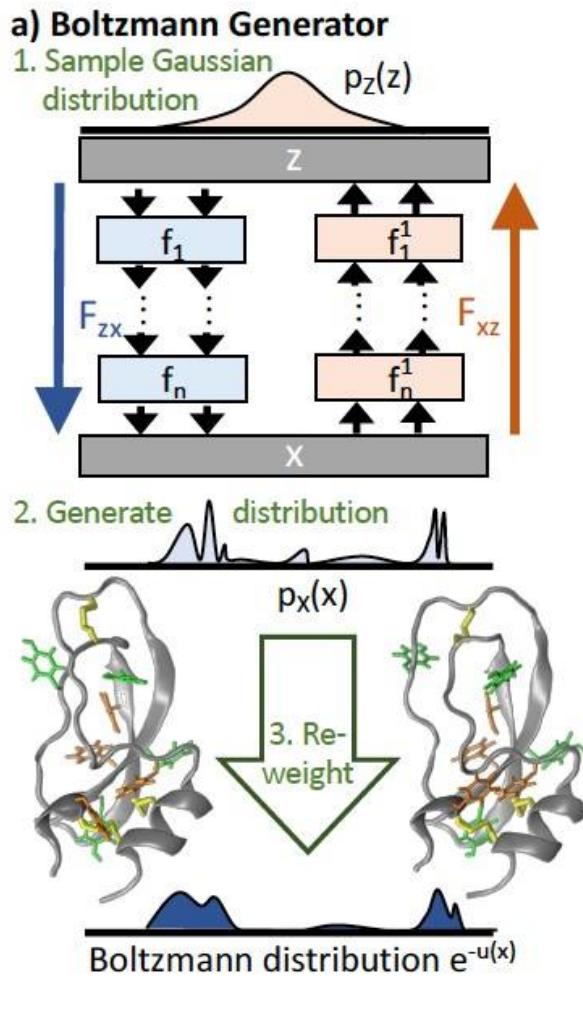
Liu et al. *Proceedings of International Conference on Computer Vision (2015)*

Boltzmann Generators: Learn to sample $x \sim e^{-E(x)}$ directly



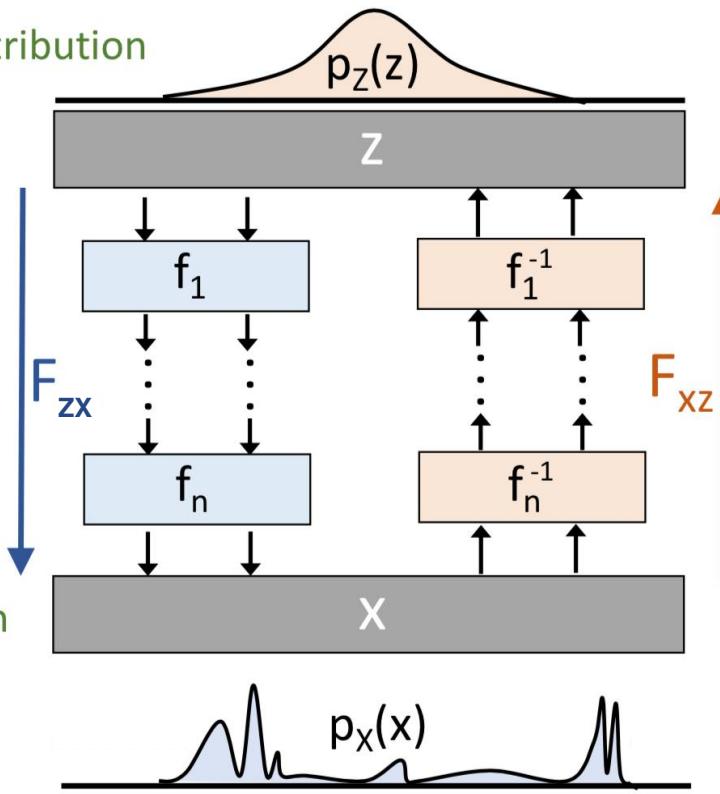
Equilibrium Stat Mech How can we do it with ML

Boltzmann Generators

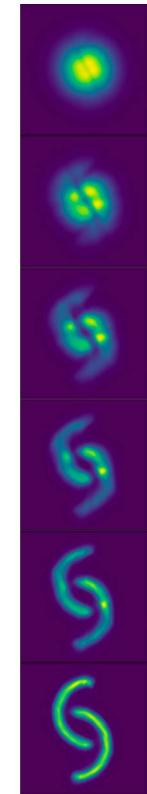


Normalizing Flows

1. Sample Gaussian distribution



2. Generate distribution



Idea: Transformation of random variables for invertible transformations

$$p_X(\mathbf{x}) = p_Z(\mathbf{z}) \left| \det \frac{\partial F_{ZX}}{\partial \mathbf{z}} \right|$$

