science.sciencemag.org/content/365/6457/eaaw1147/suppl/DC1

Supplementary Materials for

Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning

Frank Noé*†, Simon Olsson*, Jonas Köhler*, Hao Wu

*These authors contributed equally to this work.
†Corresponding author. Email: frank.noe@fu-berlin.de

Published 6 September 2019, *Science* **365**, eaaw1147 (2019)
DOI: 10.1126/science.aaw1147

This PDF file includes:

Supplementary Text

M layer: ensuring invertibility in dihedral space For the mixed coordinate transformation layer M, it must be avoided that the Boltzmann Generator samples internal coordinates that are outside the range that are generated by the Cartesian coordinate transformation $I(\mathbf{x})$ (by default $[-\pi, \pi]$ before normalization). While angle values outside these bounds pose no problem for the placement of atom positions as they are automatically periodically wrapped during this process, they would break invertibility $\mathbf{z} \to \mathbf{x} \to \mathbf{z}$, and thus invalidate the random variable transformation principle of the Boltzmann Generator. Here we avoid this problem by adding a simple quadratic loss during training that penalizes angles generated outside the $[-\pi, \pi]$ range with a weight w_{torsion} and is inactive within the range. This excludes violations of invertibility for most, but not all samples. To ensure we are working on the manifold which is invertible up to a global roto-translation we then simply discard those samples for which invertibility $\mathbf{z} \to \mathbf{x} \to \mathbf{z}$ is violated.

Derivation of the KL loss as free energy For invertible transformation F_{xz}, we use the following relationship of the entropies of the two distributions:

$$\begin{aligned}
H_X &= -\int_{\mathbf{x}} q_X(\mathbf{x}) \log q_X(\mathbf{x})\, \mathrm{d}\mathbf{x} \\
&= -\int_{\mathbf{z}} q_X(F_{zx}(\mathbf{z})) \log \left(q_X(F_{zx}(\mathbf{z}))\, R_{zx}(\mathbf{z}) \right)\, \mathrm{d}\mathbf{z} \\
&= -\int_{\mathbf{z}} \mu_Z(\mathbf{z}) \log q_X(F_{zx}(\mathbf{z}))\, \mathrm{d}\mathbf{z} \\
&= -\int_{\mathbf{z}} \mu_Z(\mathbf{z}) \log \left(\mu_Z(\mathbf{z}) R_{zx}(\mathbf{z})^{-1} \right)\, \mathrm{d}\mathbf{z} \\
&= -\left(\int_{\mathbf{z}} \mu_Z(\mathbf{z}) \log \mu_Z(\mathbf{z})\, \mathrm{d}\mathbf{z} \right) + \mathbb{E}_{\mathbf{z}\sim\mu_Z(\mathbf{z})} \log R_{zx}(\mathbf{z}) \\
&= \boxed{H_Z + \mathbb{E}_{\mathbf{z}\sim\mu_Z(\mathbf{z})} \left[\log R_{zx}(\mathbf{z}) \right]}
\end{aligned} \tag{17}$$

Hence we have:

$$\begin{aligned}
\mathrm{KL}_{\boldsymbol{\theta}} \left[\mu_Z \parallel q_Z \right] &= -H_Z + \log Z_X + \mathbb{E}_{\mathbf{z}\sim\mu_Z(\mathbf{z})} \left[u(F_{zx}(\mathbf{z}; \boldsymbol{\theta})) \right] - \mathbb{E}_{\mathbf{z}\sim\mu_Z(\mathbf{z})} \left[\log R_{zx}(\mathbf{z}; \boldsymbol{\theta}) \right] \\
&= -H_X + \log Z_X + \mathbb{E}_{\mathbf{z}\sim\mu_Z(\mathbf{z})} \left[u(F_{zx}(\mathbf{z}; \boldsymbol{\theta})) \right] \\
&= -H_X + \log Z_X + \mathbb{E}_{\mathbf{x}\sim\mu_X(\mathbf{x}; \boldsymbol{\theta})} \left[u(\mathbf{x}) \right] \\
&= \boxed{\mathrm{KL}_{\boldsymbol{\theta}} \left[q_X \parallel \mu_X \right].}
\end{aligned}$$

Then, the KL loss function becomes the free energy shown in Eq. (14).

Simulation systems and general hyper-parameter choices

The "MD" simulations of the model systems (double well, Mueller potential, solvated particle dimer) are not using actual molecular dynamics, but are emulated with Metropolis Monte Carlo with small local steps. In each step, a random vector from an isotropic Gaussian distribution with a system-dependent standard deviation σ_{Metro} is added to the present configuration. This proposed configuration is accepted or rejected with a standard Metropolis acceptance criterion.

All Boltzmann Generator networks are composed of invertible blocks of non-volume preserving RealNVP layers. Each block contains two such layers to make sure that all dimensions are subject to a nonlinear transformation (Fig. 1b). Each configuration $\mathbf{x}$ or latent vector $\mathbf{z}$ is split into a channel of "even" and "odd" dimensions, defining the pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{z}_1, \mathbf{z}_2)$, respectively. To describe the network architecture used, we use R to denote RealNVP block and W for a PCA-based whitening layer. A subscript is used to denote the number of repetitions of a motif, e.g. R_{10} are ten stacked RealNVP blocks.

We always used ReLU (rectified linear units) nonlinearities for the translation networks (T in Eq. 5,7) and tanh nonlinearities for the scaling networks (S in Eq. 5,7). For each Boltzmann Generator, T and S use equal network architectures with l_{hidden} hidden layers containing n_{hidden} neurons each.

All networks are trained using the Adam adaptive stochastic gradient descent method (*43*). Other choices and hyper-parameters are described below.

