**DEUTSCHES ELEKTRONEN-SYNCHROTRON**
**Ein Forschungszentrum der Helmholtz-Gemeinschaft**

DESY 20-148
HU-EP-20/16
arXiv:2007.07115
July 2020

# Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models

K. A. Nicoli, C. J. Anders, P. Kessel

*Machine Learning Group, Technische Universität, Berlin*

L. Funcke

*Perimeter Institute for Theoretical Physics, Waterloo, Canada*

T. Hartung

*Department of Mathematics, King's College London, UK*

K. Jansen

*John von Neumann-Institut für Computing NIC/DESY, Zeuthen*

S. Nakajima

*Machine Learning Group, Technische Universität, Berlin*
*and*
*RIKEN Center for AIP, Tokyo, Japan*

P. Stornati

*John von Neumann-Institut für Computing NIC/DESY, Zeuthen*
*and*
*Institut für Physik, Humboldt-Universität, Berlin*

ISSN 0418–9833

**PLATANENALLEE 6 - 15738 ZEUTHEN**

To be sure that your reports and preprints are promptly included in the
HEP literature database
send them to (if possible by air mail):

DESY
Zentralbibliothek
Notkestraße 85
22607 Hamburg
Germany

DESY
Bibliothek
Platanenallee 6
15738 Zeuthen
Germany

HU-EP-20/16

# Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models

Kim A. Nicoli,[1] Christopher J. Anders,[1] Lena Funcke,[2] Tobias Hartung,[3]
Karl Jansen,[4] Pan Kessel,[1] Shinichi Nakajima,[1, 5] and Paolo Stornati[4, 6]

[1]*Machine Learning Group, Technische Universität Berlin, Berlin, Germany*
[2]*Perimeter Institute for Theoretical Physics, Waterloo, Canada*
[3]*Department of Mathematics, King's College London, London, United Kingdom*
[4]*NIC, DESY Zeuthen, Zeuthen, Germany*
[5]*RIKEN Center for AIP, Tokyo, Japan*
[6]*Institut für Physik, Humboldt-Universität, Berlin, Germany*

In this work, we demonstrate that applying deep generative machine learning models for lattice field theory is a promising route for solving problems where Markov Chain Monte Carlo (MCMC) methods are problematic. More specifically, we show that generative models can be used to estimate the absolute value of the free energy, which is in contrast to existing MCMC-based methods which are limited to only estimate free energy differences. We demonstrate the effectiveness of the proposed method for two-dimensional $\phi^4$ theory and compare it to MCMC-based methods in detailed numerical experiments.

*Introduction.* The free energy of a physical system is of great importance since it can be related to several thermodynamical observables. In particular, at non-zero temperature, it allows to compute the entropy, the pressure or, more generally, the equation of state of the considered physical system. For example, QCD at high temperature –as a generic strongly interacting field theory– plays an essential role in the physics of the early universe and is now extensively probed in large-scale heavy ion experiments [1]. Hence, knowing such thermodynamic quantities from QCD alone is of very high relevance.

The main tool to study strongly-coupled field theories, such as QCD, is to discretize them on a spacetime lattice and use Monte-Carlo Markov-Chain (MCMC) methods to numerically calculate the relevant physical quantities. Unfortunately, these thermodynamical quantities are challenging to compute using existing MCMC methods. The fundamental difficulty is that MCMC is not able to directly estimate the partition function of the lattice field theory. Therefore, the absolute value of the free energy cannot be estimated straightforwardly.

Instead, there are a number of MCMC methods to estimate differences of free energies. One typically chooses a free energy difference $\Delta F = F_b - F_a$ such that $F_a$ is known either exactly or approximately. One can then deduce the value of the free energy $F_b = \Delta F + F_a$ at the desired point in parameter space. If the free energy $F_a$ is not known exactly, this induces an unwanted approximation error. Most of the methods to estimate $\Delta F$ rely on integrating a derivative of the partition function over a trajectory in the parameter space of the lattice field theory [2]. Alternatively, one can use a reweighting procedure to calculate free energy differences between neighbouring points of the discretized trajectory and then sum them up [2, 3]. These approaches require simulations at each parameter point of the discretized trajectory which is numerically costly and leads to accumulation of errors. This effect is often the dominant contribution to the error - especially if the trajectory passes a phase transition. Such situations arise for example in the context of studying the deconfined phase of $SU(3)$ Yang–Mills theory [4, 5]. We stress that the accumulation of the statistical error along the trajectory and the approximation error of its starting point are not independent. The former could be reduced if a better starting point was available. There are also non-equilibrium methods based on Jarzynski's identity to estimate free energy differences without the need for integration [5–7]. However, also these methods require expensive repeated simulations corresponding to an ensemble of non-equilibrium trajectories through phase-space.

