Letter

Temperature steerable flows and Boltzmann generators

Manuel Dibak [1,*] Leon Klein [1,*] Andreas Krämer [1] and Frank Noé [1,2,3,4,†]

[1]Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 12, 14195 Berlin, Germany
[2]Department of Physics, Freie Universität Berlin, Arnimallee 12, 14195 Berlin, Germany
[3]Department of Chemistry, Rice University, Houston, Texas 77005, USA
[4]Microsoft Research, Station Road, Cambridge, United Kingdom

(Received 4 August 2021; accepted 10 August 2022; published 10 October 2022)

Boltzmann generators approach the sampling problem in many-body physics by combining a normalizing flow and a statistical reweighting method to generate samples in thermodynamic equilibrium. The equilibrium distribution is usually defined by an energy function and a thermodynamic state. Here, we propose temperature steerable flows (TSFs) which are able to generate a family of probability densities parametrized by a choosable temperature parameter. TSFs can be embedded in generalized ensemble sampling frameworks to sample a physical system across multiple thermodynamic states.

DOI: 10.1103/PhysRevResearch.4.L042005

I. INTRODUCTION

Sampling equilibrium states of many-body systems such as molecules, materials, or spin models is one of the grand challenges of statistical physics. Equilibrium densities of system states $\mathbf{x}$ are often given in the form

$$\mu_X(\mathbf{x}) \propto \exp[-u(\mathbf{x})], \tag{1}$$

where $u(\mathbf{x})$ is a reduced (unitless) energy that combines the system's potential $U(\mathbf{x})$ (if momenta are of interest we have the Hamiltonian energy instead) with thermodynamic variables that define the statistical ensemble. In the canonical ensemble the reduced energy is given by $u(\mathbf{x}) = U(\mathbf{x})/\tau$ where the thermal energy $\tau = k_B T$ is proportional to the temperature T and k_B is the Boltzmann constant. In order to model a system across a range of thermodynamic states, we would like to sample a family of densities parametrized by the thermodynamic control variables—in the canonical ensemble,

$$\mu_X^\tau(\mathbf{x}) \propto \exp\left(-\frac{U(\mathbf{x})}{\tau}\right). \tag{2}$$

The most common approaches to sample densities (1) in physics and chemistry are Markov chain Monte Carlo (MCMC) or molecular dynamics (MD)—both proceed in steps, making small changes to $\mathbf{x}$ at a time, and guarantee that the target density (1) will be sampled asymptotically. The convergence of such sampling algorithms is often slowed down by barriers in the energy landscape, which may result in very long, possibly unfeasible simulation times.

Additionally, many applications require running simulations at various thermodynamic states, e.g., to study the phase behavior and temperature dependence of materials and biological matter [1,2]. Moreover, generalized ensemble methods such as parallel tempering (PT) are frequently used to facilitate transitions across energy barriers and thereby enhance sampling. However, these techniques often require dozens of parallel simulations to enable Monte Carlo exchanges between different temperatures [3,4].

A novel alternative to traditional sampling methods are generative machine learning models. Recently, there has been a lot of interest in training normalizing flows [5–9] to sample densities of many-body physics systems such as in Eq. (1) directly without having to run long, correlated simulation chains. Normalizing flows (see Supplemental Material [10] for a brief introduction) transform an easy-to-sample prior distribution $p_Z(\mathbf{z})$, e.g., a multivariate normal distribution, via a transformation $\mathbf{x} = f(\mathbf{z})$ to the output distribution $p_X(\mathbf{x})$. If $f(\mathbf{z})$ is invertible, $p_X(\mathbf{x})$ can be computed by the change of variable formula

$$p_X(\mathbf{x}) = p_Z(\mathbf{z})|\det J_f(\mathbf{z})|^{-1}. \tag{3}$$

Boltzmann generators (BGs) [11] combine normalizing flows to minimize the distance between Eqs. (1) and (3) with a statistical reweighting or resampling method to generate unbiased samples from Eq. (1). This and similar approaches have been used to sample configurations of molecular and condensed matter systems [11,12], spin models [13–15], and gauge configurations in lattice quantum chromodynamics [16–19]. However, these previous generative approaches are only able to sample at a single predefined thermodynamic state.