In the first iterations of training that involves minimizing the loss term J_{KL}, i.e. when the Boltzmann Generator first starts generating samples whose free energies are being minimized, there is a significant chance of generating extremely high energy values. We therefore regularize the energy as follows:

$$\boxed{E_{\text{reg}} = \begin{cases} E & E < E_{\text{high}} \\ E_{\text{high}} + \log\left(E - E_{\text{high}} + 1\right) & E_{\text{high}} \leq E < E_{\text{max}} \\ E_{\text{high}} + \log\left(E_{\text{max}} - E_{\text{high}} + 1\right) & E_{\text{max}} < E \end{cases}}$$

where $E_{\text{max}} = 10^{20}$ is a cutoff just set to avoid overflow and E_{high} is initially very large and then gradually reduced during training, but is left at a value far above the equilibrium energies. The aim is that after training almost all samples end up in the linear regime $E < E_{\text{high}}$ where the employed loss functions are meaningful.

Double well We define a two-dimensional toy model which is bistable in x-direction and harmonic in y-direction:

$$E(x,y)=\frac{1}{4}ax^{4}-\frac{1}{2}bx^{2}+cx+\frac{1}{2}dy^{2} \tag{18}$$

with $a=c=d=1$ and $b=6$ – see Fig. 2a. The system is simulated with a Metropolis step of $\sigma_{\text{Metro}}=0.1$. To estimate the average time needed for a return trip between both states, we construct another systems with $a=0.25$ and $b=1.5$ that has the same position of minima and the same energy difference between them, but a much smaller barrier. For the "flat" systems frequent transitions between the two end-states are observed. The return-trip time of the original system is then estimated by $t=t_{\text{flat}}\exp\left(B-B_{\text{flat}}\right)$, where B,B_{flat} are the energy barriers for the original and the "flat" system from either one of the two minima, and t,t_{flat} are the times taken for a round-trip between the states. This results in an estimate of $t=4\cdot10^{6}$ simulation steps for a return trip in the double well system shown in Fig. 2a.

Boltzmann Generators for the double well system use the following hyper-parameters and training schedules:

Results figure	Input samples	Network	l_{hidden}	n_{hidden}	temperatures
Fig. 2a-f, Suppl. Fig. S2	1000	R_4	3	100	1.0
Fig. 6c	100	R_4	3	100	0.5, 1.0, 2.0, 4.0

	Fig. 2a-f		Fig. 6c	
iter	200	500	200	100
batch	128	1000	128	1000
lr	0.01	0.001	0.01	0.001
w_{ML}	1	1	1	1
w_{KL}	0	1	0	1
w_{RC}	0	0/1*	0	0

$*$: $w_{RC}=0$ for "green" results and $w_{RC}=1$ for "orange" results in Fig. 2. For Fig. S2 different training schedules were compared, as described in the figure caption.

Mueller potential

A scaled version of the Mueller potential was defined as:

$$E(x,y)=\alpha\sum_{j=1}^{4}A_{j}\exp\left[a_{j}\left(x-\hat{x}_{j}\right)^{2}+b_{j}\left(x-\hat{x}_{j}\right)\left(y-\hat{y}_{j}\right)+c_{j}\left(y-\hat{y}_{j}\right)^{2}\right]$$

with scaling parameter $\alpha = 0.1$ and:

	1	2	3	4
a_j	-1	-1	-6.5	0.7
b_j	0	0	11	0.6
c_j	-10	-10	6.5	0.7
A_j	-200	-100	-170	15
$\hat{x}_j$	1	0	-0.5	-1
$\hat{y}_j$	0	0.5	1.5	1

Boltzmann Generator architecture and training schedules were chosen as follows:

Results figure	Input samples	Network	l_{hidden}	n_{hidden}	temperatures
Fig. 2g-m	100	R_5	3	100	1.0
Fig. 6d	100	R_5	3	100	0.25, 0.5, 1, 2, 3

	Fig. 2g-m		Fig. 6d	
iter	200	500	200	100
batch	128	1000	128	1000
lr	0.01	0.001	0.01	0.001
w_{ML}	1	1	1	1
w_{KL}	0	1	0	1
w_{RC}	0	0/1*	0	0

$*$: $w_{RC} = 0$ for "green" results and $w_{RC} = 1$ for "orange" results in Fig. 2.

Bistable particle dimer in a Lennard-Jones fluid Here we simulate two-dimensional system of a bistable particle dimer in a dense bath of $n_s = 36$ solvent particles with Lennard-Jones repulsion. A similar system has been proposed in (*25*). The configuration vector is defined by alternating $x-$ and $y-$ positions and starting with the two dimer particles:

$$\mathbf{x} = \left[\mathbf{x}_{1x}, \mathbf{x}_{1y}, \mathbf{x}_{2x}, \mathbf{x}_{2y}, ..., \mathbf{x}_{(n_s+2)x}, \mathbf{x}_{(n_s+2)y}\right].$$