It is therefore desirable to develop methods which allow the direct estimation of the free energy at a given point in parameter space.

In the following, we will propose such a method based on deep generative machine learning models. As we will discuss, our method comes with rigorous error estimators and asymptotic guarantees. Over the last years, deep generative models have been applied with great success to generate, for example, high-resolution images, natural speech, and text (see [8] for an overview). In [9], a machine-learning-based regression algorithm for determining action parameters from an ensemble of field configurations is proposed and [10] uses a neural network to predict the structure of phase transitions from field configurations. References [11–15] conjecture a relation between Restricted Boltzmann Machines and Quantum Fields in the context of the holographic duality. In the recent works [16–18], deep generative models have also been used in the context of lattice quantum field theories (see also [19, 20]). The main objective of these works was to reduce the integrated autocorrelation of the simulations. In contrast, this work demonstrates that deep generative models can be used to estimate quantities which are not (directly) obtainable by MCMC approaches.

We also note that generative models have been used in [21] to estimate free energy differences in the context of statistical mechanics by combining these models with the Zwanzig free energy perturbation method [22]. Contrary to this approach, our method estimates the absolute value of the free energy. We furthermore note that the free energy can also be directly computed using the Tensor Renormalization Group method, see [23] for an application to $\phi^4$-theory. For other novel approaches to obtain thermodynamic quantities and, in particular, the equation of state, see [24, 25].

In the following, we will give a brief overview of relevant aspects of lattice field theories and generative models. We will then discuss how generative models can be used to estimate the free energy and compare this approach to MCMC-based methods in numerical experiments.

*Lattice Field Theory.* A lattice field theory can be described by an action $S(\phi)$. In the following, we will consider (euclidian) real scalar field theory for concreteness, i.e. $\phi(x) \in \mathbb{R}$ for each lattice site $x \in \Lambda$ of the lattice $\Lambda$. The path integral then reduces to an ordinary high-dimensional integral. Therefore, expectation values of operators $\mathcal{O}(\phi)$ can be calculated by

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}[\phi]\, \mathcal{O}(\phi) \exp(-S(\phi)) \,,$$

where we defined $\mathcal{D}[\phi] = \prod_{x\in\Lambda} \mathrm{d}[\phi(x)]$ and the partition function $Z$ is given by

$$Z = \int \mathcal{D}[\phi]\, \exp(-S(\phi)) \,.$$

If we impose periodic boundary conditions in time for a lattice with temporal extend $N_T$, the theory is at finite temperature $T = \frac{1}{\beta} = \frac{1}{N_T a}$, where $a$ denotes the lattice spacing. The free energy is then defined by

$$F = -T\, \ln(Z) \,, \tag{1}$$

and can be related to the pressure $p = -\frac{F}{V}$, where $V$ denotes the spatial volume of the lattice $\Lambda$ whose number of lattice sites we denote by $|\Lambda|$. Similarly, the entropy $H$ can be obtained from the free energy by $F = U - TH$, where the $U$ is the internal energy.

*Deep Generative Models.* We focus on a particular subclass of generative models called normalizing flows (see [26] for a recent review). These flows are distributions $q_\theta$ with learnable parameters $\theta$. They also have the appealing property that they allow for efficient sampling and calculation of the probability of the samples.