This Letter shows that normalizing flows can be generalized to families of ensembles across multiple temperatures and thereby greatly increase the range of thermodynamic

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

states accessible to a sampling algorithm. Specifically, we develop temperature steerable flows (TSFs) that correctly parametrize the distribution p_X by a temperature variable τ such that it follows Eq. (2). We evaluate the method on the XY model [20] finding the correct temperature dependence of the magnetization. Moreover, we show for two small peptides, alanine dipeptide and tetrapeptide, that the TSF is capable of producing samples close to equilibrium at different temperatures. When trained on a single high temperature, the TSF can simultaneously sample at lower temperatures of interest, allowing a reliable estimation of the physical observables and conformational distributions. Finally, due to this property, TSFs are used to facilitate exchanges in classical parallel tempering MD and thereby reduce autocorrelation times significantly.

II. TEMPERATURE STEERABLE FLOWS

A. Temperature scaling

Up to a normalization constant, a change from temperature τ to τ' corresponds to raising the Boltzmann distribution to the power of $\kappa = \tau/\tau'$, $\mu_X^{\tau'}(\mathbf{x}) \propto [\mu_X^{\tau}(\mathbf{x})]^{\kappa}$. We now consider normalizing flows f_τ with priors p_Z^τ that depend on τ as a steerable parameter. Using Eq. (3) the output distribution of a flow scales temperatures equally, if for any two temperatures τ, τ',

$$p_Z^{\tau'}(\mathbf{z})|\det J_{f_{\tau'}}(\mathbf{z})|^{-1} \propto [p_Z^{\tau}(\mathbf{z})|\det J_{f_\tau}(\mathbf{z})|^{-1}]^{\kappa}. \tag{4}$$

In this Letter we thus consider flows to be temperature steerable, if they preserve this scaling condition. We construct flows that preserve this proportionality in two different manners: by either keeping the Jacobian constant and preserving the scaling condition in the prior or selecting a constant prior and respecting the scaling condition in the flow.

B. Temperature steerable flows by volume preservation

The proportionality in the prior distribution can be matched by Gaussians with variance τ, i.e., $p_Z^\tau(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \tau)$, which fulfills $p_Z^{\tau'}(\mathbf{z}) \propto [p_Z^\tau(\mathbf{z})]^\kappa$. This results in a condition on the Jacobian of the flow $|\det J_{f_\tau}(\mathbf{z})|^\kappa \propto |\det J_{f_{\tau'}}(\mathbf{z})|$. Hence, flows with constant Jacobians, i.e., $|\det J_{f_{\tau'}}(\mathbf{z})| = \text{const}$, are temperature steerable.

This still holds for so-called augmented normalizing flows [21], where the prior and target distributions are augmented with a Gaussian distribution $p_A^\tau(\mathbf{q}) = \mathcal{N}(\mathbf{q}|0, \tau)$ and $p_A^\tau(\mathbf{p}) = \mathcal{N}(\mathbf{p}|0, \tau)$, respectively. This augmented flow f_τ is trained to match the output distribution $p_{X,A}^\tau(\mathbf{x}, \mathbf{p})$ with the joint target distribution $\mu_X^\tau(\mathbf{x})p_A^\tau(\mathbf{p})$. The auxiliary variables can be interpreted as physical momenta [6], making the architecture similar to the Hamiltonian Monte Carlo method [22]. However, in contrast we do not propagate the system by Hamiltonian dynamics, but learn a (deterministic) flow, as in Hamiltonian flows [23,24]. As we are mostly interested in the Boltzmann distribution $\mu_X(\mathbf{x})$ of the positions, this architecture can also be viewed as a stochastic normalizing flow [12].

To generate configuration samples $\mathbf{x}$ from the marginal output distribution $p_X^\tau(\mathbf{x})$ at temperature τ, we follow three consecutive steps: (1) sample the latent configuration $\mathbf{z} \sim p_Z^\tau(\mathbf{z})$ and auxiliary momenta $\mathbf{q} \sim p_A^\tau(\mathbf{q})$, and define the point in phase space $(\mathbf{z}, \mathbf{q})$; (2) propagate the point in phase space by the flow $(\mathbf{x}, \mathbf{p}) = f_\tau(\mathbf{z}, \mathbf{q})$; and (3) project onto the configuration variables $\mathbf{x}$.

