Title: Solving physics many-body problems with deep learning

Speakers: Frank Noe

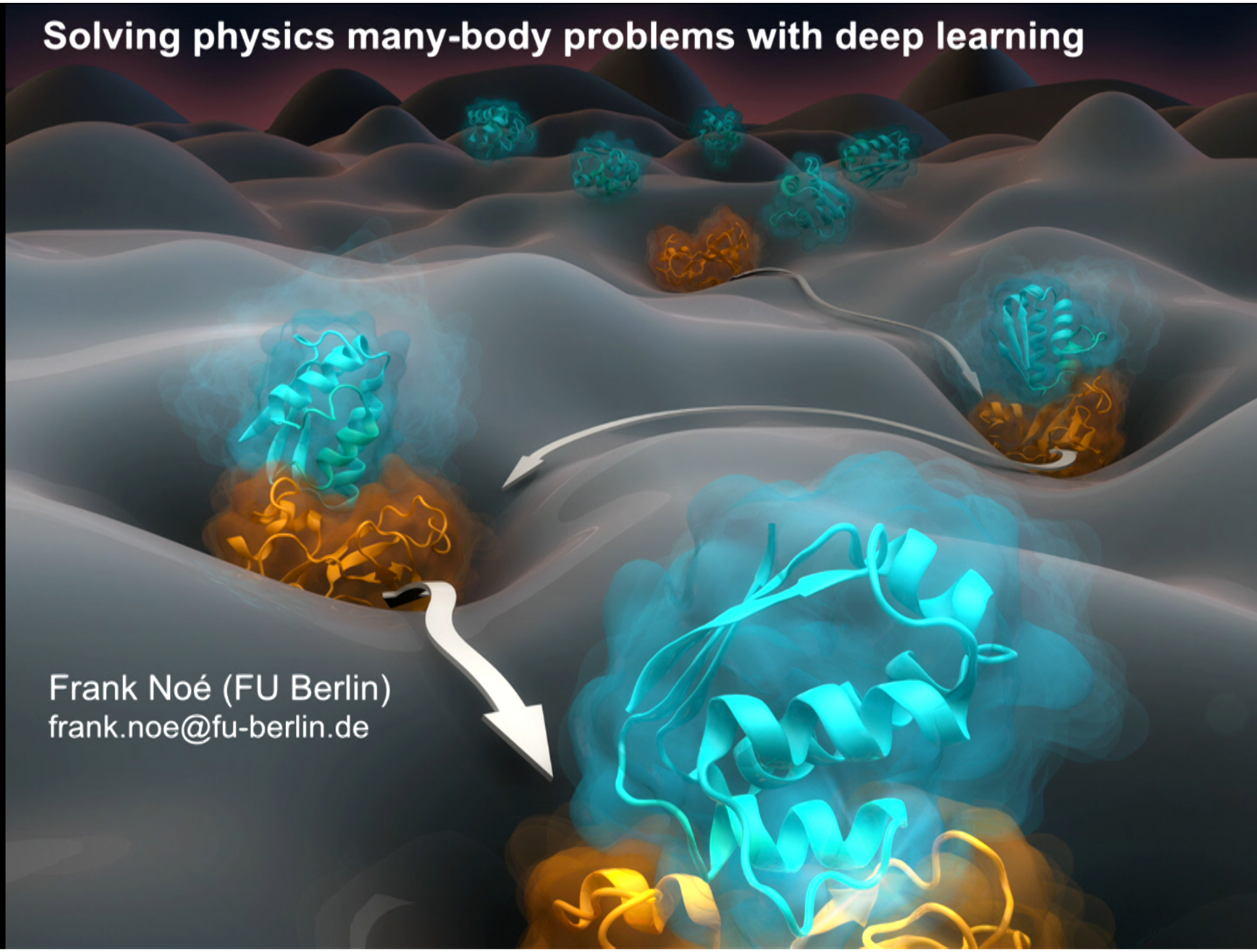
Series: Machine Learning Initiative

Date: November 12, 2019 - 1:00 PM

URL: http://pirsa.org/19110081

Abstract:

Solving classical and quantum physics many-body systems are amongst the hardest problems in the natural sciences, but also of fundamental importance for applications such as material and drug design. In this talk, I will give a an overview of fundamental physics problems at multiple time- and lengthscales and describe deep learning methods to address them: 1) solving the quantum-chemical electronic SchrÃ¶dinger equation with deep variational Monte Carlo, 2) learning to coarse-grain many-body systems, and 3) sampling equilibrium states of classical many-body systems with generative learning.


Solving physics many-body problems with deep learning
Frank Noé (FU Berlin)
frank.noe@fu-berlin.de

Statistical Mechanics

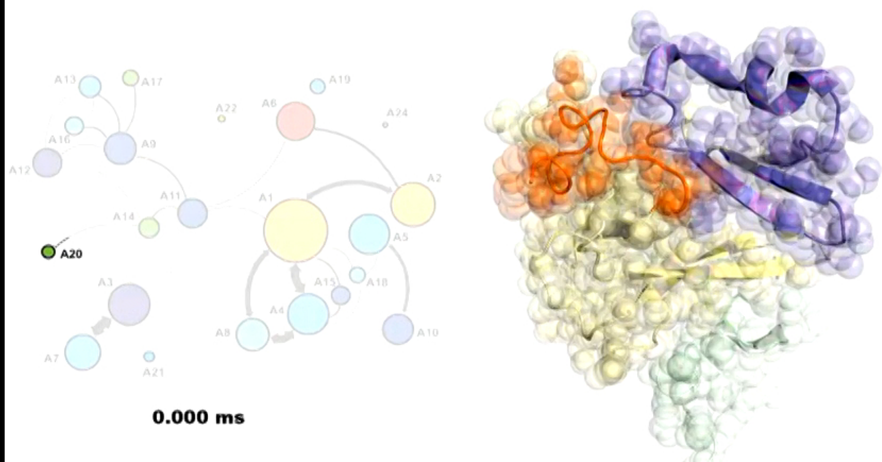
$$\mathbf{x} \sim \mathrm{e}^{-E(\mathbf{x}_1,\ldots,\mathbf{x}_N)}$$

Sampling problem

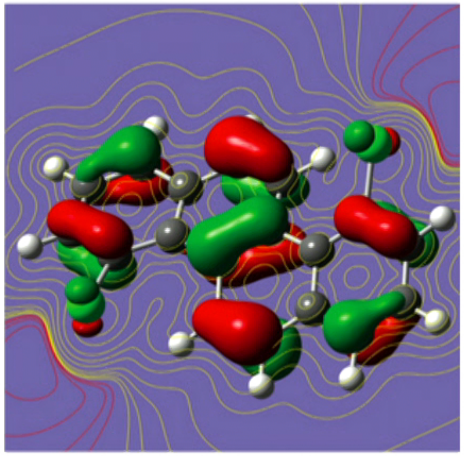
Quantum Mechanics

$$\hat{H}\psi = E_{\mathbf{x}}\psi$$

Electronic structure problem



Video from: Wiewora et al, eLife 8:e45403 (2019)



Statistical Mechanics

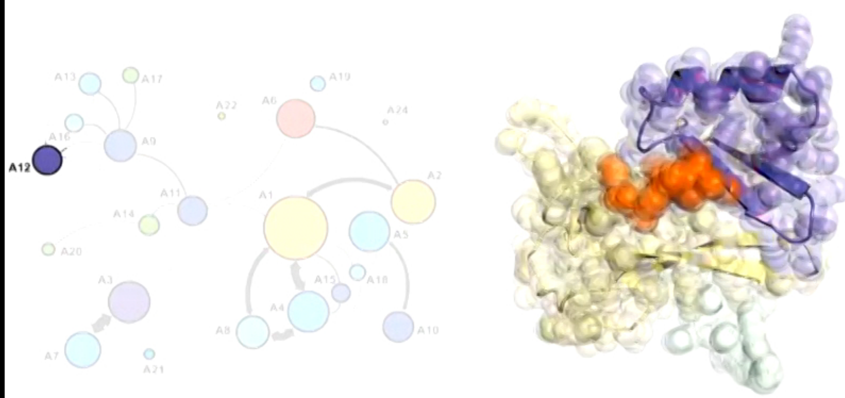
$$\mathbf{x} \sim e^{-E(\mathbf{x}_1, \ldots, \mathbf{x}_N)}$$

Sampling problem

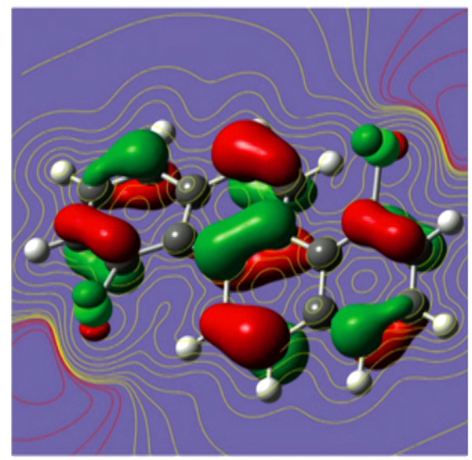
Quantum Mechanics

$$\hat{H}\psi = E_{\mathbf{x}}\psi$$

Electronic structure problem



Video from: Wiewora et al, eLife 8:e45403 (2019)



Statistical Mechanics

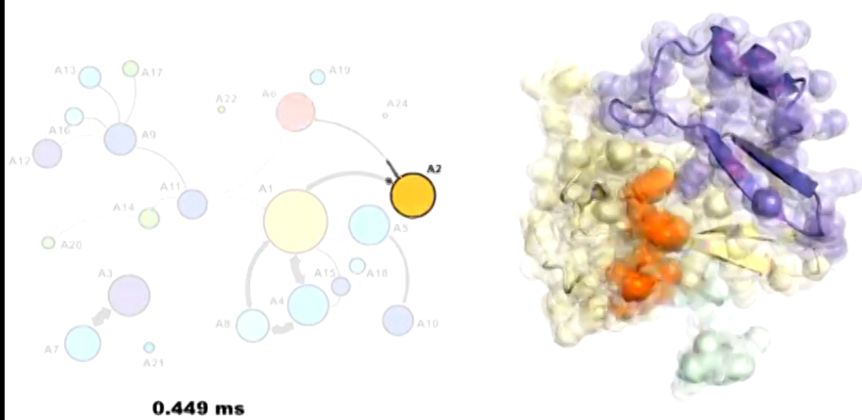
$$\mathbf{x} \sim \mathrm{e}^{-E(\mathbf{x}_1,\ldots,\mathbf{x}_N)}$$