In more detail, these flows are constructed by defining an invertible neural network $g_\theta$. For a brief overview of neural networks, we refer to the Supplement. The samples $\phi \in \mathbb{R}^{|\Lambda|}$ are obtained by applying this network to samples $z \in \mathbb{R}^{|\Lambda|}$ drawn from a simple prior distribution $q_Z$ such as a standard normal $\mathcal{N}(0,1)$:

$$\phi = g_\theta(z) \,, \qquad z \sim q_Z \,. \tag{2}$$

Since the network $g_\theta$ is invertible by assumption, it then follows by the change of variable theorem that $\phi \sim q_\theta$ with

$$q_\theta(\phi) = q_Z(g_\theta^{-1}(\phi)) \left| \frac{\mathrm{d}g_\theta}{\mathrm{d}z} \right|^{-1} \,. \tag{3}$$

The architecture of the neural network $g_\theta$ is chosen such that i.) invertibility of $g_\theta$ and ii.) efficient evaluation of the Jacobian determinant $\left|\frac{\mathrm{d}g_\theta}{\mathrm{d}z}\right|$ are ensured. A particular example of such an architecture is *Non-linear Independent Component Estimation* (NICE) [27] for which the neural network $g_\theta$ consists of invertible coupling layers $y^l : \mathbb{R}^{|\Lambda|} \to \mathbb{R}^{|\Lambda|}$, i.e.

$$g_\theta(z) = \left(y^L \circ y^{L-1} \circ \cdots \circ y^1\right)(z) \tag{4}$$

Invertiblity and efficient evaluation of Jacobian determinant is then ensured by splitting the components of the layer $y^l = (y^l_u, y^l_d)$ in two parts $y^l_u \in \mathbb{R}^{|\Lambda|-k}$ and $y^l_d \in \mathbb{R}^k$ for given $k \in \{1, |\Lambda|-1\}$. The layer $y^{l+1} = (y^{l+1}_u, y^{l+1}_d)$ is then recursively defined by

$$\begin{aligned} y^{l+1}_u &= y^l_u \,, \\ y^{l+1}_d &= y^l_d + m(y^l_u) \,, \end{aligned} \tag{5}$$

where $m$ is another neural network (not necessarily satisfying the two requirements from above). Due to the splitting, this can be easily inverted by

$$\begin{aligned} y^l_u &= y^{l+1}_u \,, \\ y^l_d &= y^{l+1}_d - m(y^{l+1}_u) \,, \end{aligned}$$

and the determinant of the Jacobian is given by

$$\det \frac{\partial y^{l+1}}{\partial y^l} = \begin{vmatrix} \frac{\partial y^{l+1}_u}{\partial y^l_u} & \frac{\partial y^{l+1}_u}{\partial y^l_d} \\ \frac{\partial y^{l+1}_d}{\partial y^l_u} & \frac{\partial y^{l+1}_d}{\partial y^l_d} \end{vmatrix} = \begin{vmatrix} \mathbb{I} & 0 \\ * & \mathbb{I} \end{vmatrix} = 1 \,.$$

The total Jacobian determinant is then $\left|\frac{\mathrm{d}g_\theta}{\mathrm{d}z}\right| = 1$ since it is the product of the Jacobian determinant of each layer.

*Training.* We want to train a generative model which samples field configurations $\phi \sim q_\theta$ approximately from the path-integral distribution

$$p(\phi) = \frac{1}{Z}\, \exp(-S(\phi)) \,. \tag{6}$$

For this, the Kullback–Leibler divergence [28] between the normalizing flow $q_\theta$ and the target distribution $p$ is minimized, i.e.

$$\begin{aligned} \mathrm{KL}(q_\theta || p) &= \int \mathcal{D}[\phi]\, q_\theta(\phi)\, \ln\left(\frac{q_\theta(\phi)}{p(\phi)}\right) \\ &= \beta\left(F_q - F\right) \,, \end{aligned}$$

where we have defined the variational free energy

$$\beta F_q = \mathbb{E}_{\phi \sim q_\theta} \left[ S(\phi) + \ln q_\theta(\phi) \right] \tag{7}$$

as well as the expectation value $\mathbb{E}_q[\mathcal{O}] = \int \mathcal{D}[\phi] q_\theta(\phi) \mathcal{O}(\phi)$ and the free energy $F = -\frac{1}{\beta} \ln(Z)$. This divergence vanishes if and only if the distributions $q$ and $p$ are identical [29].

