Temperature-steerable flows

Manuel Dibak*†, Leon Klein*†, Frank Noé†‡

†Institute of Mathematics, Freie Universität Berlin, ‡Department of Chemisty, Rice University, Houston TX

*equal contribution

Freie Universität Berlin

Motivation

- Sampling equilibrium states of many-body systems is one of the grand challenges of statistical physics. Equilibrium densities of such systems with energy $U(x)$ often follow the Boltzmann distribution

$$\mu_X(x) \propto \exp(-U(x)/\tau)$$

where $\tau = k_B T$, with the temperature T and the Boltzmann constant k_B.
- Normalizing flows can be used to sample such densities directly without having to run long, correlated simulation chains [1].
- Normalizing flows transform an easy to sample prior distribution $p_Z(z)$, e.g. a multivariate normal distribution, via a transformation $x = f(z)$ to the output distribution $p_X(x)$ [2,3]. If $f(z)$ is invertible, $p_X(x)$ can be computed by the change of variable formula

$$p_X(x) = p_Z(z) \left|\det J_f(z)\right|^{-1}, \tag{1}$$

where $\left|\det J_f(z)\right|^{-1}$ is the inverse of the Jacobian.
- Advanced sampling methods, such as parallel tempering, require samplers at different temperatures. This is only possible with multiple instances of flows so far.

Goal

Derive flow f_τ, parametrized with the temperature τ, that correctly transforms the parametrized prior distribution $p_Z^\tau(z)$ to the target Boltzmann distribution $\mu_\tau(x)$ at temperature τ.

Temperature scaling condition

A change to temperature τ' of the Boltzmann distribution corresponds to raising it by the power of $\kappa = \tau/\tau'$

$$p_X^{\tau'}(x) \propto [p_X^\tau(x)]^\kappa .$$

Using Eq. (1) we observe that the temperature scaling is exact, if for any two temperatures τ, τ'

$$p_Z^{\tau'}(z) \left|\det J_{f_{\tau'}}(z)\right|^{-1} \propto \left[p_Z^\tau(z) \left|\det J_{f_\tau}(z)\right|^{-1}\right]^\kappa . \tag{2}$$

The temperature scaling condition is fulfilled by

1. selecting a Gaussian prior $p_Z^\tau(z) = \mathcal{N}(z|\vec{0}, \tau)$
2. building a flow with $\left|\det J_{f_\tau}(z)\right|^\kappa \propto \left|\det J_{f_{\tau'}}(z)\right|$

One possible choice are (modified) Augmented Normalizing Flows [4].

Training

- negative log likelihood (nll), requires samples from the target

$$\mathcal{L}_{ML} = \text{nll} = KL\left(\mu_X(x)||p_X(x)\right) = \mathbb{E}_{x\sim\mu_X(x)}\left[-\log p_X(x)\right] \tag{3}$$

- energy based training, requires energy of the target

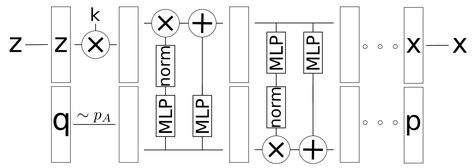
$$\mathcal{L}_{KL} = KL\left(p_X(x)||\mu_X(x)\right) = \mathbb{E}_{z\sim p_Z(z)}\left[U(D(z)) - \log\left|\det J_D(z)\right|\right]$$

Combine both for training $\mathcal{L} = (1-\lambda)\,\mathcal{L}_{ML} + \lambda\mathcal{L}_{KL}$

Sampling with Temperature-steerable flows

Sampling from the model at temperature τ

1. Sample $\vec{z} \sim p_Z^\tau(\vec{z}) = \mathcal{N}(0, \tau)$
2. Sample auxiliary momenta $\vec{q} \sim p_A^\tau(\vec{q}) = \mathcal{N}(0, \tau)$, and define the point in phase space $\vec{v} = (\vec{z}, \vec{q})$
3. Propagate $\vec{v}$ by the volume preserving dynamics shown on the right
4. Project onto the configuration variables $\vec{x}$