Sampling problem

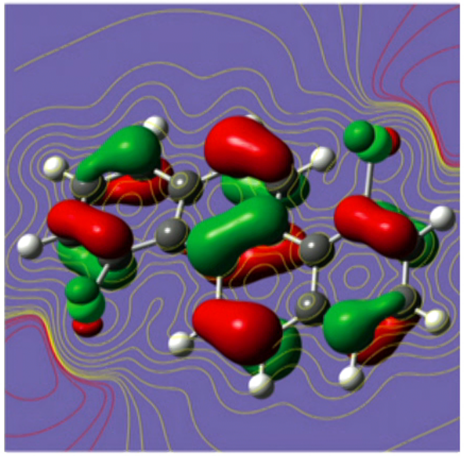
Quantum Mechanics

$$\hat{H}\psi = E_{\mathbf{x}}\psi$$

Electronic structure problem



Video from: Wiewora et al, eLife 8:e45403 (2019)



Statistical Mechanics

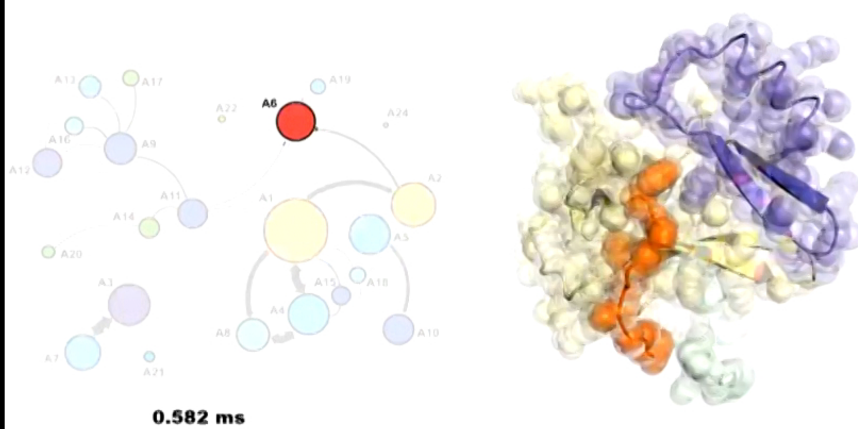
$$\mathbf{x} \sim \mathrm{e}^{-E(\mathbf{x}_1,\ldots,\mathbf{x}_N)}$$

Sampling problem

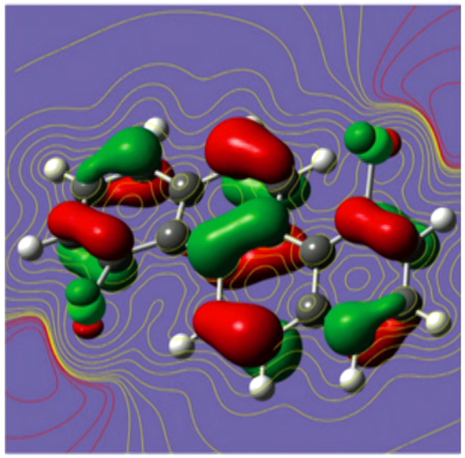
Quantum Mechanics

$$\hat{H}\psi = E_{\mathbf{x}}\psi$$

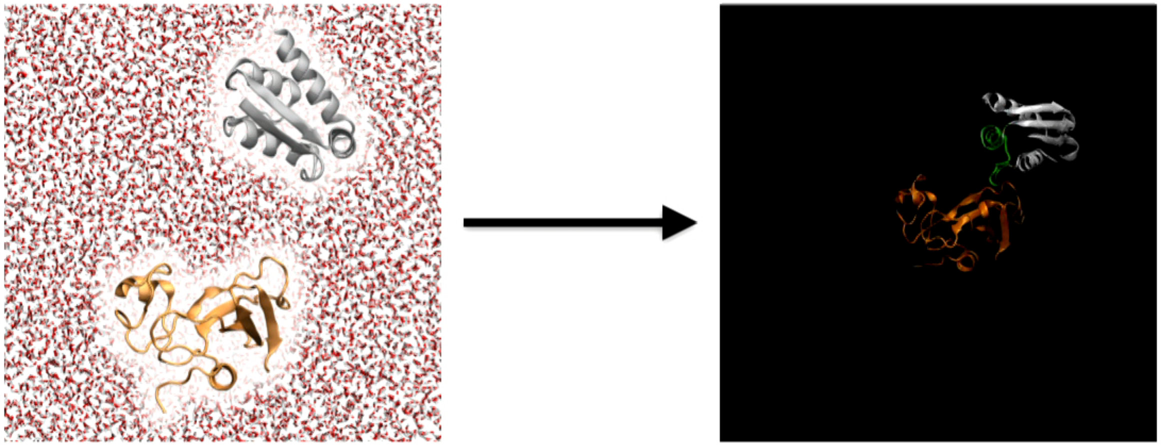
Electronic structure problem



Video from: Wiewora et al, eLife 8:e45403 (2019)



Samping problem of many-body physics systems



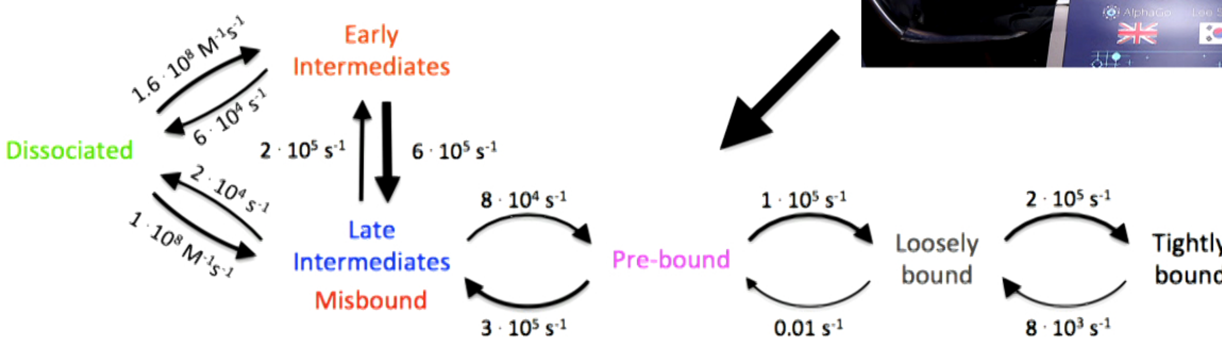
Microsecond
MD Trajectories

Samping problem of many-body physics systems

Plattner, Doerr, De Fabritiis, Noé
Nature Chemistry 9, 1005 (2017)

total 2 ms data

Reinforcement learning



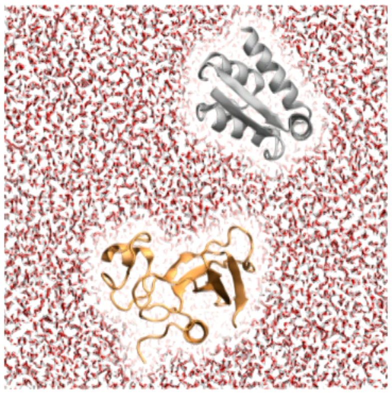
Adaptive Markov State Model — **seconds to hours kinetics**

Freie Universität Berlin

Samping problem of many-body physics systems

Standard simulation methods in molecular physics are INSANELY expensive

Current approach



very long time

a lot of energy

2 ms MD = 20,000 GPU days
500 gigajoule

PhD

1 non-transferable model

Burn a Saturn V rocket and deliver 50 ton payload to lunar orbit

1500 gigajoule

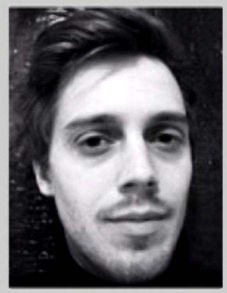
Freie Universität Berlin

Thermodynamics **Sampling problem**

Boltzmann Generators
sampling equilibrium states of many-body systems with deep learning

Noé, Olsson, Köhler, Wu, **Science** 365: eaww1147 (2019)