The KL divergence is minimized by gradient descent with respect to the parameter $\theta$ of the flow $q_\theta$. Since the free energy $F$ does not depend on the flow $q$, the variational free energy $F_q$ can equivalently be minimized. Therefore, the training procedure does not require a target distribution (6) with a tractable partition function $Z$. Using the explicit expression for the probability of the flow (3), we can rewrite the variational free energy as

$$\beta F_q = \mathbb{E}_{z \sim q_Z} \left[ S(g_\theta(z)) - \ln \left| \frac{\mathrm{d} g_\theta}{\mathrm{d} z} \right| (z) + \ln q_Z(z) \right] .$$

In training, the expectation value is approximated by its Monte-Carlo estimate. In machine learning, this approach of learning a model from an unnormalized target distribution is very well established [30–33]. Recently, the same method has been used in the context of lattice field theories [16]. Furthermore, this approach has been applied to quantum chemistry [34] and statistical physics [35–37].

The variational free energy does not allow us to infer the value of the KL divergence since the free energy $F$ is not known. In order to alleviate this shortcoming, we define the random variable $C(\phi) = S(\phi) + \ln q_\theta(\phi)$, which is related to the variational free energy by $\beta F_q = \langle C \rangle_q$. In the Supplement, we show that

$$\mathrm{KL}(q_\theta || p) = \tfrac{1}{2} \operatorname{Var}_q(C) + \mathcal{O}(\mathbb{E}_q[|w-1|^3]) ,$$

where we have defined the importance weight $w(\phi) = \frac{p(\phi)}{q(\phi)}$. Thus convergence of training will result in a small variance $\operatorname{Var}_q(C)$. In practice, a Monte-Carlo estimate of this quantity can be calculated without any significant overhead during training as $C(\phi)$ is also needed for Monte-Carlo estimation of the variational free energy $F_q$, see (7). It is therefore advisable to closely monitor the variance of $C$ during training.

*Estimation of Thermodynamical Observables.* The partition function $Z$ can be rewritten as

$$Z = \int \mathcal{D}[\phi] \, q_\theta(\phi) \, \tilde{w}(\phi) \, , \tag{8}$$

where we have defined the unnormalized importance weight $\tilde{w}(\phi) = \frac{\exp(-S(\phi))}{q_\theta(\phi)}$. Therefore, the partition function can be estimated by Monte-Carlo as follows

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} \tilde{w}(\phi_i) \qquad \text{with} \qquad \phi_i \sim q_\theta \, . \tag{9}$$

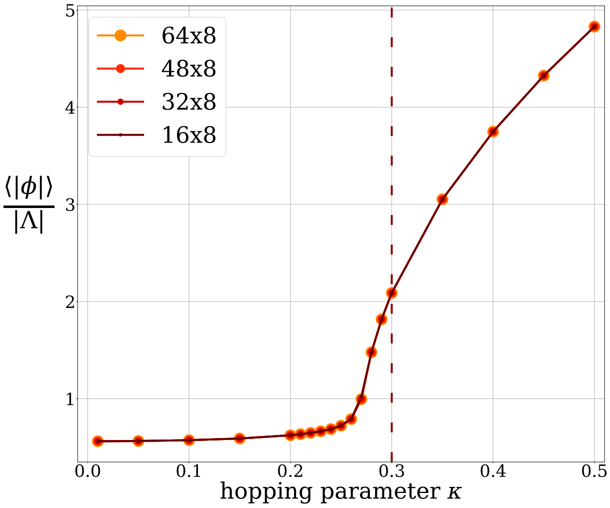

FIG. 1. Absolute magnetization density as a function of hopping parameter $\kappa$ for bare coupling $\lambda = 0.022$. Results for various lattice sizes overlap. The values were estimated with an overrelaxed HMC [38–41]. The dashed line denotes the hopping parameter value $\kappa = 0.3$ for the free energy estimation in the numerical experiments.

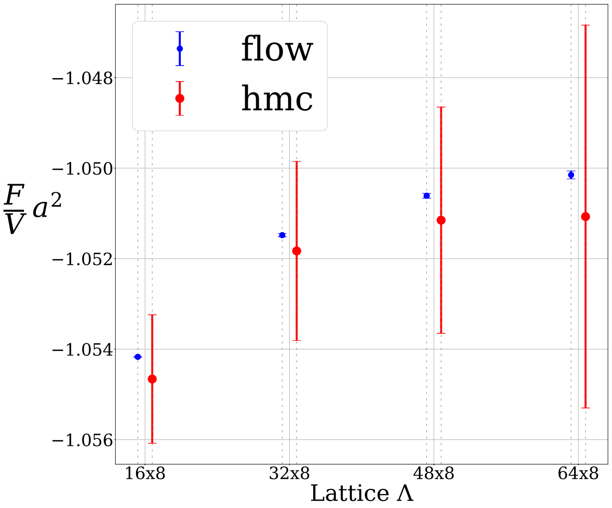

FIG. 2. Estimate of free energy density at $\lambda = 0.022$ and $\kappa = 0.3$ obtained by both the **flow-based** and **MCMC-based** method for various lattice sizes. MCMC estimates are obtained from integrating free energy differences. Both methods use the same number of samples (5.6 M) for estimation. Errors are obtained with the delta and uwerror method [42] for flow and HMC respectively (see Appendix for Jackknife error analysis).