Defining the dimer distance $d = \|\mathbf{x}_1 - \mathbf{x}_2\|$, and the Heaviside step function h, we use the potential energy:

$$\begin{aligned}
U(\mathbf{x}) &= k_d(\mathbf{x}_{1x} + \mathbf{x}_{2x})^2 + k_d\mathbf{x}_{1y}^2 + k_d\mathbf{x}_{2y}^2 \\
&+ \frac{1}{4}a(d - d_0)^4 - \frac{1}{2}b(d - d_0)^2 + c(d - d_0)^4 \\
&+ \sum_{i=1}^{n+2} h(-\mathbf{x}_{ix} - l_{\text{box}})k_{\text{box}}(-\mathbf{x}_{ix} - l_{\text{box}})^2 + \sum_{i=1}^{n+2} h(\mathbf{x}_{ix} - l_{\text{box}})k_{\text{box}}(\mathbf{x}_{ix} - l_{\text{box}})^2 \\
&+ \sum_{i=1}^{n+2} h(-\mathbf{x}_{iy} - l_{\text{box}})k_{\text{box}}(-\mathbf{x}_{iy} - l_{\text{box}})^2 + \sum_{i=1}^{n+2} h(\mathbf{x}_{iy} - l_{\text{box}})k_{\text{box}}(\mathbf{x}_{iy} - l_{\text{box}})^2 \\
&+ \epsilon \sum_{i=1}^{n+1} \sum_{j=i+1, j\neq 2}^{n+2} \left(\frac{\sigma}{\|\mathbf{x}_i - \mathbf{x}_j\|}\right)^{12}
\end{aligned}$$

where the five rows correspond to: (1) Constraints for the center and the y-position of the particle dimer, (2) particle dimer interaction, (3,4) box constraints in $x-$ and $y-$direction, (5) particle repulsion. The following parameter values were used (all in reduced units):

Parameter	ϵ	σ	k_d	d_0	a	b	c	l_{box}	k_{box}
Value	1.0	1.1	20.0	1.5	25.0	10.0	-0.5	3.0	100.0

To initialize training, we run Metropolis Monte Carlo simulations with a Metropolis step length of $\sigma_{\text{Metro}} = 0.02\sqrt{\tau}$, where τ is the relative temperature. To estimate the time taken for a return-trip between open and closed dimer states, we take the same approach as for the double-well system above: We conduct a simulation with 10^6 simulation steps for a system with maximally flattened energy ($a = 2.5$ and $b = 1.0$). Still no transition from closed to open states occur, we thus estimate the *lower bound* for the return trip to be $t = 10^6 \exp(B - B_{\text{flat}}) \approx 1.2 \cdot 10^{12}$ where B, B_{flat} are the intrinsic barrier heights for the unchanged and flattened system.

For validation of the free energy profiles predicted in Fig. 3e and Fig. 6e, we perform Umbrella Sampling simulations (*1*) for each relative temperature using 35 Umbrella potentials on the dimer distance between values of 0.5 and 2.5 and with a force constant of 500 (reduced units). Each umbrella simulation was $50,000$ steps, and to avoid hysteresis effects, we ran the umbrella sequence forward and backward, resulting in a total of $3 \cdot 70 \cdot 50,000 = 10.5$ million simulation steps for Fig. 3e. For Fig. 6e we ran 3 such simulations at each of 5 temperatures, resulting in 52.5 million simulation steps.

For initializing the training by example (ML), 10^5 simulation steps are stored for the "open" and "closed" dimer states, with no transitions between these states occurring in the simulations. For the free energy difference approach using two Boltzmann Generators (Fig. 6e) only $10,000$ simulation steps were used and

Gaussian noise with a standard deviation of 0.05 was added to the configurations. In order to avoid having to learn the permutational invariance of the diffusing solvent particles from the data, we remove this invariance by relabeling solvent particles using the Hungarian algorithm (*44*).

Boltzmann Generator training was done using the following hyper-parameters:

Results figure	Input samples	Network	l_{hidden}	n_{hidden}	temperatures
Fig. 3	100,000	R_8	3	200	0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4
Fig. 6e	10,000	WR_8	4	100	1, 2, 3

and following training schedules:

	Fig. 3						Fig. 6e							
iter	20	200	300	300	1000	2000	100	40	40	40	40	40	100	200
batch	256	8000	8000	8000	8000	8000	128	1000	1000	1000	1000	1000	1000	1000
lr	10^{-3}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-3}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
w_{ML}	1	100	100	100	20	0.01	1	1000	300	100	50	20	5	1
w_{KL}	0	1	1	1	1	1	-	1	1	1	1	1	1	1
w_{RC}	0	1	5	10	10	10	0	0	0	0	0	0	0	0
E_{high}	-	10^4	10^4	10^4	2000	1000	-	10^6	10^6	10^5	$5 \cdot 10^4$	$5 \cdot 10^4$	$5 \cdot 10^4$	$5 \cdot 10^4$

For Fig. 6e, we used a total of about 10^6 energy calls for training of both Boltzmann Generators and computing their free energy differences

Hyper-parameter optimization While the results shown in this paper appear to be robust over different network architectures, we demonstrate on the particle dimer as an example how hyper-parameter optimization can be conducted for Boltzmann Generators, and used the resulting hyper-parameters for the results in Fig. 3. The hyper-parameters were chosen by minimizing the estimator variance for the free energy profile along dimer distance d. Each trained network makes predictions for the free energy profile shown in Fig. 3e. Using bootstrapping the standard error over all free energies along the profile between $d = [0.5, 2.5]$ are computed, resulting in $(\epsilon_{0.5}, \epsilon_{1.0}, \epsilon_{2.0})$ for the three temperatures and $\bar{\epsilon} = \sqrt{\epsilon_{0.5}^2 + \epsilon_{1.0}^2 + \epsilon_{2.0}^2}$ as a total estimator error. Results of hyper-parameter optimization are shown in Table SS1.

Bovine Pancreatic Trypsin Inhibitor To treat more complicated molecular systems, a linking mechanism was implemented to exchange system coordinates, potential energies and forces in between the TensorFlow

(*27*) and OpenMM software libraries (*26*).