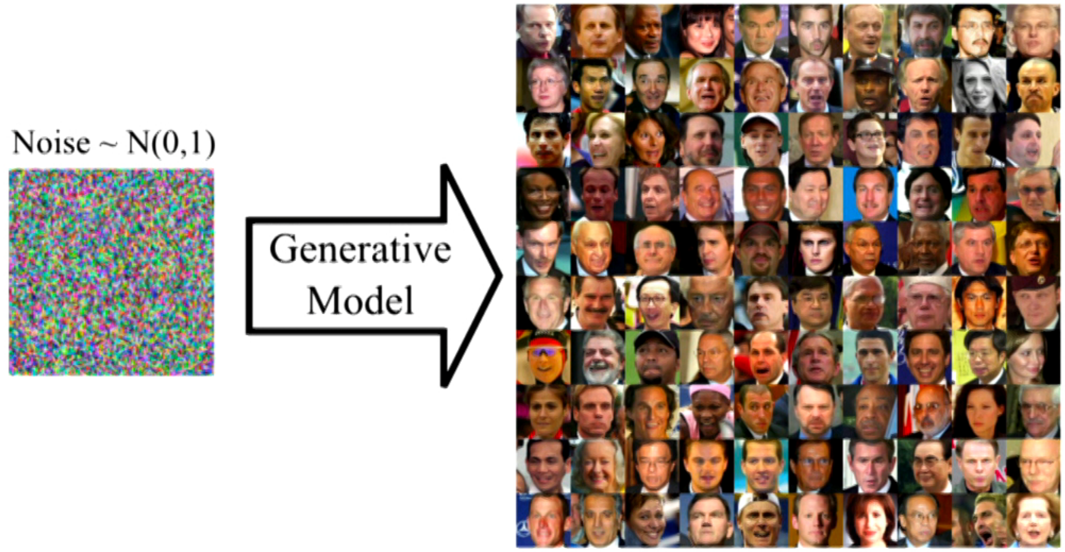
Simon Olsson



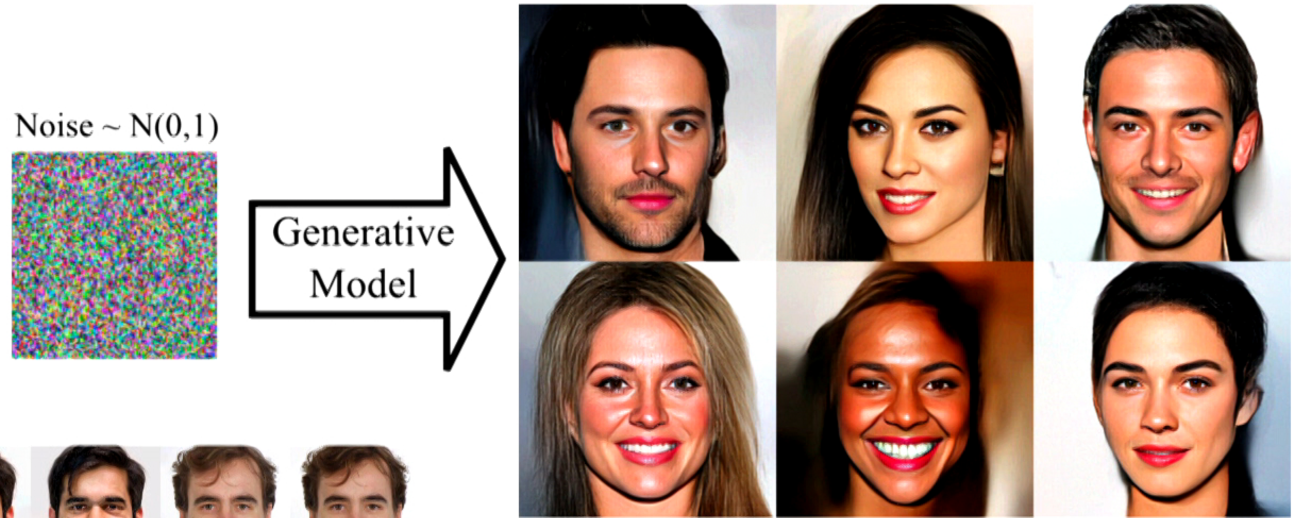
Jonas Köhler

Hao Wu

Directed generative models



Glow

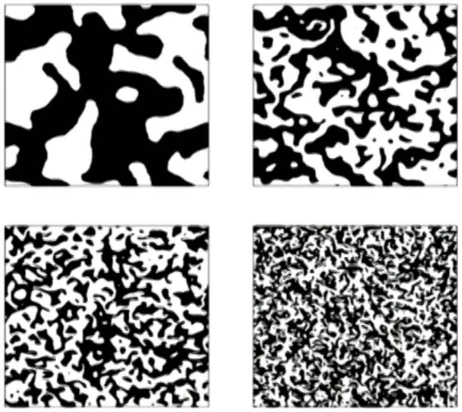


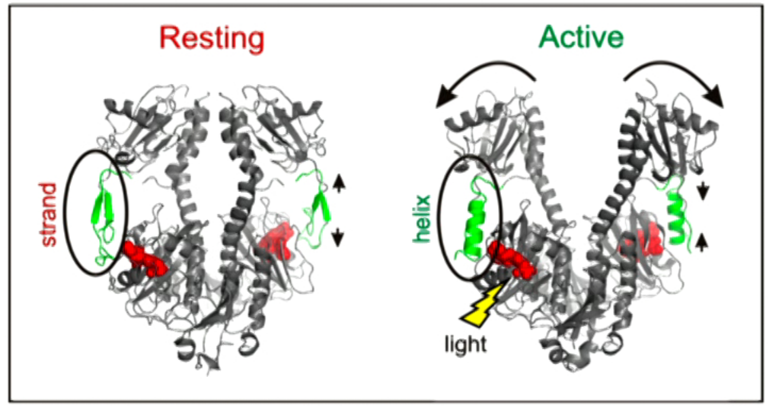
Interpolations in latent space

Kingma, Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. NeurIPS 2018

Freie Universität Berlin

Boltzmann Generators





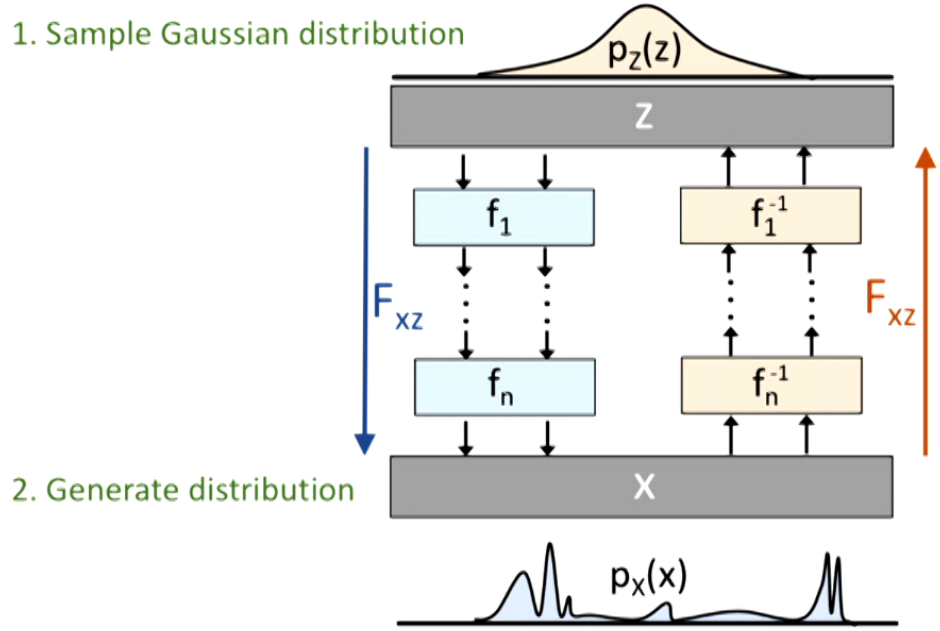
- **Input**: Reduced Potential Energy $u(\mathbf{x})$ in coordinates $\mathbf{x} \in \mathbb{R}^n$, e.g. $u(\mathbf{x}) = U(\mathbf{x})/k_B T$ (canonical ensemble).
- **Aim**: Generate *independent* Samples from Equilibrium Distribution.

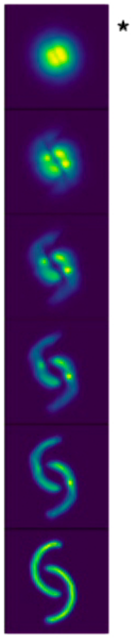
$$\mu(\mathbf{x}) \propto \mathrm{e}^{-u(\mathbf{x})}$$

- **Problem**:
  - Direct MC (proposal for all n degrees of freedom + rejection or reweighting) not available.
  - Standard approach: MD/MCMC with local moves $\rightarrow$ sampling problem.

Freie Universität Berlin

Flows and Normalizing Flows





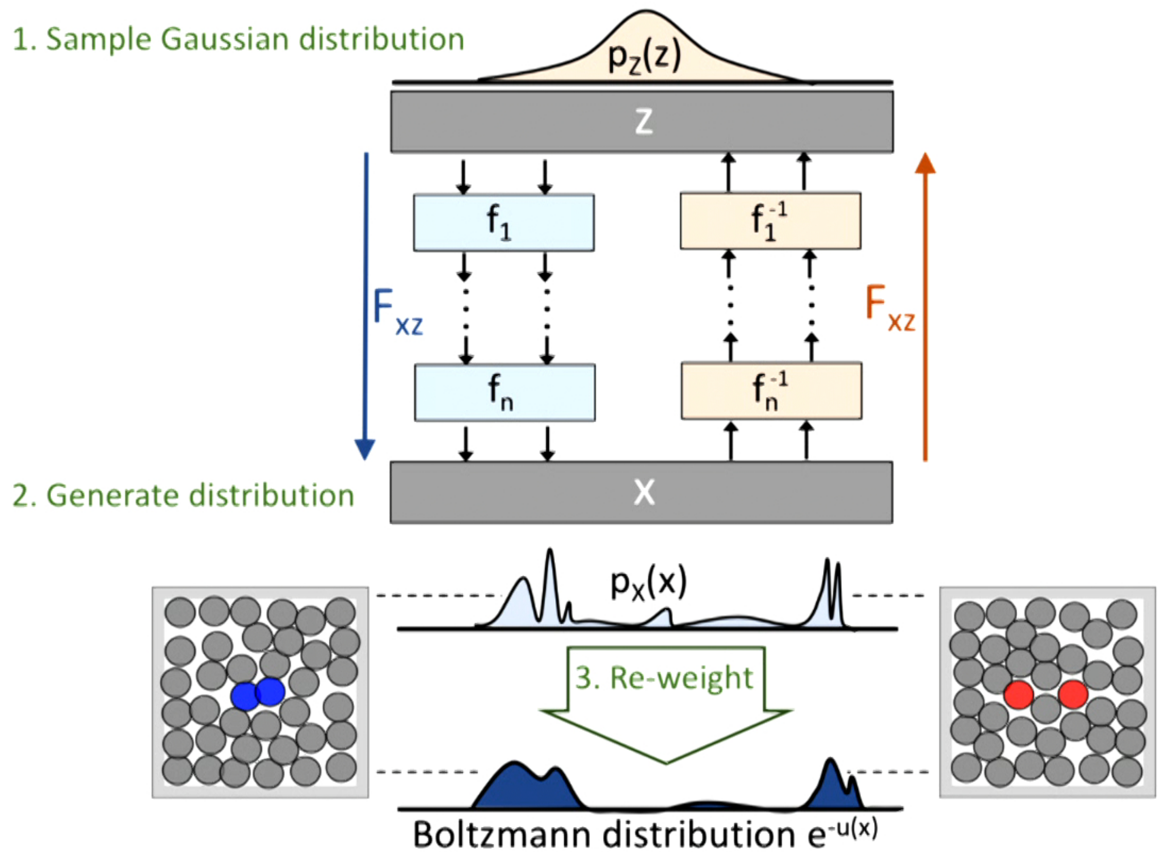
Main idea: Transformation of random variables for bijective transformations:

$$p_X(\mathbf{x}) = p_Z(f(\mathbf{x})) \left| \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}^\top} \right|$$

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

Freie Universität Berlin

Boltzmann Generators: Exact probability generator + reweighting

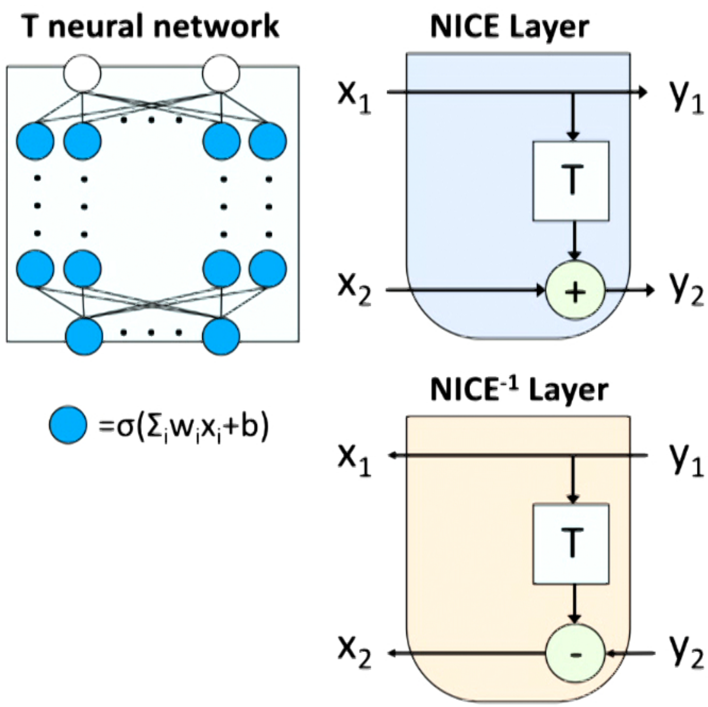


Boltzmann Generators learn to generate **unbiased** samples from target distribution $e^{-u(\mathbf{x})}$