We emphasize that the sampling procedure does not need to be sequential (as for a Markov Chain). As a result, it can very efficiently be parallelized and does not suffer from autocorrelation. From $\hat{Z}$, one can then easily estimate the free energy by

$$\hat{F} = -T \, \ln \hat{Z} \, . \tag{10}$$

From the free energy (10), one can then straightforwardly obtain estimates for the pressure and entropy, as explained above. The estimator (10) has been extensively studied in the context of training an Importance Weighted Variational Autoencoder (IWAE) [43–45]. It was shown in [43] that it is a statistically consistent estimator if $q$ has support larger or equal to the target $p$. In [44], its variance and bias were derived using the delta method (see also [36, 45]). For convenience, we summarize the relevant results in the Supplement. Alternatively, one can use the Jackknife method to estimate the bias and variance [46].

*Numerical Experiments.* We apply the proposed method to two-dimensional real scalar field theory with action

$$S = \sum_{x\in\Lambda} -2\kappa \sum_{\hat{\mu}=1}^{2} \varphi(x)\varphi(x+\hat{\mu}) + (1-2\lambda)\varphi(x)^2 + \lambda\,\varphi(x)^4\,,$$

where $\kappa$ is the hopping parameter and $\lambda$ denotes the bare coupling constant of the theory. The action is invariant under $\mathbb{Z}_2$-transformations, i.e. $\phi \to -\phi$. Figure 1 shows the absolute magnetization $\langle|\phi|\rangle$ as a function of the hopping parameter $\kappa$. As the hopping parameter $\kappa$ increases, spontaneous magnetization is observed.

In the following, we will estimate the free energy $F_e$ at $\lambda_e = 0.022$ and $\kappa_e = 0.3$ for lattice sizes $|\Lambda| = N_L \times N_T$ of $64\times 8$, $48\times 8$, $32\times 8$, $16\times 8$ with both the flow-based and an MCMC-based method.

Using the flow method, we can directly estimate these free energies. We modify the NICE architecture to ensure that the flow $q_\theta$ is invariant under $\mathbb{Z}_2$-transformations, i.e. $q_\theta(\phi) = q_\theta(-\phi)$. By the definition (3) of $q_\theta$, an odd function $g_\theta(-z) = -g_\theta(z)$ implies $\mathbb{Z}_2$-invariance of $q_\theta$. The map $g_\theta$ is odd if all its coupling blocks $y^l$ are odd, see (4). The latter condition can be ensured by choosing an odd neural network $m$ for the coupling (5) which we achieve by using tanh non-linearities and vanishing biases for the network $m$.

After training has completed, the free energy is then computed using the proposed estimator (10). For error analysis, we use both the Jackknife as well as the delta-method and check that they lead to consistent error estimates. In many applications, generative models suffer from *mode dropping* [47], i.e. some modes of the target $p$ are not captured by the model $q_\theta$. For our specific estimation method however, a simple consistency check can be performed ensuring that mode dropping does not occur. To this end, we estimate $Z = (\mathbb{E}_p[\tilde{w}^{-1}])^{-1}$ by a single Markov chain at the target point in parameter space and ensure that this leads to a compatible estimate, see Supplement.