PDE Flows: Tabak, Vanden-Eijnden, Commun. Math. Sci. 2010

NICE: Dinh, Krueger, Y. Bengio, ICLR 2015

Normalizing Flows: Rezende, Mohamed, ICML 2015

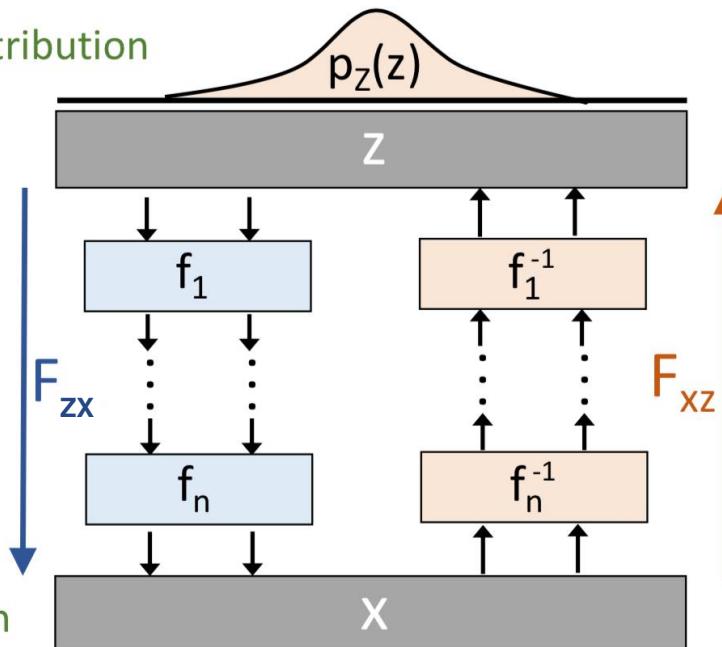
RealNVP: Dinh, Sohl-Dickstein, S. Bengio, ICLR 2017

PixelRNN: van den Oord et al, ICML 2016

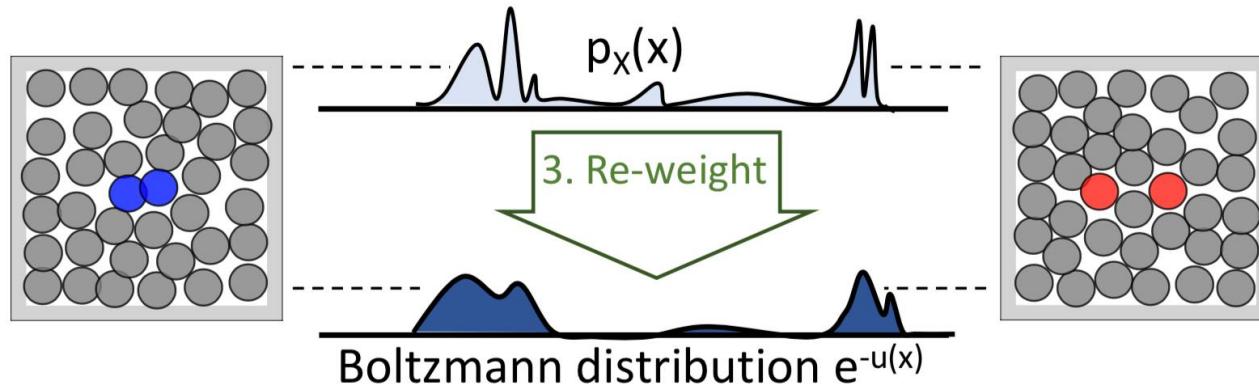
***Neural ODEs:** Chen et Al, NeurIPS 2018

Boltzmann Generator

1. Sample Gaussian distribution



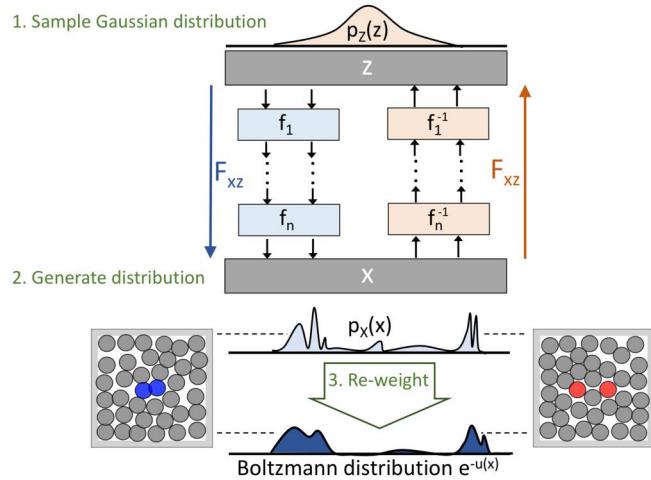
2. Generate distribution



Boltzmann Generators learn to sample
from **target distribution $e^{-u(x)}$ without bias**