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

Freie Universität Berlin

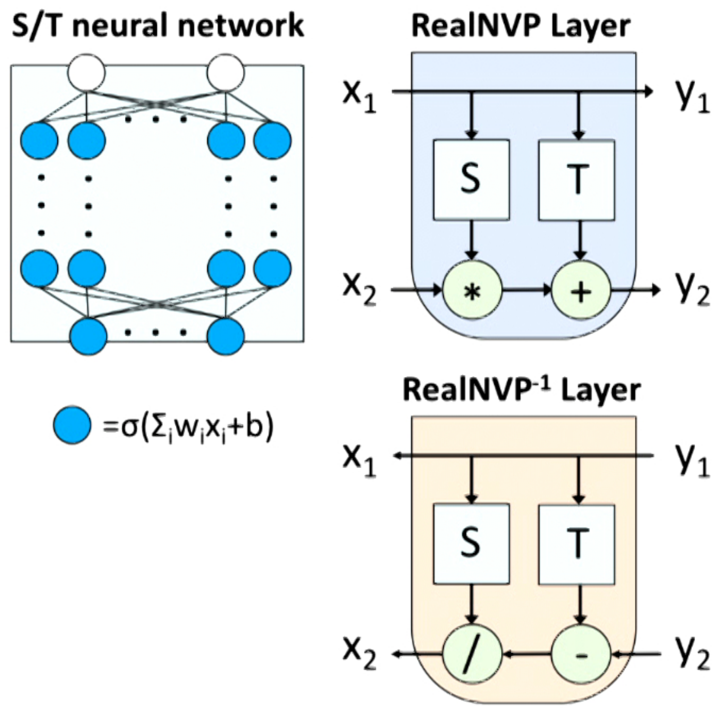
Invertible Networks



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

Freie Universität Berlin

Invertible Networks



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

Freie Universität Berlin

Boltzmann Generators: training with variational free energy

1. Train with **known** energy u(**x**)
   → Minimize KL divergence between generated and Boltzmann distribution

$$J_{KL} = \mathbb{E}_{\mathbf{z}\sim\mu_Z(\mathbf{z})}\left[u(F_{zx}(\mathbf{z};\theta)) - \log\left|\mathbf{J}_{zx}(\mathbf{z};\theta)\right|\right]$$

↑ Sample Gaussian — ↑ Energy of generated samples — ↑ Jacobian (Entropy) of generated samples

2. Maximum generator likelihood of training data **x**
   → Minimize KL divergence between latent and Gaussian distribution

$$\begin{aligned} J_{ML} &= -\mathbb{E}_{\mathbf{x}\sim\rho(\mathbf{x})}\left[\log p_X(\mathbf{x};\theta)\right] \\ &= \mathbb{E}_{\mathbf{x}\sim\rho(\mathbf{x})}\left[\frac{1}{\sigma^2}\left\|F_{xz}(\mathbf{x};\theta)\right\|^2 - \log\left|\mathbf{J}_{xz}(\mathbf{x};\theta)\right|\right] \end{aligned}$$

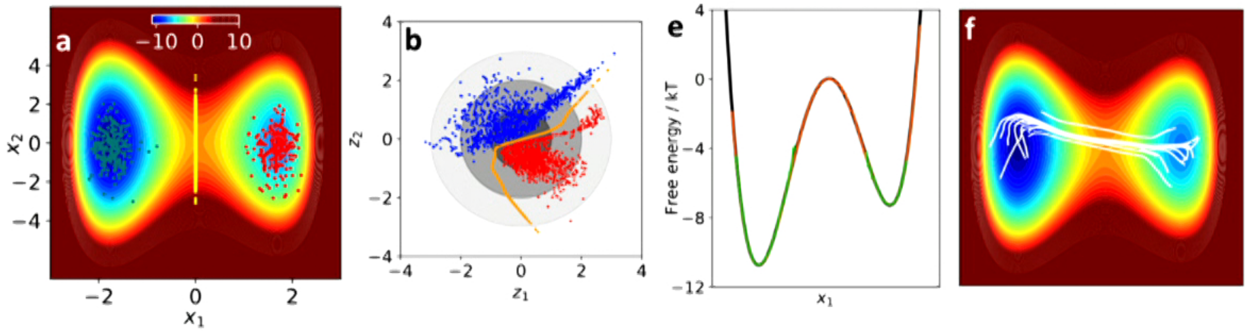
↑ Sample Data — ↑ Energy in Harmonic oscillator — ↑ Jacobian (Entropy) of data samples

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

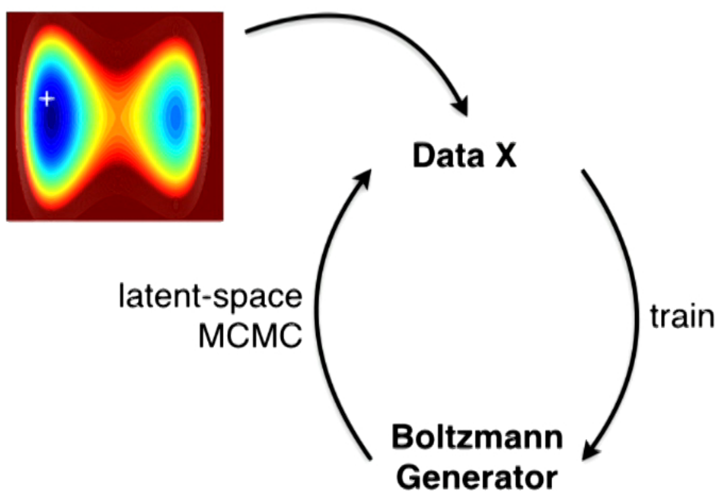
Freie Universität Berlin

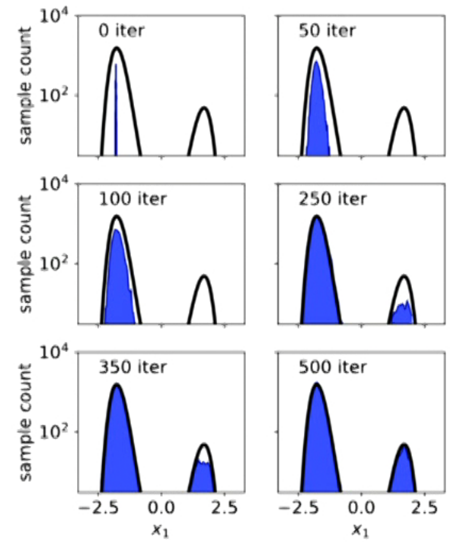
Boltzmann Generators: Double well

Train with data from multiple states:



Adaptive Exploration from one configuration:

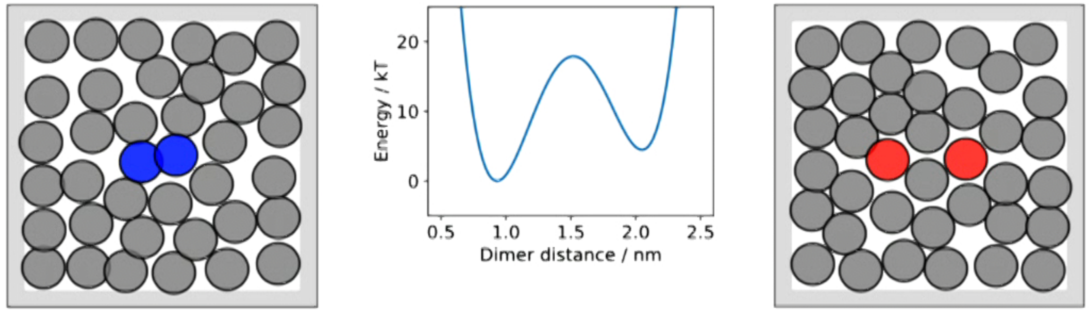




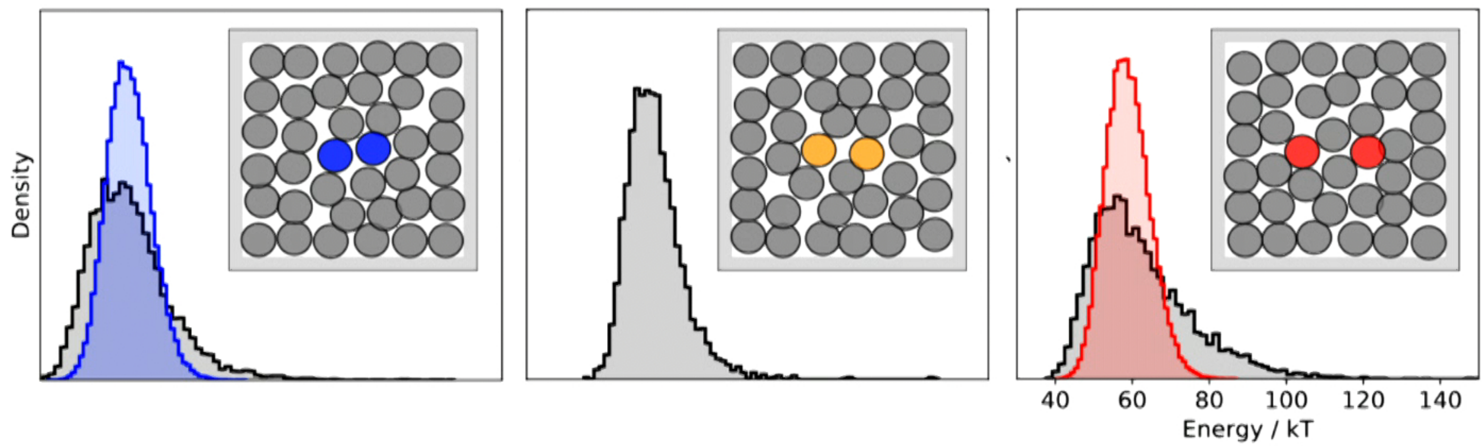
Noé, Olsson, Köhler, Wu, **Science** 365: eaww1147 (2019). arXiv:1812.01729