We set up an all-atom model of the bovine pancreatic trypsin inhibitor (BPTI) protein which has been characterized extensively by biophysical experiment and molecular simulations, using the published crystal structure topology (pdb: 5PTI (*45*)) and AMBER 99 SB ILDN (*46*) parameters to model intramolecular interactions of the protein, and solvation effects were treated implicitly using a generalized Born model (GBSA-OBC) with parameters adopted for use for AMBER99 and it variants (*47, 48*).

We generated data for ML based training using conformational states by sampling 6 initial configuration from a previously published 1 millisecond simulation of BPTI in explicit solvation (*28*) corresponding to high density areas in the slow collective coordinates (*49*). The six configurations were chosen using k-means clustering of the 15 leading components computed with TICA (time-lagged independent component analysis) (*29*) at 1 μs lag-time and using the following features: cosines and sines of all backbone torsions and χ_1 angles of cysteine residues forming a flexible disulfide bond (residue 14 and 35). The selected frames correspond to time-points: 46.396, 92.682, 70.339 87.889, 827.930 and 831.050 microseconds in the original trajectory. From each of these six frames, 20 nanoseconds of MD simulation was conducted at temperature 300 K using the forcefield as specified above using a Langevin integration approach with an integration time-step of 2 femtoseconds. Configurations were stored every 0.2 picoseconds.

For BPTI we used Boltzmann Generators with a mixed coordinate transformation layer M as first layer. The Cartesian set consisted of heavy backbone atoms (N, Cα, C') and the heavy side-chain atoms of disulfide bridges. The rest of the atoms (H_α, backbone O and side-chains not involved in disulfide bridges) defined the internal coordinate set. For all BPTI Boltzmann Generators, the MD data was subsampled to 100,000 configurations, starting from 6 MD datasets in Figs. 5 and 2 MD datasets in Fig. 6f. The hyper-parameters were chosen as follows:

Results figure	Network	l_{hidden}	n_{hidden}	temperatures
Fig. 5	MR_8	4	256, 128, 256	1
Fig. 6e	MR_8	4	200, 100, 200	0.9 0.95 1.0 1.05 1.1 1.15 1.2

For the results in Fig. 5, we made use of a two dimensional reaction coordinate loss defined by the two first time-lagged independent components estimated using a previously published molecular dynamics trajectory. We used the sines and cosines of backbone torsion angles and side-chain χ_1 angles of Cys 14 and Cys 38 as basis functions and estimated the projection with a with a lag-time of 100 nanoseconds. The loss is defined as the negative entropy of a batch distribution projected onto these coordinates, estimated using a soft binning with a 11×11 grid spanning the values $[-1, 6]$ and $[-2.1, 1.75]$ respectively.

Training was initiated by three times 2000 iterations of maximum likelihood with batch sizes 128, 256 and 512, respectively. Subsequently, following stages of mixed maximum likelihood and energy-based training was conducted for the Boltzmann Generator in Fig. 5, where $w_{RC} = 20.0$, $w_{\text{torsion}} = 1.0$, $w_{\text{ML}} = 1.0$ were used throughout:

iter	15	15	15	15	15	15	20	20	30	50	50	300
batch	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
lr	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
E_{high}	10^{10}	10^{9}	10^{8}	10^{7}	10^{6}	10^{5}	10^{5}	10^{5}	10^{4}	10^{4}	10^{3}	10^{3}
w_{KL}	10^{-12}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-3}	$5 \cdot 10^{-3}$	10^{-3}	$5 \cdot 10^{-3}$	$5 \cdot 10^{-2}$	$5 \cdot 10^{-2}$	$5 \cdot 10^{-2}$

For the computation of free energy differences using two Boltzmann Generators we used 20 ns MD simulation from states “X” and “O” as shown in Fig. 6f, and the following simplified training protocol:

iter	2000	2000	2000	30	30	30	30	30	30	400
batch	128	256	512	5000	5000	5000	5000	5000	5000	5000
E_{high}	-	-	-	10^{10}	10^{9}	10^{8}	10^{7}	10^{6}	10^{5}	10^{4}
lr	10^{-3}	10^{-3}	10^{-3}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
w_{KL}	0	0	0	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-2}
w_{tor}	-	-	-	0.01	0.1	0.1	0.1	1	1	1