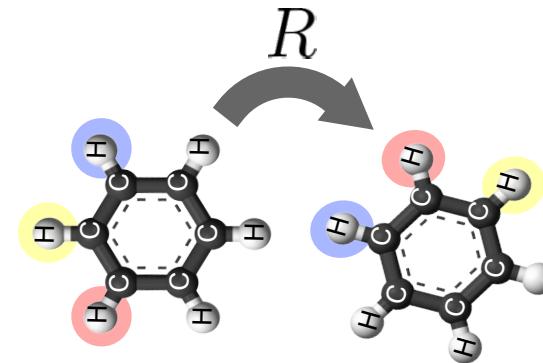
Overview: Boltzmann-Generating Flows

Boltzmann Generators



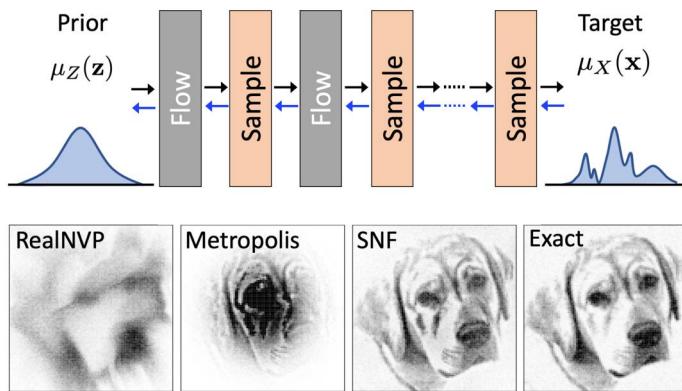
Noé, Olsson, Köhler, Wu,
Science 365: eaww1147 (2019). arXiv:1812.01729

Equivariant Flows



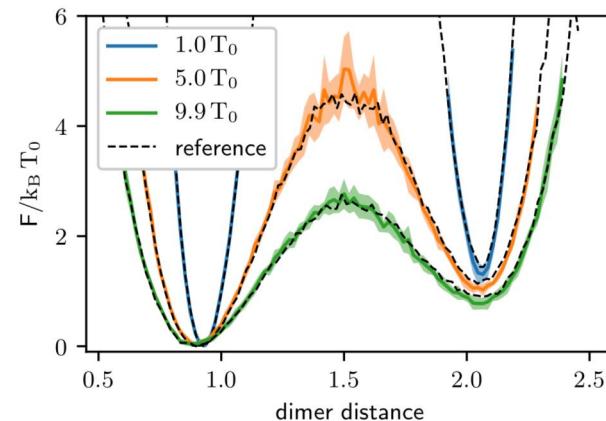
Jonas Köhler, Leon Klein, Frank Noé
ICML 2020 arXiv:2006.02425