Freie Universität Berlin

Boltzmann Generators: Bistable dimer in dense colloid solvent



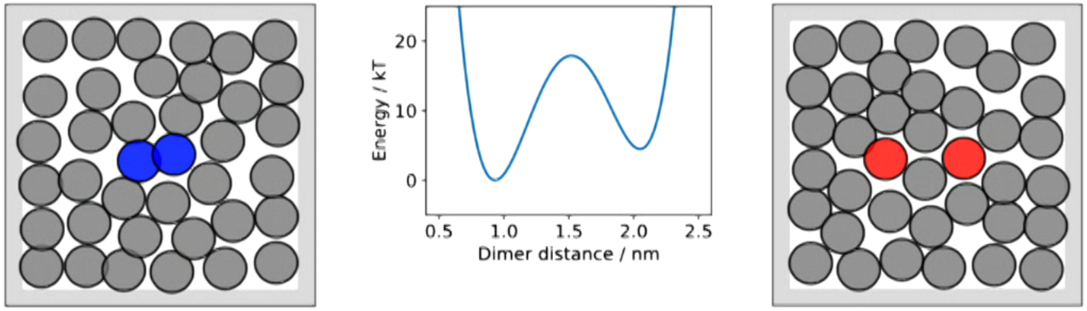
Boltzmann-Generated samples:



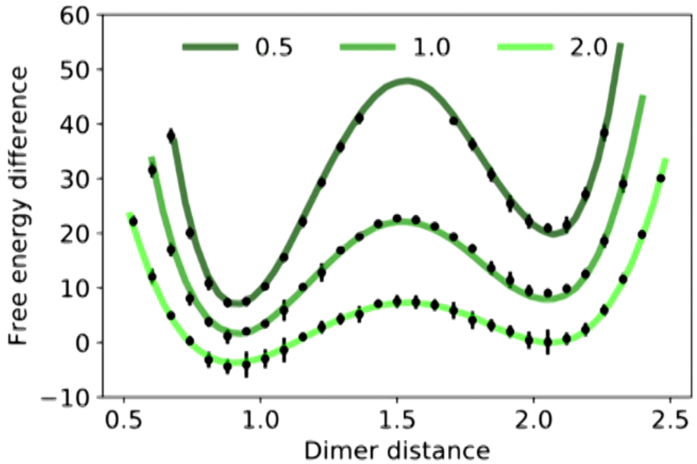
Noé, Olsson, Köhler, Wu, **Science** 365: eaww1147 (2019). arXiv:1812.01729

Freie Universität Berlin

Boltzmann Generators: Bistable dimer in dense colloid solvent



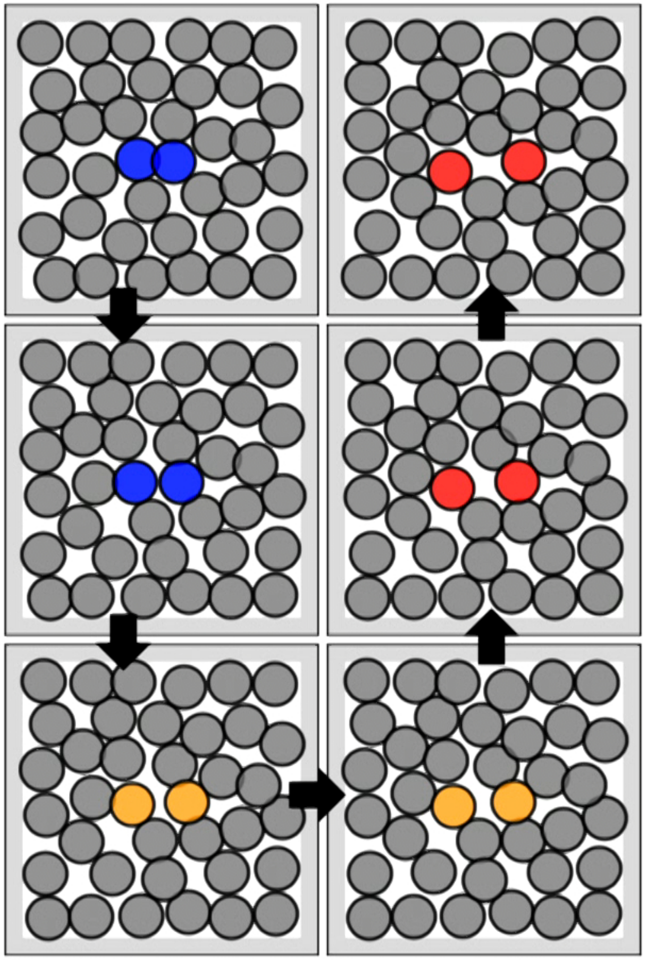
Boltzmann-Generated free energy:



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

Freie Universität Berlin

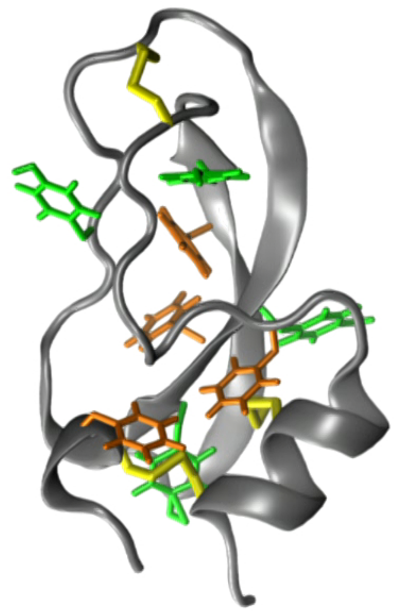
Boltzmann Generators: Bistable dimer in dense colloid solvent