Supplementary Figures

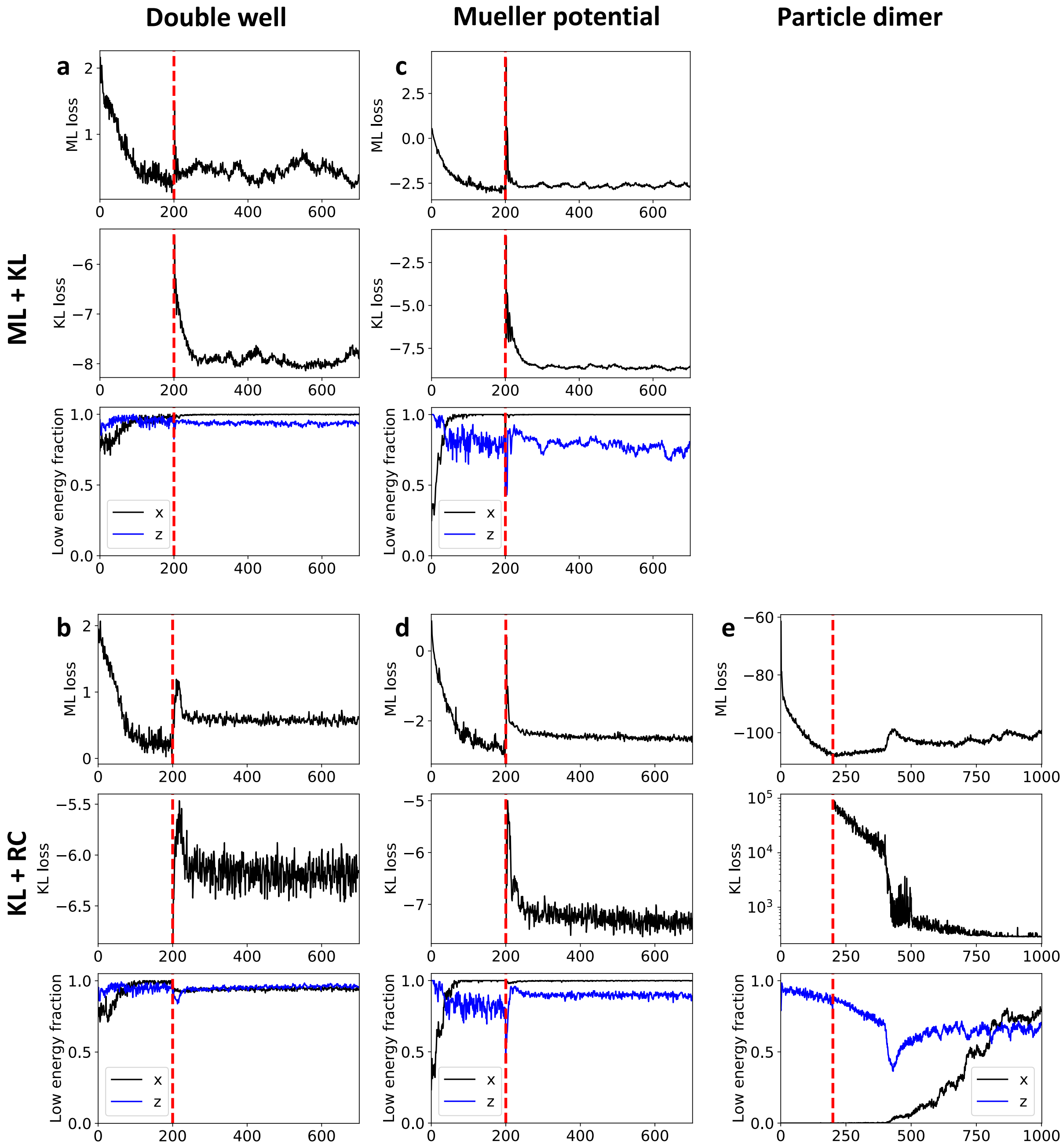

Figure S1: **Convergence metrics for Boltzmann Generators reported in Fig. 2 and Fig. 3**. Boltzmann Generators were trained with by energy and example (top row), or by energy and using a reaction coordinate loss (bottom row). Panels show: ML loss J_{ML} (top), KL loss J_{KL} (middle) and the low-energy fractions in configuration space $\mathbf{x}$ and in latent space $\mathbf{z}$ (bottom). These are defined by the fraction of Boltzmann Generator samples ($\mathbf{z} \rightarrow \mathbf{x}$) whose configuration energies are within the 99% percentile of the energies of the input data, and the fraction of input data that when mapped to latent space ($\mathbf{x} \rightarrow \mathbf{z}$) are within the 99% percentile of the energy distribution of a harmonic oscillator with the respective dimension. **a**-**b**): Double well potential. **c**-**d**): Mueller potential. **e**) Solvated particle dimer.