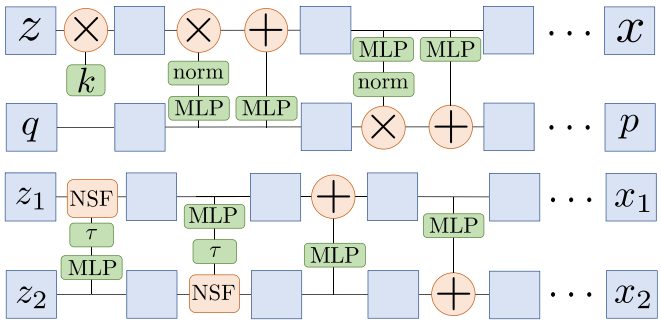

FIG. 1. Temperature steerable flow architectures based on coupling layers, which include elementwise multiplication ($\times$) and addition ($+$). Top: Auxiliary momenta $\mathbf{q}$ and coordinates $\mathbf{z}$ are coupled with volume-preserving networks where the outputs of the multilayer perceptron (MLP) used to generate the scaling variables are normalized. The first layer multiplies the latent space coordinates $\mathbf{z}$ with a scalar factor k, which adjusts for the difference in entropy between latent and phase space. Bottom: Temperature steerable neural spline flow architecture. Samples from the uniform distribution are split into two channels which are conditioned on the neural spline flow (NSF) transformation of the other channel. The parameters for the flow are transformed to the given temperature τ. This is followed by several layers of volume-preserving transformations [25]. See Supplemental Material [10] for a description of the flow transformations.

An expressive volume-preserving dynamics, i.e., $|\det J_{f_\tau}(\mathbf{z}, \mathbf{q})| = 1$, is obtained by altering real-valued non-volume-preserving transformations [25], such that the product of the outputs of the scaling layers is equal to unity. This is done by subtracting the mean of the log outputs from each scaling layer as in Ref. [26]. In addition to these volume-preserving layers we scale the latent space coordinates by a trainable scalar, which allows us to adjust for the entropy difference between the prior and the target. The resulting flow architecture, which still fulfills the scaling condition (4), is shown in Fig. 1 (top).

As the flow fulfills the temperature scaling condition, a temperature change of the augmented prior, i.e., $\tau \to \tau'$, changes the output accordingly. In the case of a factorized output distribution $p_{X,A}^\tau(\mathbf{x}, \mathbf{p}) = p_X^\tau(\mathbf{x})p_A^\tau(\mathbf{p})$, the marginal output distribution $p_X^\tau(\mathbf{x})$ is scaled correctly with the temperature as well. This is ensured if the joint target distribution $\mu_{X,A}^\tau(\mathbf{x}, \mathbf{p}) = \mu_X^\tau(\mathbf{x})p_A^\tau(\mathbf{p})$ is matched correctly.

C. Temperature steerable flows with uniform prior

Instead of a Gaussian prior, one can also use a uniform prior distribution on the unit box $[0, 1]^d$ in combination with a single flow layer that scales with the temperature to construct a TSF. While finding a flow architecture that precisely reproduces the temperature scaling property is difficult, a good approximation is obtained using neural spline flows [27,28]. With this type of flow we can adjust the parameters given the temperature, such that the temperature scaling is

approximately correct (see Supplemental Material [10]). In addition we combine it with volume-preserving flows, i.e., nonlinear independent component estimation (NICE) [29], to obtain a more expressive transformation [see Fig. 1 (bottom)].

D. Training

As in Ref. [11], the flows are trained by a combination of a maximum-likelihood and energy-based loss. Maximum-likelihood training minimizes the negative log likelihood

$$\begin{aligned}\mathcal{L}_{\mathrm{ML}} &= \left\langle -\log p^{\tau}_{X,A} \right\rangle_{\mu^{\tau}_{X,A}} \\ &= \left\langle -\log p_{Z,A}\left[f^{-1}_{\tau}(\mathbf{x},\mathbf{p})\right] \right. \\ &\quad \left. - \log\left|\det J_{f^{-1}_{\tau}}(\mathbf{x},\mathbf{p})\right| \right\rangle_{\mathbf{x},\mathbf{p}\sim\mu^{\tau}_{X}(\mathbf{x})p^{\tau}_{A}(\mathbf{p})},\end{aligned} \tag{5}$$

which agrees with the forward Kullback-Leibler divergence up to a constant. Computing this expectation requires samples from the product distribution $\mu^{\tau}_{X}(\mathbf{x})p^{\tau}_{A}(\mathbf{p})$, where the configurations $\mathbf{x}$ are generated by (MD) simulations, and momenta $\mathbf{p}$ are independent Gaussian noise.