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

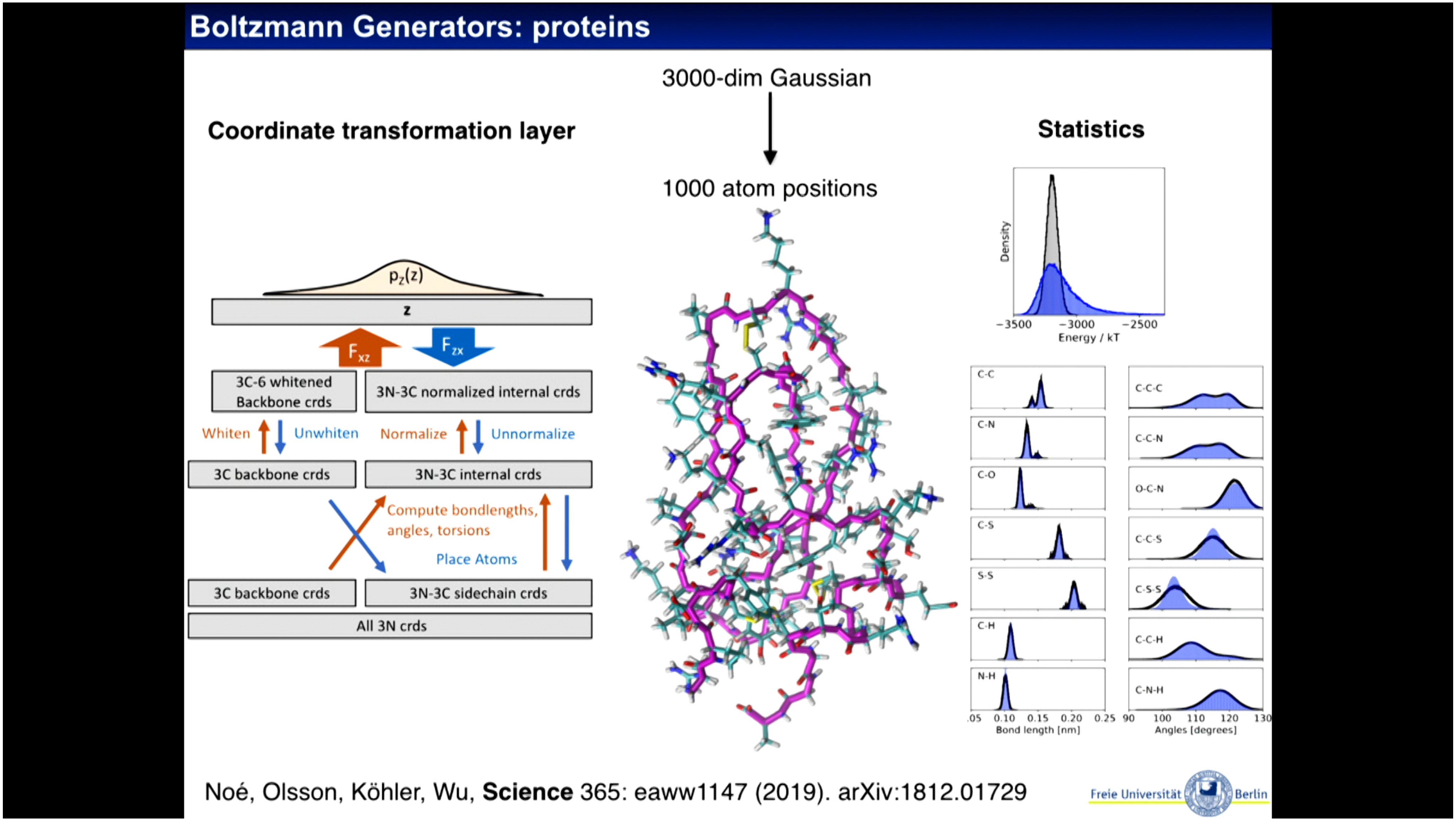
Freie Universität Berlin

Boltzmann Generators: towards proteins



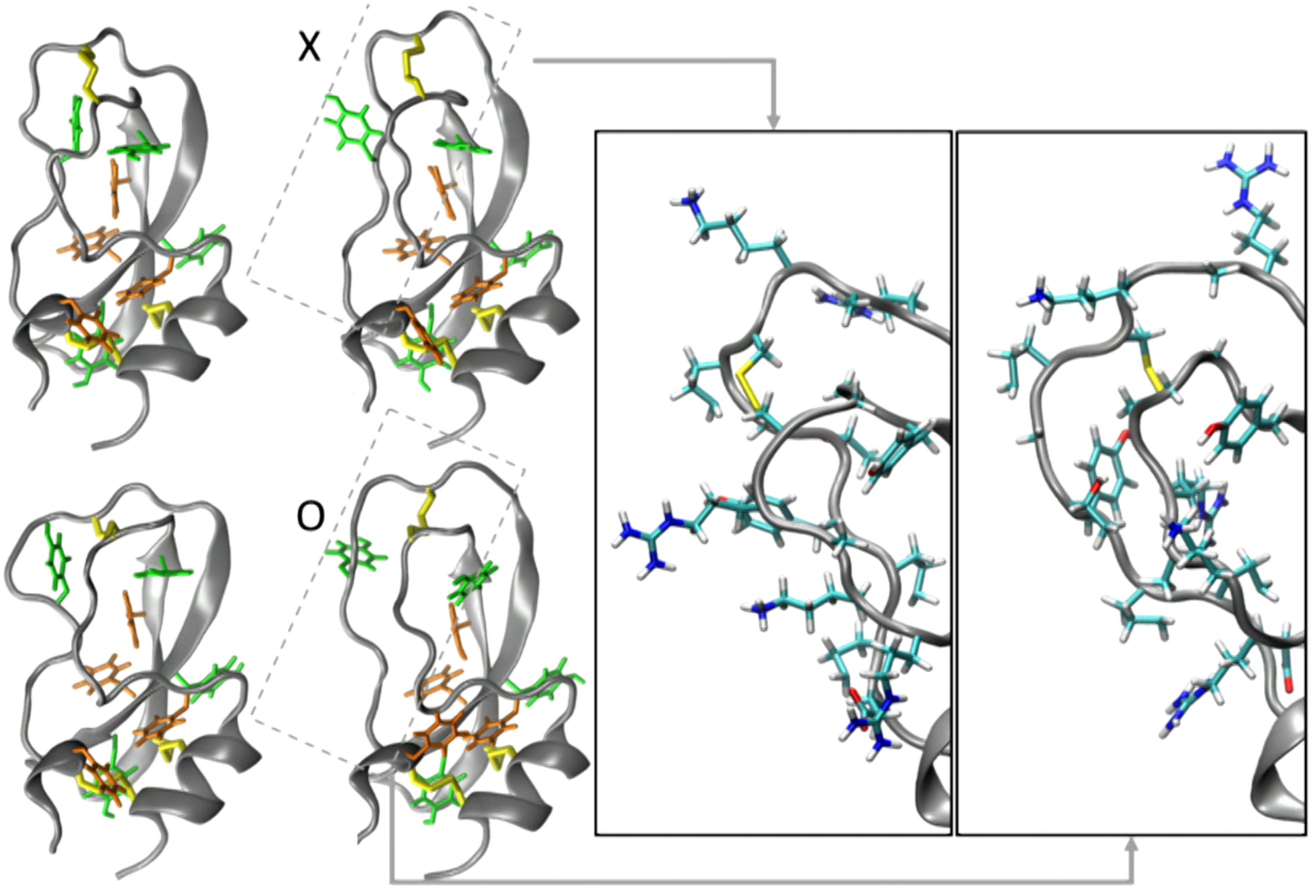
Noé, Olsson, Köhler, Wu, **Science** 365: eaww1147 (2019). arXiv:1812.01729

Freie Universität Berlin



Boltzmann Generators: proteins
Coordinate transformation layer
$p_z(z)$
z
F_{xz}
F_{zx}
3C-6 whitened Backbone crds
3N-3C normalized internal crds
Whiten
Unwhiten
Normalize
Unnormalize
3C backbone crds
3N-3C internal crds
Compute bondlengths, angles, torsions
Place Atoms
3C backbone crds
3N-3C sidechain crds
All 3N crds
3000-dim Gaussian
1000 atom positions
Statistics
Density
Energy / kT
C-C
C-N
C-O
C-S
S-S
C-H
N-H
Bond length [nm]
C-C-C
C-C-N
O-C-N
C-C-S
C-S-S
C-C-H
C-N-H
Angles [degrees]
Noé, Olsson, Köhler, Wu, **Science** 365: eaww1147 (2019). arXiv:1812.01729
Freie Universität Berlin

Boltzmann Generators: proteins



1 millisecond MD
= 1 year on Anton Supercomputer

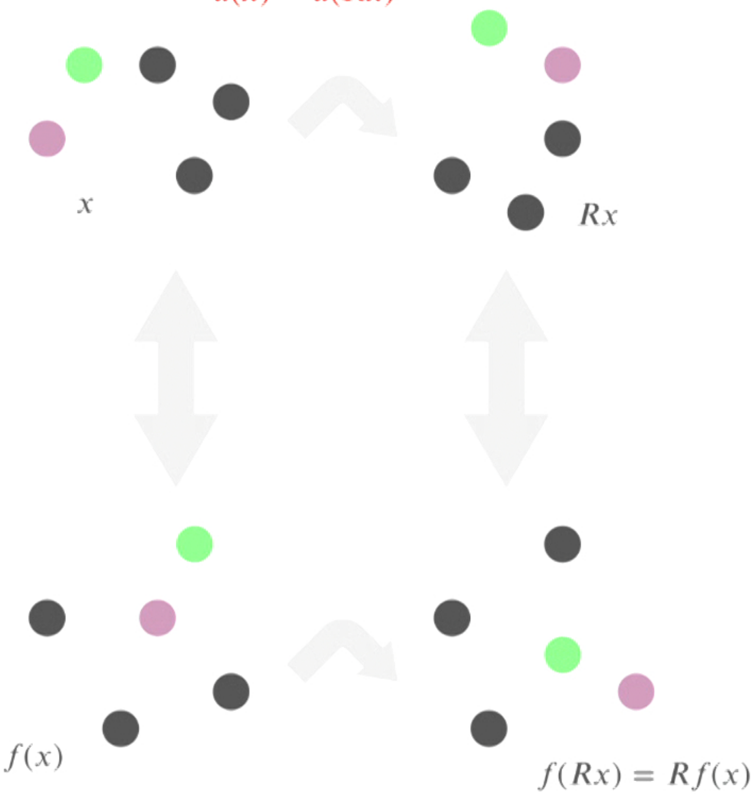
Boltzmann Generator
5 hours training on 1 GPU

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

Freie Universität Berlin

Equivariant Boltzmann Generators

$$u(x) = u(Rx)$$



$$\log p(f(x) = \log p(f(Rx)) = \log p(R(fx))$$

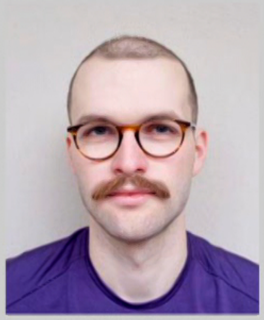
Köhler, Klein, Noé: NeurIPS workshop ML and the physical sciences, 2019

Freie Universität Berlin

Quantum Mechanics **Electronic structure problem**

Deep neural network solution of the electronic Schrödinger Equation

Hermann, Schätzle, Noé, **arXiv** 1909.08423 (2019)



Jan Hermann

Zeno Schätzle

Electronic Schrödinger Equation

- **Electronic Schrödinger Equation**:

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

  - $\mathbf{r} = (\mathbf{r}_1,\ldots,\mathbf{r}_N)^\top \in \mathbb{R}^{3N}$ spatial electron coordinates
  - E electronic energy, ψ electronic wave function.

- **Hamiltonian** operator:

$$\hat{H} := \sum_i \Big(-\tfrac{1}{2}\nabla^2_{\mathbf{r}_i} - \sum_I \frac{Z_I}{|\mathbf{r}_i - \mathbf{R}_I|} \Big) + \sum_{i<j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$$

  - Z_I nuclear charges
  - $\mathbf{R}_I$ fixed nuclear coordinates (Born--Oppenheimer approximation).

Electronic Schrödinger Equation - Antisymmetric wavefunctions

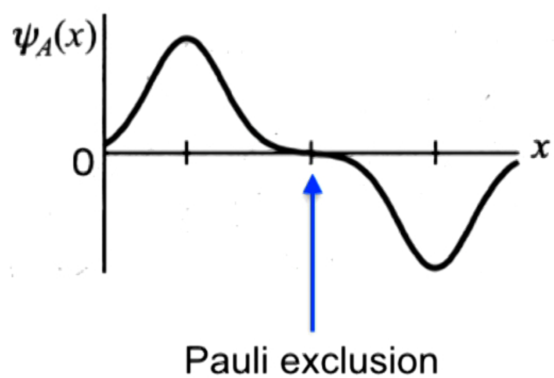
- **Electronic Schrödinger Equation**:

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

  - $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)^\top \in \mathbb{R}^{3N}$ spatial electron coordinates
  - E electronic energy, ψ electronic wave function.

- **Electronic spins** $s_i \in \{\uparrow, \downarrow\}$.
  ψ must be antisymmetric wrt exchange of equal-spin electrons

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



Variational Approach — Keystone for machine learning

- **Variational Approach**

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

Machine Learning loss function

Represent with neural network

Questions:

- Computation: How do we evaluate $E[\psi] = \int d\mathbf{r}\, \psi(\mathbf{r}) \hat{H} \psi(\mathbf{r})$?
- Representation: How do we build physics into $\psi(\mathbf{r})$?
  (antisymmetry, asymptotics, …)

Quantum Monte Carlo is the natural choice!

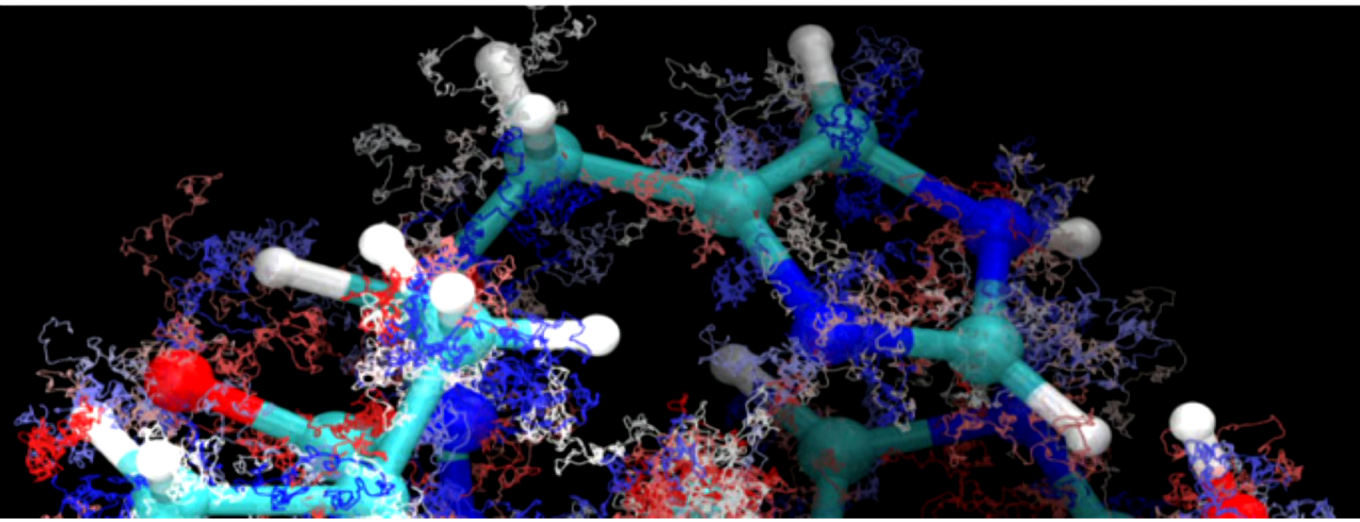
- **Variational Approach**

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

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

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

Sample Minibatches



Why is Quantum Monte Carlo an accurate method?