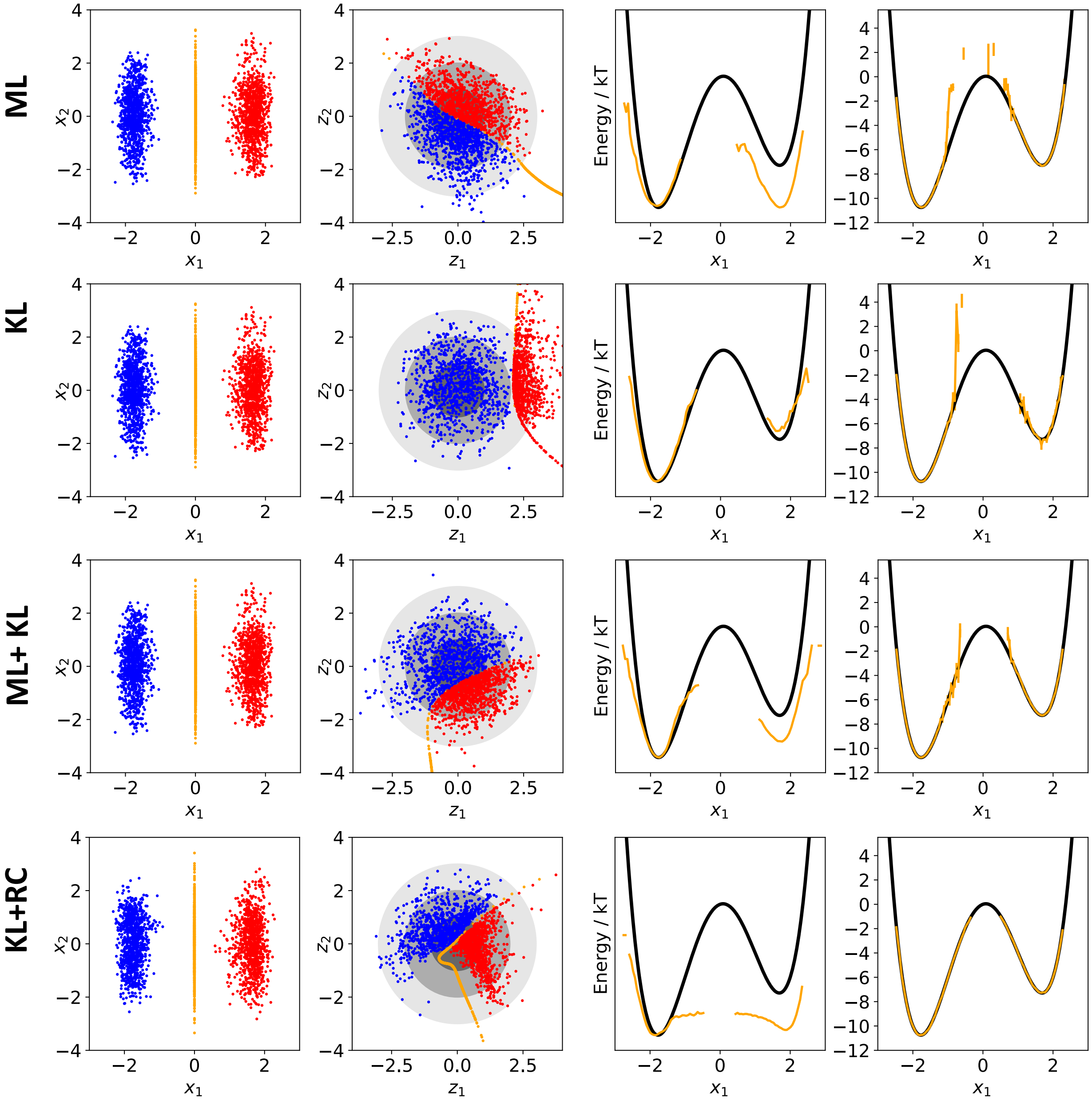

Figure S2: **Different training methods for Boltzmann Generators using the double well example (Fig. 2)**. Columns show: (1) distribution in configuration space $\mathbf{x}$, (2) distribution in latent space $\mathbf{z}$, (3) free energy of Boltzmann Generator output $p_X(\mathbf{x})$ along x_1, (4) free energy after reweighting, vertical bars show uncertainties (one standard deviation, 68% percentile). Training proceeds by 200 iterations of ML and then 500 iterations of the method given in the rows, using equal weights for these modes. Training by example (ML) only reproduces the distribution of the training data, which can be reweighted to the Boltzmann distribution in this low-dimensional example but reweighting from the ML-generated distribution fails for high-dimensional examples. Training by energy (KL) alone tends to collapse to a single metastable state. ML+KL combined samples closer to the Boltzmann distribution than ML and avoids metastable state collapse, but samples high-energy transition states with low probability. KL+RC performs best in this example.

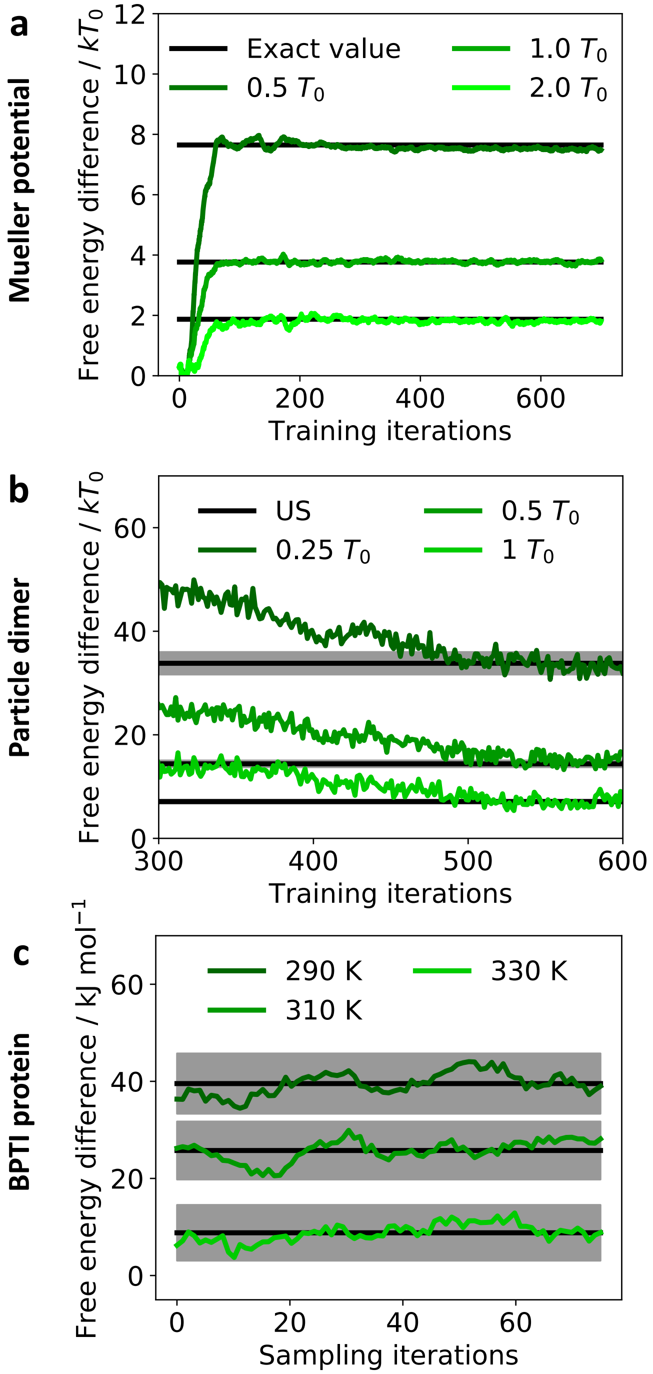

Figure S3: **Convergence of free energy differences shown in Fig. 6**. **a**) Mueller potential. **b**) Solvated Particle dimer. Convergence is shown for the last two training cycles, as the free energies are far from converged in earlier stages. References are from umbrella sampling, with intervals showing one standard error from 3 repeats. **c**) BPTI protein. For better performance, the free energy differences were not estimated during the training schedule reported in the Supp. Mat. Instead 75 training iterations were appended afterwards, and in each iteration 1000 samples of J_{KL} were obtained at each temperature for free energy estimation. Black lines / grey intervals show the mean and one standard error estimates from this phase using 5 independent repeats.