As the target energy $u(\mathbf{x},\mathbf{p})$ is defined by the physical system of interest, we can also use energy-based training, which minimizes the reverse Kullback-Leibler divergence

$$\begin{aligned}\mathcal{L}_{\mathrm{KL}} &= \left\langle -\log \mu^{\tau}_{X,A} + \log p^{\tau}_{X,A} \right\rangle_{p^{\tau}_{X,A}} \\ &= \left\langle \tau^{-1}\left(U(\mathbf{x}) + \frac{|\mathbf{p}|^2}{2}\right) - \log\left|\det J_{f_\tau}(\mathbf{z},\mathbf{q})\right| \right\rangle_{\mathbf{z},\mathbf{q}\sim p^{\tau}_{Z}(\mathbf{z})p^{\tau}_{A}(\mathbf{q})} \\ &\quad + \text{const},\end{aligned} \tag{6}$$

with $(\mathbf{x},\mathbf{p}) = f_\tau(\mathbf{z},\mathbf{q})$. This expectation is computed over the thermodynamic ensemble generated by the flow at a given temperature. A TSF trained with Eqs. (5) and (6) will implicitly learn a representation of the Boltzmann distribution that is transferable across temperatures. It can still be useful to combine different target temperatures during training to broaden the range of temperatures at which the TSF performs well.

Furthermore, we can also combine training by example and training by energy [11] using a convex combination $\mathcal{L} = (1-\lambda)\mathcal{L}_{\mathrm{ML}} + \lambda\mathcal{L}_{\mathrm{KL}}$.

E. Unbiased sampling: Importance sampling and latent Monte Carlo

As in Ref. [11], we use two different methods to produce unbiased samples from the target distribution μ^{τ}_{X}. First, we employ the flow as an importance sampler and compute thermodynamic observables $\langle o\rangle_{\mu^{\tau}_{X}}$ by Zwanzig reweighting,

$$\langle o\rangle_{\mu^{\tau}_{X}} = \frac{\langle o\cdot e^{-U/\tau-\log p^{\tau}_{X}}\rangle_{p^{\tau}_{X}}}{\langle e^{-U/\tau-\log p^{\tau}_{X}}\rangle_{p^{\tau}_{X}}}. \tag{7}$$

Second, we extend the flow-based MCMC moves from Ref. [11] to the (augmented) phase space.

A proposal $\mathbf{x}'$ is generated from configuration $\mathbf{x}$ by sampling auxiliary momenta $\mathbf{p}\sim p^{\tau}_{A}(\mathbf{p})$, then applying the inverse dynamics $(\mathbf{z},\mathbf{q}) = f^{-1}_{\tau}(\mathbf{x},\mathbf{p})$, followed by a random displacement $(\mathbf{z}',\mathbf{q}') = (\mathbf{z}+\Delta\mathbf{z},\mathbf{q}+\Delta\mathbf{q})$, with $\Delta\mathbf{z},\Delta\mathbf{q}\sim\mathcal{N}(0,\sigma^2)$, and finally transforming back $(\mathbf{x}',\mathbf{p}') = f_\tau(\mathbf{z}',\mathbf{q}')$. Accepting such a step with probability

$$\begin{aligned}&p^{\tau}_{\mathrm{acc}}[(\mathbf{x},\mathbf{p})\to(\mathbf{x}',\mathbf{p}')] \\ &= \min\left\{1, \exp\left[-\tau^{-1}\left(U(\mathbf{x}')-U(\mathbf{x})+\frac{\|\mathbf{p}'\|^2}{2}-\frac{\|\mathbf{p}\|^2}{2}\right)\right]\right\}\end{aligned} \tag{8}$$

guarantees detailed balance in configuration space and thus ensures convergence to the Boltzmann distribution. As the TSF is able to generate distributions at several temperatures, we can combine the MCMC moves with PT [4,30,31]. Additionally to TSF-MCMC steps at a set of temperatures, samples can be randomly exchanged between two randomly chosen temperatures with the usual acceptance probability. A summary of the sampling algorithm is given in the Supplemental Material [10].

III. EXPERIMENTS

We carry out experiments for the XY model and two small peptides, showing that TSFs can sample the respective Boltzmann distributions at different temperatures efficiently. The resulting flows are used to compute observables at low-temperature states from high-temperature simulations and compute temperature-dependent quantities such as absolute free energy from samples at a single thermodynamic state (see Supplemental Material [10]).