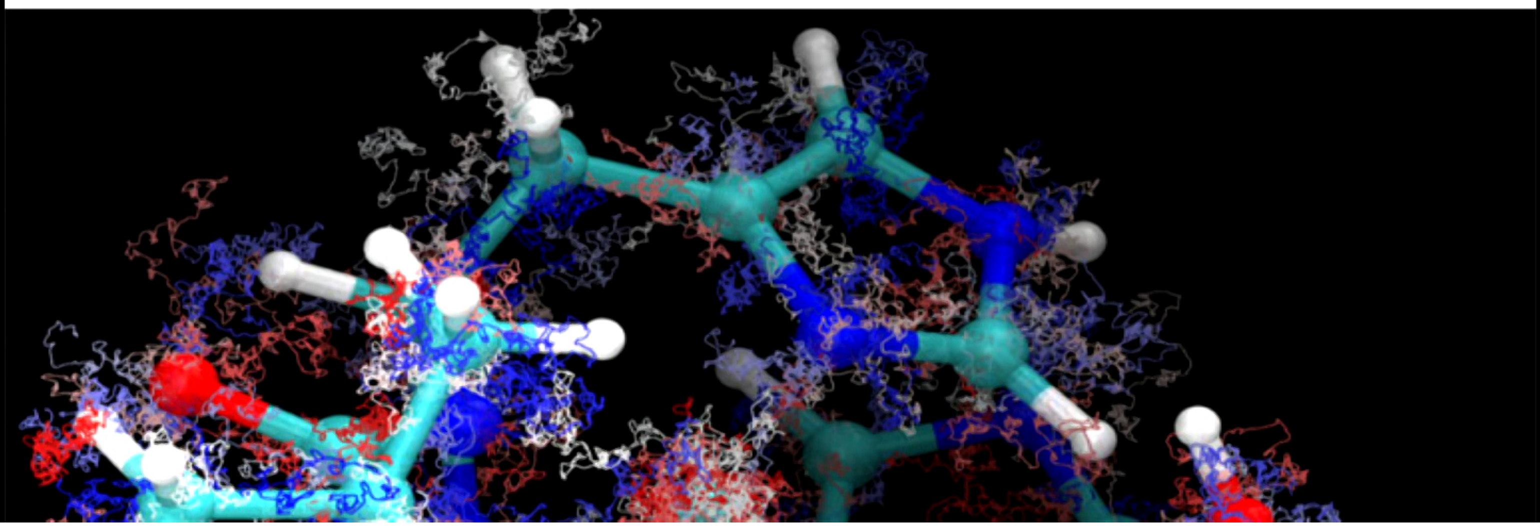
Monte Carlo converges slowly:

$$\mathrm{Var}\{E[\psi;\theta]\} = \frac{\mathrm{Var}_{\mathbf{r}}\{E_{\mathrm{loc}}[\psi](\mathbf{r};\theta)\}}{N_{\mathrm{sample}}}$$

But: **Variance** of local energy **decreases** during training

At the eigenstate: $E_{\mathrm{loc}}[\psi](\mathbf{r};\theta) = \hat{H}\psi(\mathbf{r};\theta)/\psi(\mathbf{r};\theta)$ = constant in **r** !

$\hat{H}\psi = E\psi$



Quantum Monte Carlo + Neural Networks

- **Variational Approach**

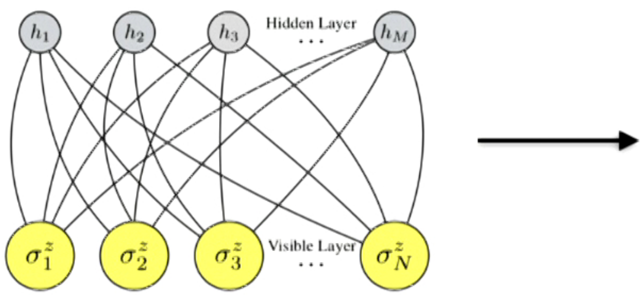
$$E_0 = \min_{\psi} E[\psi] \le \min_{\theta} E[\psi(\mathbf{r};\theta)] \qquad E[\psi] = \int d\mathbf{r}\, \psi(\mathbf{r}) \hat{H} \psi(\mathbf{r})$$

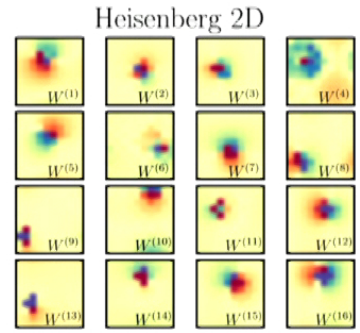
- **VMC**: Variational Quantum Monte Carlo:

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

QMC with Restricted Boltzmann Machines:

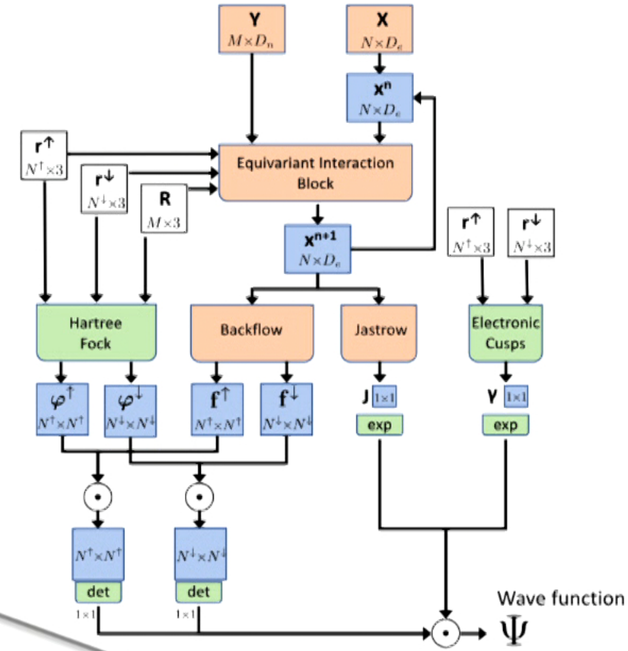
Carleo, Troyer: Solving the Quantum Many-Body Problem with Artificial Neural Networks, Science 355: 602-606 (2011)



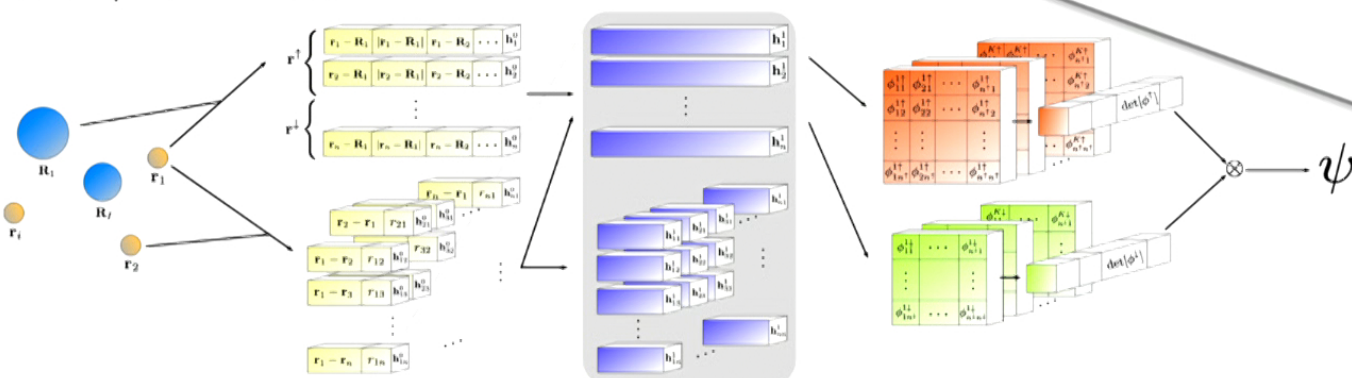


Quantum Chemistry Approaches (both Sept 2019)

PauliNet: (this work)
Hermann, Schätzle, Noé: **arXiv**:1909.08423
Deep neural network solution of the electronic Schrödinger equation



FermiNet
Pfau, Spencer, Matthews, Foulkes: **arXiv**:1909.02487
Ab-Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks



PauliNet Wavefunction

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

Incorporating Physics

1) Enforce Antisymmetry
⟶ Backflow

2) Baseline Performance
⟶ Hartree Fock Input

3) Correlation Energy
⟶ Slater-Jastrow Backflow

4) Asymptotics
⟶ Cusp conditions

PauliNet Wavefunction

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

spin-up electrons spin-down electrons

Assignment to spin-groups avoids spin coordinates

PauliNet Wavefunction

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

antisymmetric part

1) Enforcing antisymmetry: Standard QM approach

- **Antisymmetry** for **single-electron** functions: Slater determinants.

$$\psi_{\mathrm{HF}}(\mathbf{r}) := \det[\varphi_\mu(\mathbf{r}_i^\uparrow)] \det[\varphi_\mu(\mathbf{r}_i^\downarrow)]$$

$$\det[\varphi_\mu(\mathbf{r}_i^\uparrow)] = \begin{vmatrix} \varphi_1(\mathbf{r}_1^\uparrow) & \varphi_2(\mathbf{r}_1^\uparrow) & \cdots & \varphi_N(\mathbf{r}_1^\uparrow) \\ \varphi_1(\mathbf{r}_2^\uparrow) & \varphi_2(\mathbf{r}_2^\uparrow) & \cdots & \varphi_N(\mathbf{r}_2^\uparrow) \\ \vdots & \vdots & & \vdots \\ \varphi_1(\mathbf{r}_N^\uparrow) & \varphi_2(\mathbf{r}_N^\uparrow) & \cdots & \varphi_N(\mathbf{r}_N^\uparrow) \end{vmatrix}$$

→ Limited representative power

→ Many Slater determinants needed for high accuracy (CI)

PauliNet Wavefunction

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

antisymmetric part

1) Enforcing antisymmetry in PauliNet: Backflow

- **Antisymmetry** for **many-electron** functions $\varphi_\mu(\mathbf{r}_i) \to \varphi_{\mu i}(\mathbf{r})$ holds if $\varphi_{\mu i}(\mathbf{r})$ is equivariant wrt exchange $\mathscr{P}$ of same-spin electrons[1],[2]

$$\varphi_\mu(\mathbf{r}_i) \to \varphi_{\mu i}(\mathbf{r}),$$
$$\mathscr{P}\varphi_{\mu i}(\mathbf{r}) = \varphi_{\mu i}(\mathscr{P}\mathbf{r})$$

- **PauliNet**: $\varphi_{\mu i}(\mathbf{r}) = \varphi_\mu(\mathbf{r}_i^\uparrow) f_{\mu i}(\mathbf{r};\theta)$, $f_{\mu i}$ equivariant – SchNet[3].