Architecture	nl_{layers}	nl_{hidden}	w_{ML}	w_{RC}	$\epsilon_{0.5}$	ϵ_1	ϵ_2	$\sqrt{\sum \epsilon^2}$
R_8	4	200	0.1	10.0	1.62	2.07	2.04	3.33
R_4	·	·	·	·	2.23	1.83	1.53	3.27
R_6	·	·	·	·	1.69	1.64	2.29	3.28
R_{12}	·	·	·	·	1.49	1.85	2.0	3.10
R_8	3	200	0.1	10.0	1.51	1.97	1.64	2.97
R_4	·	·	·	·	1.41	1.59	1.78	2.77
R_6	·	·	·	·	1.49	1.73	1.76	2.88
R_{12}	·	·	·	·	1.84	1.28	2.24	3.17
R_8	2	·	·	·	1.85	1.58	2.50	3.48
·	4	·	·	·	1.69	1.51	1.52	2.73
·	3	50	·	·	1.32	1.71	2.11	3.02
·	·	100	·	·	2.85	2.05	2.16	4.12
·	·	**200**	**0.01**	·	**1.58**	**1.33**	**1.33**	**2.45**
·	·	·	1.0	·	1.87	1.93	1.63	3.15
·	·	·	0.1	1.0	1.66	1.83	1.75	3.02
·	·	·	·	5.0	1.73	1.72	1.81	3.03
·	·	·	·	20.0	1.88	2.06	1.84	3.34

Table S1: **Hyper-parameter selection for the particle dimer**. In the architecture, R corresponds to a RealNVP block, i.e. two layers with channel swaps (Fig. 1b). The subscript indicates the number of repetitions, e.g. $\mathrm{R}_4 = \mathrm{RRRR}$, corresponding to eight single layers. All nonlinear transformations (T, S) the given number of layers (nl_{layers}) and hidden nodes (nl_{hidden}). All networks were trained on the following range of relative temperatures: $\tau \in [0.1, 0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4]$ and used $w_{KL} = 1.0$.

References and Notes

1. G. M. Torrie, J. P. Valleau, Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.* **23**, 187–199 (1977). doi:10.1016/0021-9991(77)90121-8

2. H. Grubmüller, Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Phys. Rev. E* **52**, 2893–2906 (1995). doi:10.1103/PhysRevE.52.2893 Medline

3. A. Laio, M. Parrinello, Escaping free-energy minima. *Proc. Natl. Acad. Sci. U.S.A.* **99**, 12562–12566 (2002). doi:10.1073/pnas.202427399 Medline

4. J. Hénin, G. Fiorin, C. Chipot, M. L. Klein, Exploring Multidimensional Free Energy Landscapes Using Time-Dependent Biases on Collective Variables. *J. Chem. Theory Comput.* **6**, 35–47 (2010). doi:10.1021/ct9004432 Medline

5. C. C. Wang, J. Bomback, W. T. Donlon, C. R. Huo, J. James, Optical third-harmonic generation in reflection from crystalline and amorphous samples of silicon. *Phys. Rev. Lett.* **57**, 1647–1650 (1986). doi:10.1103/PhysRevLett.57.1647 Medline

6. K. Hukushima, K. Nemoto, Exchange Monte Carlo Method and Application to Spin Glass Simulations. *J. Phys. Soc. Jpn.* **65**, 1604–1608 (1996). doi:10.1143/JPSJ.65.1604

7. E. Marinari, G. Parisi, Simulated Tempering: A New Monte Carlo Scheme. *Europhys. Lett.* **19**, 451–458 (1992). doi:10.1209/0295-5075/19/6/002

8. J. G. Kirkwood, Statistical Mechanics of Fluid Mixtures. *J. Chem. Phys.* **3**, 300–313 (1935). doi:10.1063/1.1749657

9. D. Frenkel, B. Smit, *Understanding Molecular Simulation* (Academic, 2001).

10. P. V. Klimovich, M. R. Shirts, D. L. Mobley, Guidelines for the analysis of free energy calculations. *J. Comput. Aided Mol. Des.* **29**, 397–411 (2015). doi:10.1007/s10822-015-9840-9 Medline

11. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015). doi:10.1038/nature14539 Medline

12. Z. Zhu, M. E. Tuckerman, S. O. Samuelson, G. J. Martyna, Using novel variable transformations to enhance conformational sampling in molecular dynamics. *Phys. Rev. Lett.* **88**, 100201 (2002). doi:10.1103/PhysRevLett.88.100201 Medline

13. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial networks," in *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems*, arXiv:1406.2661 (2014).

14. D. P. Kingma, M. Welling, "Auto-encoding variational Bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, arXiv:1312.6114 (2014).

15. T. Karras, T. Aila, S. Laine, J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proceedings of the 7nd International Conference on Learning Representations (ICLR)*, arXiv:1710.10196 (2018).

16. A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, D. Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning (ICML)* arXiv:1711.10433 (2018).