A. XY model

As an example with angular symmetry, we investigate the XY model, which can be considered a continuous state-space version of the Ising model. In our experiments we consider quadratic two-dimensional lattices with $N\times N$ spins. Each spin has a continuous angle $\theta_i\in[-\pi;\pi]$ and is represented by $\mathbf{s}_i = (\cos\theta_i,\sin\theta_i)^T$. Each spin interacts with its four nearest neighbors and an external field $\mathbf{h} = (h,0)$. Hence, the Hamiltonian of the system is given by

$$\mathcal{H}(s_1,\ldots,s_{N^2}) = -J\sum_{\langle ij\rangle}\mathbf{s}_i\cdot\mathbf{s}_j - \sum_i \mathbf{h}\cdot\mathbf{s}_i,$$

where $\sum_{\langle ij\rangle}$ denotes the sum over all nearest-neighbor pairs with periodic boundaries and J is the interaction constant. For our experiments we select the parameters as $J = h = k_B T_0$ and a lattice of 16×16 spins. As observable, we select the mean-squared magnetization per spin $\langle M^2\rangle/N^2 = N^{-2}\langle\sum_i \mathbf{s}_i\cdot\mathbf{s}_i\rangle$ at a given temperature. For producing reference configurations, we use long runs of Glauber dynamics [32]. The TSF consists of a uniform prior, a temperature scaled NSF, followed by seven blocks of circular NICE [see Fig. 1 (bottom) and Supplemental Material [10] for details]. Training is performed solely with the energy-based loss [Eq. (6)]. Since NSFs are only approximately temperature scaling, we use a convex combination of temperatures $T = \{0.5, 1.0, 1.3\}T_0$ for training. We generate data sets at temperatures ranging $0.5T_0, 0.6T_0, \ldots, 1.0T_0$ and observe an excellent overlap of the reweighted energies at the highest and lowest temperature with the reference configurations [Fig. 2 (top)]. Furthermore,

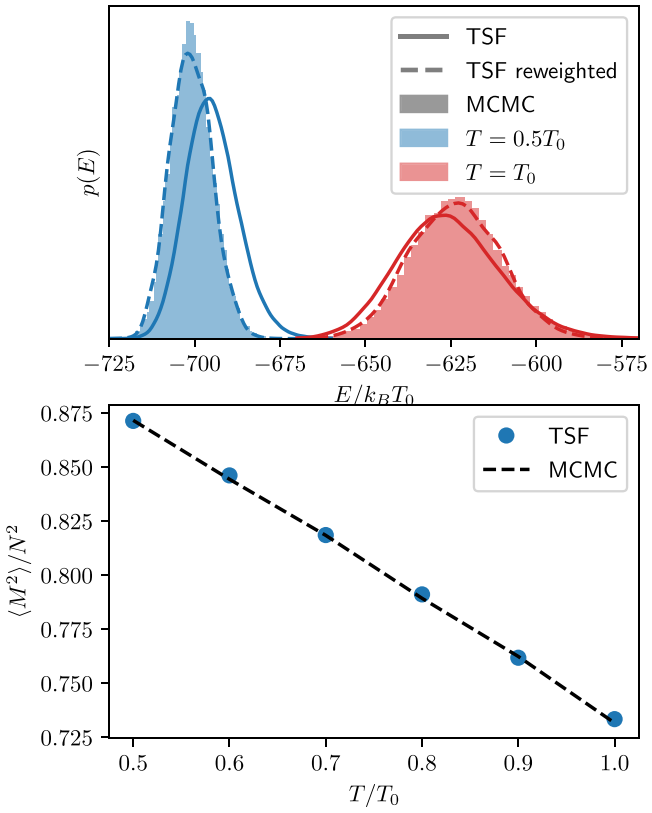

FIG. 2. Results of the TSF trained on the two-dimensional XY model. Top: Distribution of energies at the training temperatures T_0 and $0.5T_0$ obtained by MCMC, TSF, and TSF with reweighting. Bottom: Magnetization as a function of the temperature compared between the TSF and MCMC samples.

we compare the mean-squared magnetization per spin and again find excellent agreement between TSF and Glauber dynamics [Fig. 2 (bottom)].

B. Alanine di- and tetrapeptide

We further test TSFs on the alanine di- and tetrapeptide molecules in an implicit solvent model. For this system we use an invertible coordinate transformation layer and operate the TSF on a representation of the molecule in terms of distances and angles. Our goal is to use samples at $T = 600$ K to train the TSF and then use the TSF to sample at $T = 300$ K, comparing it to a MD simulation at $T = 300$ K.

1. Alanine tetrapeptide

With the TSF we are able to generate samples at $T = 300$ K that closely resemble the equilibrium distribution. To demonstrate this, we project the configurations into the space of the slowest transition between states. These are determined by a time-independent component analysis (TICA [33]) from an exhaustive MD simulation at $T = 300$ K. We observe generally good agreement with MD simulations at the low temperature [Figs. 3(e)–3(g)] in the relevant slowest coordinates, while slightly underestimating the barrier height [Fig. 3(g)].