→ Greater representative power

→ Hope: few or one Slater determinant give high accuracy

[1] P. López Ríos et al, *Phys. Rev. E*, 74:066701(2006)
[2] D. Luo, B. K. Clark. *Phys. Rev. Lett.* 122:226401 (2019)
[3] K. T. Schütt et al., *J. Chem. Phys.* 148:241722 (2018)

PauliNet Wavefunction

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

2) Baseline Performance: Hartree-Fock

- Use fixed Hartree-Fock (HF) orbitals $\varphi_\mu(\mathbf{r}_i^\uparrow)$
- Pro: HF fast, captures physics of atoms and molecules qualitatively
- Con: poor quantitative predictions, no electron-electron correlations.

PauliNet Wavefunction

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

3) **Correlation Energy** (beyond Hartree-Fock): **deep learning**

- **Jastrow factor**[4] $J(\mathbf{r};\theta)$: Antisymmetric part is multiplied by nonnegative symmetric function $e^{J(\mathbf{r};\theta)}$. Can capture complex electron correlations but not change the nodal surface.
- **Backflow**[5,6] $\mathbf{f}(\mathbf{r};\theta)$: Generalize one-electron orbitals $\varphi_\mu(\mathbf{r}_i)$ to many-electron orbitals $\varphi_{\mu i}(\mathbf{r})$. Can change the nodal surface.

---

[4] N. D. Drummond et al. *Phys. Rev. B* 70:235119 (2004)
[5] P. López Ríos et al, *Phys. Rev. E*, 74:066701(2006)
[6] D. Luo, B. K. Clark. *Phys. Rev. Lett.* 122:226401 (2019)

PauliNet Wavefunction

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

4) Asymptotics

- **Cusp conditions**:

$$\frac{1}{\psi_0}\frac{\partial \psi_0}{\partial|\mathbf{r}_i - \mathbf{R}_I|}\bigg|_{\mathbf{r}_i=\mathbf{R}_I} = -Z_I \qquad \frac{1}{\psi_0}\frac{\partial \psi_0}{\partial|\mathbf{r}_i - \mathbf{r}_j|}\bigg|_{\mathbf{r}_i=\mathbf{r}_j} = \begin{cases} \frac{1}{4} & s_i = s_j \\ \frac{1}{2} & s_i \neq s_j \end{cases}$$

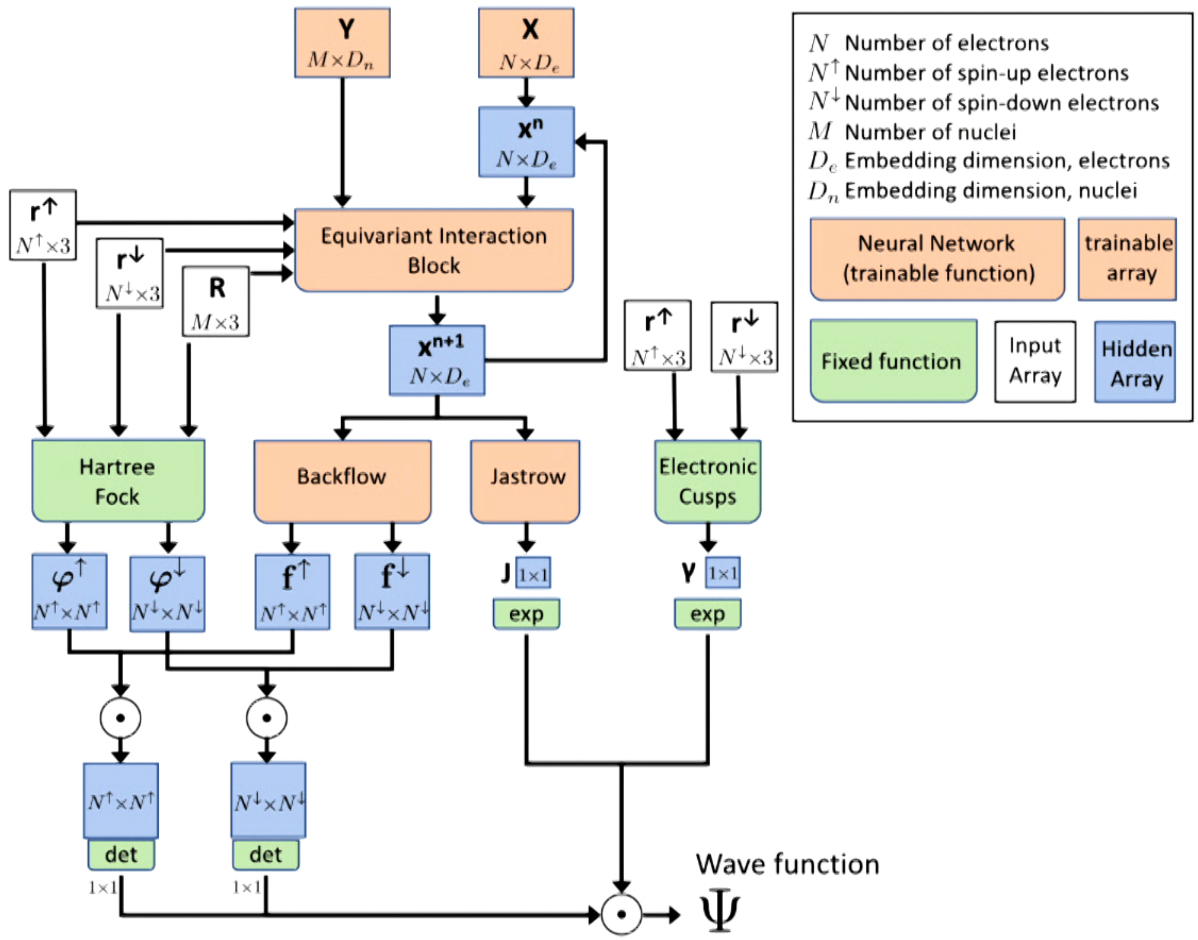
- **PauliNet**:
  - Nuclear cusps are built into $\varphi_\mu(\mathbf{r}_i)$[7].
  - Electronic cusps: use asymptotic function

$$\gamma(\mathbf{r}) := \sum_{i<j} -\frac{c_{ij}}{1+|\mathbf{r}_i - \mathbf{r}_j|} \qquad c_{ij} = \begin{cases} \frac{1}{4} & s_i = s_j \\ \frac{1}{2} & s_i \neq s_j \end{cases}$$

  - Backflow and Jastrow networks are cuspless by construction:

[7] A. Ma et al: J. Chem. Phys. 122:224322 (2005)

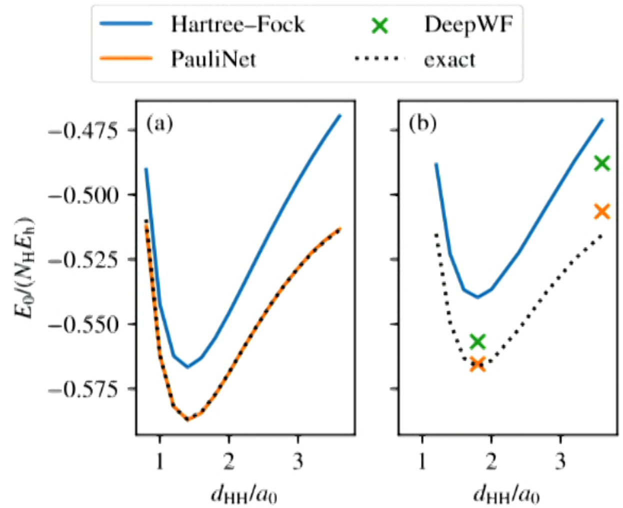
PauliNet



Y
$M \times D_n$
X
$N \times D_e$
x^n
$N \times D_e$
r↑
$N^\uparrow \times 3$
r↓
$N^\downarrow \times 3$
R
$M \times 3$
Equivariant Interaction Block
x^{n+1}
$N \times D_e$
r↑
$N^\uparrow \times 3$
r↓
$N^\downarrow \times 3$
Hartree Fock
Backflow
Jastrow
Electronic Cusps
$\varphi^\uparrow$
$N^\uparrow \times N^\uparrow$
$\varphi^\downarrow$
$N^\downarrow \times N^\downarrow$
$f^\uparrow$
$N^\uparrow \times N^\uparrow$
$f^\downarrow$
$N^\downarrow \times N^\downarrow$
J 1×1
exp
γ 1×1
exp
$N^\uparrow \times N^\uparrow$
det
1×1
$N^\downarrow \times N^\downarrow$
det
1×1
Wave function
Ψ
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: Results

system[a]	E_0/E_h (% E_{corr}) VMC		DMC		DeepWF		PauliNet	
H_2^b					−1.1738	98.4%	−1.174 37(6)	99.7%
LiH[c]	−8.0635[f]	91.5%	−8.0703[f]	99.7%	−7.8732	[g]	−8.0690(3)	98.1%
Be[d]	−14.6311[d]	61.6%	−14.6572[d]	89.2%	−14.6141	43.6%	−14.6546(7)	86.5%
B[d]	−24.6056[d]	60.0%	−24.6398[d]	88.3%	−24.2124	[g]	−24.634(2)	83.5%
H_{10}^e					−5.5685	63.8%	−5.655(2)	96.0%



Acknowledgements

Collaborations

Cecilia Clementi (Rice University)
Christof Schütte (FU Berlin)
Eric Vanden-Eijnden (Courant Institut NY)
Thomas Weikl (MPI Potsdam)
Edina Rosta (King's College London)

Vijay Pande (Stanford)
Volker Haucke (FMP Berlin)
Stephan Sigrist (FU Berlin)
Katja, Faelber, Oliver Daumke (MDC)
John Chodera (MSKCC NY)
Gianni de Fabritiis (Barcelona)

Funding

erc
EINSTEIN Foundation.de
DFG Deutsche Forschungsgemeinschaft
Freie Universität Berlin