17. R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, A. Aspuru-Guzik, Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **4**, 268–276 (2018). doi:10.1021/acscentsci.7b00572 Medline

18. E. G. Tabak, E. Vanden-Eijnden, Density estimation by dual ascent of the log-likelihood. *Commun. Math. Sci.* **8**, 217–233 (2010). doi:10.4310/CMS.2010.v8.n1.a11

19. L. Dinh, D. Krueger, Y. Bengio, NICE: Non-linear independent components estimation, arXiv:1410.8516 (2015).

20. L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using Real NVP, arXiv:1605.08803 (2016).

21. D. J. Rezende, S. Mohamed, Variational inference with normalizing flows, arXiv:1505.05770 (2015).

22. D. P. Kingma, P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *NIPS'18 Proceedings of the 31st International Conference on Neural Information Processing Systems*, arXiv:1807.03039 (2018).

23. W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, D. Duvenaud, FFJORD: Free-form continuous dynamics for scalable reversible generative models, arXiv:1810.01367 (2018).

24. P. G. Bolhuis, D. Chandler, C. Dellago, P. L. Geissler, TRANSITION PATH SAMPLING: Throwing ropes over rough mountain passes, in the dark. *Annu. Rev. Phys. Chem.* **53**, 291–318 (2002). doi:10.1146/annurev.physchem.53.082301.113146 Medline

25. J. P. Nilmeier, G. E. Crooks, D. D. L. Minh, J. D. Chodera, Nonequilibrium candidate Monte Carlo is an efficient tool for equilibrium simulation. *Proc. Natl. Acad. Sci. U.S.A.* **108**, E1009–E1018 (2011). doi:10.1073/pnas.1106094108 Medline

26. P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, V. S. Pande, OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J. Chem. Theory Comput.* **9**, 461–469 (2013). doi:10.1021/ct300857j Medline

27. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, Tensorflow: Large-scale machine learning on heterogeneous systems, arXiv:1603.04467 (2016).

28. D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, W. Wriggers, Atomic-level characterization of the structural dynamics of proteins. *Science* **330**, 341–346 (2010). doi:10.1126/science.1187409 Medline

29. G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, F. Noé, Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **139**, 015102 (2013). doi:10.1063/1.4811489 Medline

30. M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, F. Noé, PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.* **11**, 5525–5542 (2015). doi:10.1021/acs.jctc.5b00743 Medline

31. M. J. Grey, C. Wang, A. G. Palmer 3rd, Disulfide bond isomerization in basic pancreatic trypsin inhibitor: Multisite chemical exchange quantified by CPMG relaxation dispersion and chemical shift modeling. *J. Am. Chem. Soc.* **125**, 14324–14335 (2003). doi:10.1021/ja0367389 Medline

32. D. Frenkel, A. J. C. Ladd, New Monte Carlo method to compute the free energy of arbitrary solids. Application to the fcc and hcp phases of hard spheres. *J. Chem. Phys.* **81**, 3188–3193 (1984). doi:10.1063/1.448024

33. W. G. Hoover, F. H. Ree, Melting Transition and Communal Entropy for Hard Spheres. *J. Chem. Phys.* **49**, 3609–3617 (1968). doi:10.1063/1.1670641

34. F. M. Ytreberg, D. M. Zuckerman, *J. Chem. Phys.* **124**, 104105 (2006). doi:10.1063/1.2174008 Medline

35. M. Chen, T.-Q. Yu, M. E. Tuckerman, Locating landmarks on high-dimensional free energy surfaces. *Proc. Natl. Acad. Sci. U.S.A.* **112**, 3235–3240 (2015). doi:10.1073/pnas.1418241112 Medline

36. J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. de Fabritiis, Frank Noé, C. Clementi, Machine learning of coarse-grained molecular dynamics force fields. *ACS Central Sci.* **5**, 755–767 (2019). doi:10.1021/acscentsci.8b00913 Medline

37. J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007). doi:10.1103/PhysRevLett.98.146401 Medline

38. M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012). doi:10.1103/PhysRevLett.108.058301 Medline

39. K. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, K.-R. Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. *Adv. Neural Inf. Process. Syst.* **30**, 992–1002 (2017).

40. J. S. Smith, O. Isayev, A. E. Roitberg. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017). doi:10.1039/C6SC05720A

41. M. R. Shirts, J. D. Chodera, Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.* **129**, 124105 (2008). doi:10.1063/1.2978177 Medline

42. F. Noé, S. Olsson, J. Köhler, H. Wu, Boltzmann generators – Sampling equilibrium states of many-body systems with deep learning, Zenodo (2019); http://doi.org/10.5281/zenodo.3242635

43. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980 (2015).

44. H. W. Kuhn, The Hungarian method for the assignment problem. *Nav. Res. Logist. Quart.* **2**, 83–97 (1955). doi:10.1002/nav.3800020109

45. A. Wlodawer, J. Walter, R. Huber, L. Sjölin, Structure of bovine pancreatic trypsin inhibitor. Results of joint neutron and X-ray refinement of crystal form II. *J. Mol. Biol.* **180**, 301–329 (1984). doi:10.1016/S0022-2836(84)80006-6 Medline

46. K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, D. E. Shaw, Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins* **78**, 1950–1958 (2010). doi:10.1002/prot.22711 Medline

47. A. Onufriev, D. Bashford, D. A. Case, Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins* **55**, 383–394 (2004). doi:10.1002/prot.20033 Medline

48. J. A. Rackers, Z. Wang, C. Lu, M. L. Laury, L. Lagardère, M. J. Schnieders, J.-P. Piquemal, P. Ren, J. W. Ponder, Tinker 8: Software Tools for Molecular Design. *J. Chem. Theory Comput.* **14**, 5273–5289 (2018). doi:10.1021/acs.jctc.8b00529

49. F. Noé, C. Clementi, Collective variables for the study of long-time kinetics from molecular trajectories: Theory and methods. *Curr. Opin. Struct. Biol.* **43**, 141–147 (2017). doi:10.1016/j.sbi.2017.02.006 Medline