2. Alanine dipeptide

We use the TSF to generate samples in configuration space and compare the Ramachandran plots. At $T = 300$ K [Figs. 3(a) and 3(b)] the TSF still finds the major minima at around $\phi \approx -2$, but undersamples the minimum at $\phi \approx 1$. This deviation from the target distribution likely stems from limited expressivity of the flow. We further utilize the TSF to compare the free-energy difference of the two states along the ϕ axis as a function of the temperature [Fig. 3(c)]. We observe an exact match at the training temperature and slight deviations when moving away from it. To recover the correct distribution along the ϕ angles, we use the Monte Carlo scheme in a PT fashion with eight temperatures in the range 300–600 K [see Supplemental Material [10] for details].

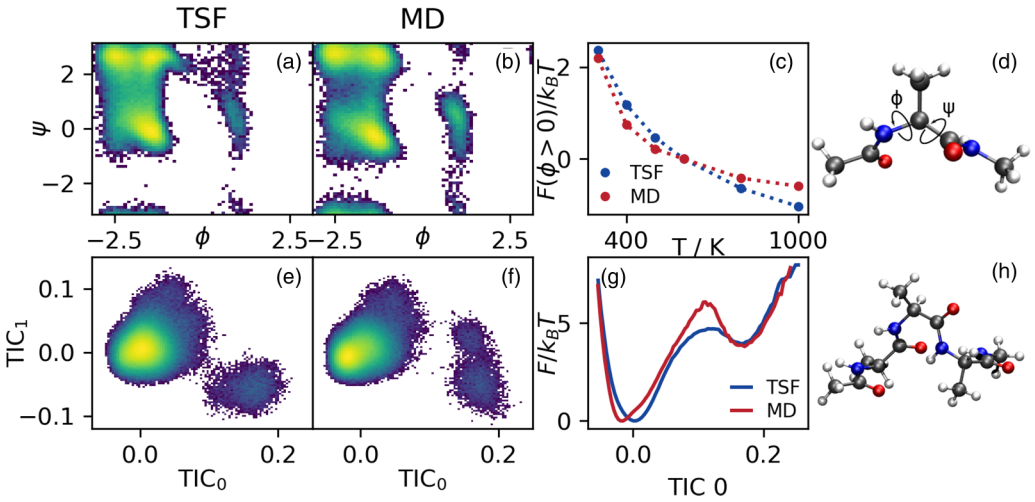

FIG. 3. Results for alanine dipeptide (ala2) [(d), first row] and alanine tetrapeptide (ala4) [(h), second row] in implicit solvent. All TSFs are trained from samples at 600 K. (a), (b) Ramachandran plots produced by the TSF and MD at 300 K. (c) Comparison of the free-energy difference of the two metastable states along the ϕ axis. (e), (f) Comparison of the first two time-independent components (TICs) of ala4 at the sampling temperature of 300 K. (g) The free energy along the first TIC.

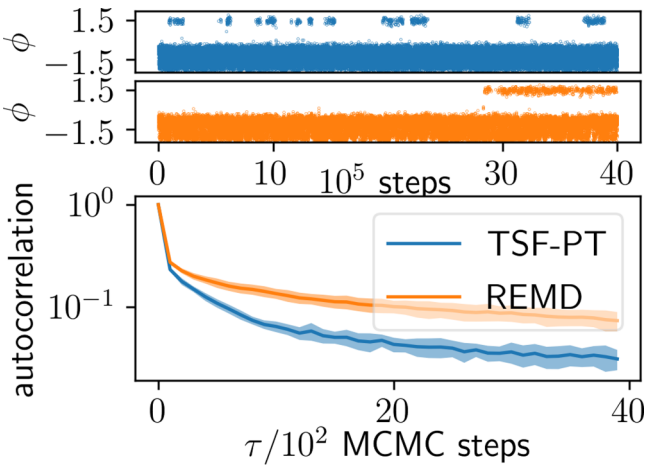

FIG. 4. Comparison of replica exchange molecular dynamics simulation (REMD) and the TSF with parallel tempering (TSF-PT), operating in a PT scheme on eight different temperatures in the range between 300 and 600 K. Top: One example trajectory of ϕ angles of the ten independent runs. Despite being over two times longer, no transitions are observed in three out of the ten REMD runs, while all of the TSF-PT trajectories cross many times between the two metastable states. Bottom: Autocorrelation of the ϕ angle as a function of underlying MCMC/MD steps. The autocorrelation function decays more rapidly in the TSF-PT method. This hints toward this method being more sampling efficient. Nontraversing trajectories of the REMD method were excluded in the calculation of the autocorrelation.