Stochastic Normalizing Flows



Hao Wu, Jonas Köhler, Frank Noé
NeurIPS 2020 arXiv:2002.06707

Temperature-steerable Flows



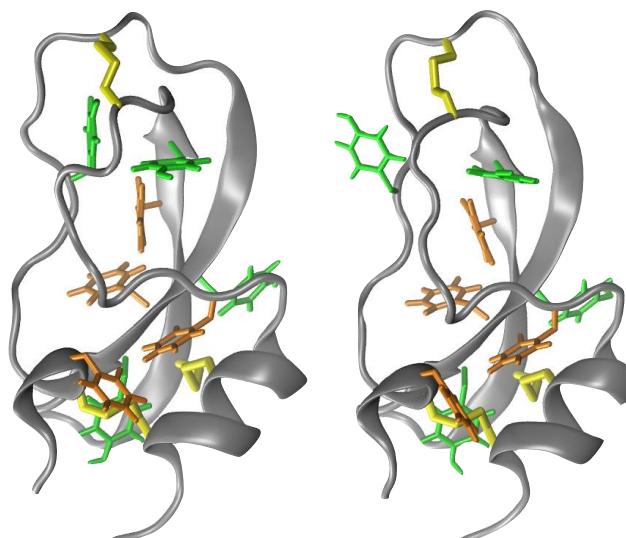
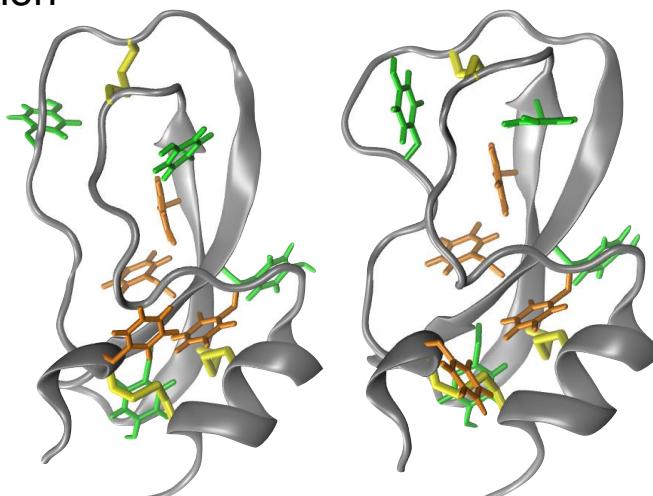
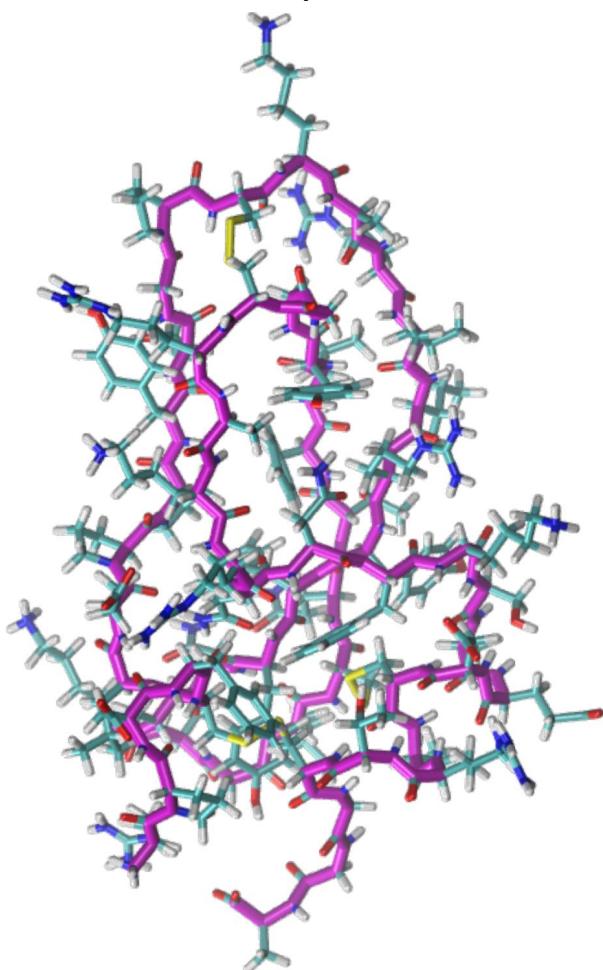
Manuel Dibak, Leon Klein, Frank Noé
NeurIPS ML4Physics workshop 2020

Boltzmann Generator

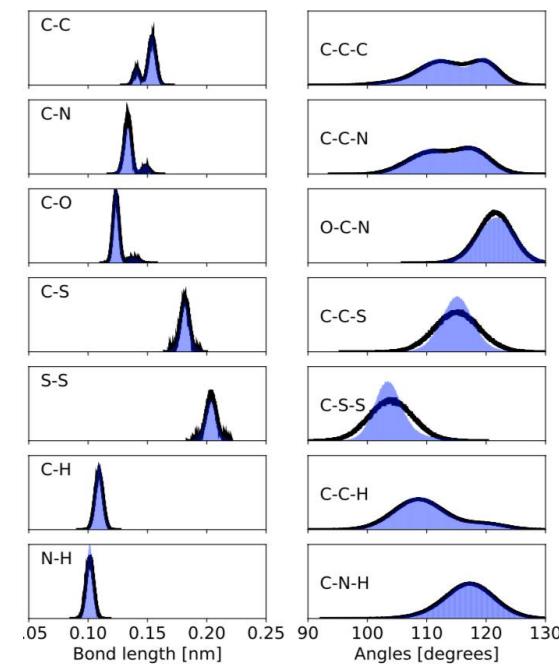
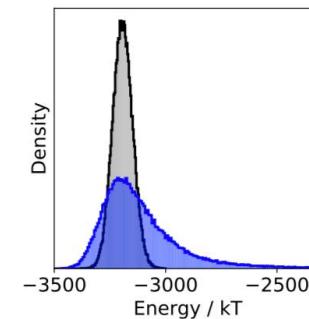
3000-dim Gaussian distribution



1000 atom positions



Statistics



Equilibrium Stat Mech Open challenges

- Scaling to larger systems
- Explicit solvent (indistinguishability)
- Parameter sharing?
- Transferability?

Beyond “low-hanging fruits”

- Can we use **explainable AI** methods to gain scientific insights into the mechanisms that give rise to the predicted physicochemical quantity?
- Can we use ML to discover new organizing principles and develop new theories?

Discovering governing equations

E.g., SINDy Autoencoders