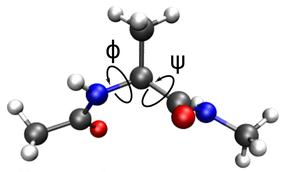


Monte Carlo sampling

1. Start with sample $\vec{x}$ and and sample corresponding momentum $\vec{p} \sim p_A^\tau(\vec{p})$ to get $\vec{\gamma} = (\vec{x}, \vec{p})$
2. Propagate $\vec{\gamma}$ by the inverse dynamics shown above to obtain the point in phase space $\vec{v}$
3. Add a random displacement $\vec{v}' = \vec{v} + \vec{\xi}$ with $\vec{\xi} \sim \mathcal{N}(0, \sigma^2)$
4. Transform back to configuration space with the dynamics
5. Accept/reject the new sample $\vec{x}'$ based on the Metropolis Hastings criterion

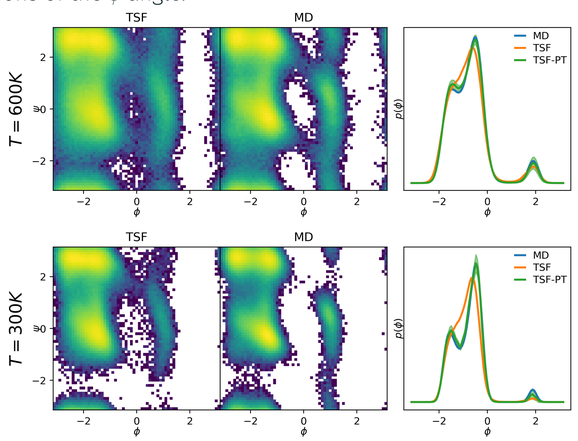
Experiments: Alanine dipeptide

System: Alanine dipeptide in implicit solvent model
Setting: Comparison between our proposed TSF and Molecular Dynamics (MD) simulation.
Training: The TSF is trained with samples at $T = 600\,\text{K}$

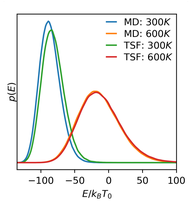


Evaluation: We use the TSF to generate samples at $T = 600\,\text{K}$ and $T = 300\,\text{K}$ and compare the Ramachandran plots and distributions of the ϕ angle.

We observe good agreement at the training temperature. At $T = 300\,\text{K}$ the TSF still finds the major minima at around $\phi \approx -2$, but undersamples the minimum at $\phi \approx 1$.
We are able to recover the correct distribution of ϕ when using the TSF as Monte Carlo sampler combined with parallel tempering (TSF-PT).

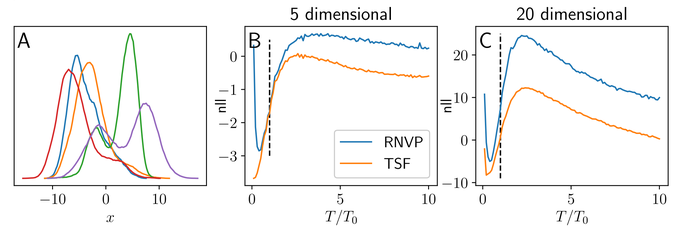


The samples generated by the TSF closely match the energy distribution compared to MD at the training temperature ($T = 600\,\text{K}$). Even at $T = 300\,\text{K}$ it is able to closely recover the distribution.



Experiments: Mixture of multi-dimensional double wells

System: Mixture of multi-dimensional double wells which are mixed via a correlation matrix A
Energy: $U(x) = U_{\text{dw}}(Ax)$ with $U_{\text{dw}}(x) = \sum_i^d a_i x_i + b x_i^2 + c x_i^4$
A) Marginal density of the first 5 coordinates of the 20 dimensional system.



Setting: Comparison between a RNVP flow [5] and our proposed TSF.
Training: Both models are trained with samples at $T = 1$.
Evaluation: The nll (Eq. 3) is evaluated at 100 temperatures in the range $T = 0.1$ to $T = 10$ for **B)** 5 and **C)** 20 dimensions. The TSF performs significantly better for temperatures further away from the training temperature $T = 1$.

References

[1] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators-sampling equilibrium states of many-body systems with deep learning. *Science* (2019)

[2] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv* (2015)

[3] George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv* (2019)

[4] C. Huang, Laurent Dinh, and Aaron C. Courville. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv* (2020)

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv* (2016)

Acknowledgments

We gratefully acknowledge support by the Deutsche Forschungsgemeinschaft (SFB1114, project C03), the European research council (ERC CoG 772230 “ScaleCell”), the Berlin Mathematics center MATH+ (AA1-6), and the German Ministry for Education and Research (BIFOLD - Berlin Institute for the Foundations of Learning and Data).
We thank Jonas Köhler, Andreas Krämer and Yaoyi Chen for insightful discussions.