To assess the efficiency of the sequential sampling procedure, we compare it to the replica exchange molecular dynamics simulation (REMD) at the same temperatures. We observe that for ten independent runs, in REMD only seven transition between the metastable states within 10 M steps, while all ten transition with the TSF-PT method, which additionally only consists of 4 M steps [Fig. 4 (top)]. Furthermore, the autocorrelation of the slowest process [Fig. 4 (bottom)], which are the transitions along the ϕ angle, decays considerably faster in the TSF-PT method. In addition, we compare the methods based on their sampling efficiency $\eta = N_{\text{eff}}/N$, where N is the number of underlying MCMC steps and N_{eff} is the effective sample size (see Supplemental Material [10] for details). Table I shows that the TSF-PT method produces independent samples at about four times the rate of the REMD method.

TABLE I. Efficiency as the number of effective steps per underlying sampling step for different sampling methods.

Method	TSF-PT	REMD	MD (600 K)	MD (300 K)
$\eta \times 10^4$	1.36	0.32	0.38	0.02

IV. DISCUSSION

In this Letter, we derived and constructed temperature steerable flows (TSFs) that correctly scale the output distribution of a BG with temperature. To this end we formulated a condition for such flows and introduced two different methods of constructing them. We showed that this type of flow can be used to train a BG at one temperature and generate distributions at lower temperatures.

For the XY model we were able to predict the correct temperature dependence of the magnetization. Furthermore, we showed for alanine dipeptide that the efficiency of parallel tempering can be improved by using our TSF for the MCMC proposals at different temperatures. Further progress could be made by combining samples at different temperatures when collecting training data and thus improve the quality of the BG.

While the presented results demonstrate the promise and uniqueness of the TSF method, practical applications to high-dimensional physical systems of interest will likely require further modifications to the network architecture. Future work should consider combining TSF with conditioner networks that respect the symmetries of the potential energy. In this spirit, the presented temperature scaling property complements existing equivariant flows that maintain group transformations such as rotation and permutation [34–37].

ACKNOWLEDGMENTS

We gratefully acknowledge support by the Deutsche Forschungsgemeinschaft (SFB1114, Projects No. C03, No. A04, and No. B08), 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, Michele Invernizzi, and Yaoyi Chen for insightful discussions. Moreover, we thank the anonymous reviewers for their constructive feedback.

[1] M. Böttcher, S. Heinze, S. Egorov, J. Sinova, and B. Dupé, *B-T* phase diagram of Pd/Fe/Ir(111) computed with parallel tempering Monte Carlo, New J. Phys. **20**, 103014 (2018).

[2] R. Wuttke, H. Hofmann, D. Nettels, M. B. Borgia, J. Mittal, R. B. Best, and B. Schuler, Temperature-dependent solvation modulates the dimensions of disordered proteins, Proc. Natl. Acad. Sci. U.S.A. **111**, 5213 (2014).

[3] D. J. Earl and M. W. Deem, Parallel tempering: Theory, applications, and new perspectives, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[4] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo Simulation of Spin-Glasses, Phys. Rev. Lett. **57**, 2607 (1986).

[5] I. Kobyzev, S. Prince, and M. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. **43**, 3964 (2021).

[6] S.-H. Li, C.-X. Dong, L. Zhang, and L. Wang, Neural Canonical Transformation with Symplectic Flows, Phys. Rev. X **10**, 021020 (2020).

[7] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, J. Mach. Learn. Res. **22**, 1 (2021).

[8] D. Rezende and S. Mohamed, Variational inference with normalizing flows, in *Proceedings of the 32nd International Conference on Machine Learning*, edited by F. Bach and D. Blei, Proceedings of Machine Learning Research Vol. 37 (PMLR, 2015), pp. 1530–1538.

[9] E. G. Tabak and E. Vanden-Eijnden, Density estimation by dual ascent of the log-likelihood, Commun. Math. Sci. **8**, 217 (2010).

[10] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.4.L042005 for a more detailed introduction of normalizing flows, more experiments, as well as additional information.

[11] F. Noé, S. Olsson, J. Köhler, and H. Wu, Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning, Science **365**, eaaw1147 (2019).

[12] H. Wu, J. Köhler, and F. Noé, Stochastic normalizing flows, Adv. Neural Inf. Process. Syst. **33**, 5933 (2020).

[13] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models, Phys. Rev. Lett. **126**, 032001 (2021).

[14] S. Li and L. Wang, Neural Network Renormalization Group, Phys. Rev. Lett. **121**, 260601 (2018).

[15] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K. Müller, and P. Kessel, Asymptotically unbiased estimation of physical observables with neural samplers, Phys. Rev. E **101**, 023304 (2020).

[16] M. S. Albergo, G. Kanwar, S. Racanière, D. J. Rezende, J. M. Urban, D. Boyda, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Flow-based sampling for fermionic lattice field theories, Phys. Rev. D **104**, 114507 (2021).

[17] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Flow-based generative models for Markov chain Monte Carlo in lattice field theory, Phys. Rev. D **100**, 034515 (2019).

[18] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Sampling using SU(N) gauge equivariant flows, Phys. Rev. D **103**, 074504 (2021).

[19] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racaniere, D. J. Rezende, and P. E. Shanahan, Equivariant Flow-Based Sampling for Lattice Gauge Theory, Phys. Rev. Lett. **125**, 121601 (2020).

[20] H. E. Stanley, Dependence of Critical Properties on Dimensionality of Spins, Phys. Rev. Lett. **20**, 589 (1968).

[21] C. Huang, L. Dinh, and A. C. Courville, Augmented normalizing flows: Bridging the gap between generative flows and latent variable models, arXiv:2002.07101.

[22] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, Hybrid Monte Carlo, Phys. Lett. B **195**, 216 (1987).

[23] S. Greydanus, M. Dzamba, and J. Yosinski, Hamiltonian neural networks, Adv. Neural Inf. Process. Syst. **32**, 15379 (2019).

[24] P. Toth, Danilo Jimenez Rezende, A. Jaegle, S. Racanière, A. Botev, and I. Higgins, Hamiltonian generative networks, arXiv:1909.13789.

[25] L. Dinh, J. Sohl-Dickstein, and S. Bengio, Density estimation using real NVP, arXiv:1605.08803.

[26] P. Sorrenson, C. Rother, and U. Köthe, Disentanglement by nonlinear ICA with general incompressible-flow networks (GIN), arXiv:2001.04872.

[27] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, Neural spline flows, Adv. Neural Inf. Process. Syst. **32**, 32 (2019).

[28] D. J. Rezende, G. Papamakarios, S. Racaniere, M. Albergo, G. Kanwar, P. Shanahan, and K. Cranmer, Normalizing flows on tori and spheres, in *Proceedings of the 37th International Conference on Machine Learning*, edited by H. Daumé III and A. Singh, Proceedings of Machine Learning Research, Vol. 119 (PMLR, 2020), pp. 8083–8092.

[29] L. Dinh, D. Krueger, and Y. Bengio, NICE: Nonlinear independent components estimation, arXiv:1410.8516.

[30] C. J. Geyer, Markov chain Monte Carlo maximum likelihood, in *Proceedings of the 23rd Symposium on the Interface*, edited by E. M. Keramidas (Interface Foundation of North America, Fairfax Station, VA, 1991), pp. 156–163.

[31] K. Hukushima and K. Nemoto, Exchange Monte Carlo method and application to spin glass simulations, J. Phys. Soc. Jpn. **65**, 1604 (1996).

[32] R. J. Glauber, Time-dependent statistics of the Ising model, J. Math. Phys. **4**, 294 (1963).

[33] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, Identification of slow molecular order parameters for Markov model construction, J. Chem. Phys. **139**, 015102 (2013).

[34] J. Köhler, L. Klein, and F. Noé, Equivariant flows: exact likelihood generative learning for symmetric densities, in *Proceedings of the 37th International Conference on Machine Learning* (Ref. [28]), pp. 5361–5370.

[35] D. J. Rezende, S. Racanière, I. Higgins, and P. Toth, Equivariant Hamiltonian flows, arXiv:1909.13739.

[36] V. G. Satorras, E. Hoogeboom, F. Fuchs, I. Posner, and M. Welling, $E(n)$ equivariant normalizing flows, in *Advances in Neural Information Processing Systems*, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (Curran Associates, Inc., 2021) Vol. 34, pp. 4181–4192.

[37] L. Zhang, Weinan E, and L. Wang, Monge-Ampère flow for generative modeling, arXiv:1809.10188.