For MCMC, we use a reweighting procedure [2, 3] which is significantly more involved and uses the relation $F_e = \Delta F_{e\,b} + F_b$. Here, $F_b$ is the free energy at $\kappa_b = 0$ and $\lambda_b = \lambda_e$. The value of $F_b$ can be analytically calculated since for vanishing Hopping parameter $\kappa$:

$$F(\lambda) = -|\Lambda|\,T\,\ln z(\lambda)\,,$$

where $|\Lambda|$ denotes the number of sites of the lattice $\Lambda$ and

$$z(\lambda) = \sqrt{\frac{1-2\lambda}{4\lambda}}\,\exp\left(\frac{(1-2\lambda)^2}{8\lambda}\right)\,K_{\frac{1}{4}}\left(\frac{(1-2\lambda)^2}{8\lambda}\right)\,,$$

with $K_n$ being the Bessel function of the second kind. We prove this relation in the Supplement. The free energy difference $\Delta F_{e\,b} = F_e - F_b = -T\ln\frac{Z_e}{Z_b}$ can be obtained by

$$\mathbb{E}_{p_b}\left[\frac{\exp(-S_e)}{\exp(-S_b)}\right] = \frac{1}{Z_b}\int \mathcal{D}[\phi]\,e^{-S_b(\phi)}\,\frac{e^{-S_e(\phi)}}{e^{-S_b(\phi)}} = \frac{Z_e}{Z_b}\,.$$

We estimate this expectation value with an overrelaxed HMC algorithm [38–41]. In practice, the variance of the estimator will become prohibitively large if the two distributions $p_b$ and $p_e$ do not have sufficient overlap. We therefore choose intermediate distributions $p_{i_1}, \dots p_{i_K}$ ensuring that neighbouring distributions $p_{i_k}$ and $p_{i_{k+1}}$ have sufficient overlap. The free energy difference can then be obtained by

$$\Delta F_{e\,b} = \Delta F_{e,i_K} + \Delta F_{i_K\,i_{K-1}} + \dots + \Delta F_{i_1\,b}\,.$$

In our numerical experiments, we keep $\lambda = 0.022$ fixed and only vary the hopping parameter $\kappa$ of the intermediate distributions $p_i$. We choose a difference in hopping parameter of $\delta\kappa = 0.01$ for $\kappa \in [0.2, 0.3]$ and $\delta\kappa = 0.05$ for all other intermediate hopping parameters $\kappa$. We therefore use $K = 14$ Markov chains with 400k steps each. Thus, a total number of 5.6 million configurations is used for estimation. For a detailed analysis of the dependence of our results on this choice of $\delta\kappa$, we refer to the Supplement.

The error analysis is performed with both the uwerr [42] and Jackknife method which are checked to lead to consistent estimates. We again refer to the Supplement for a more detailed description.

Figure 2 shows that the estimates of both the flow and MCMC are compatible within errorbars. However, the trajectory of the MCMC method has to pass the critical region which is challenging due to critical slowing down. The flow-based estimate can be directly performed at the desired point in parameter space and therefore does not suffer from this problem. This conceptual difference leads to a significantly more precise estimate by the flow-based method. For regions in parameter space which do not require the crossing of a phase transition, MCMC-based methods have errors of comparable order of magnitude (see Supplement). The ability of the flow to perform direct estimates is both of practical as well as of conceptual importance. For example in finite-temperature QCD, one often uses a trajectory whose initial free energy is approximated by the Hadron Resonance model (see for example [2]) leading to an undesirable systematic error. Furthermore, summing up the free energy differences along the trajectory leads to an accumulation of errors. This effect is often the dominant contribution to the error and is particularly pronounced in situations for which the trajectory has to cross a phase transition. Such situations are of great practical relevance, for example in the deconfined phase of $SU(3)$ gauge theory [4, 5]. We stress that both error sources are related since the initial free energy is the starting point of the trajectory.

*Conclusion.* In this letter, we have proposed a method to directly estimate the free energy and hence thermodynamical observables of lattice field theories using deep generative models. This method is of great conceptual appeal as it avoids cumbersome integration through parameter space and does not require an exactly or approximately known integration constant. Future work will focus on scaling this approach to four-dimensional gauge theories. Recent work has successfully constructed flows which are manifestly gauge-invariant [17, 18]. This recent progress, combined with the enormous ongoing advances in deep learning, makes it very promising that our method can be applied to non-abelian gauge theories, and ultimately QCD, in the not too distant future.

*Acknowledgements.* This work was supported in part by the German Ministry for Education and Research (BMBF) under Grants 01IS14013A-E, 01GQ1115, 01GQ0850, 01IS18025A and 01IS18037A. This work is also supported by the Fraunhofer Heinrich-Hertz-Institut (HHI) and by the grant funded by the DFG (EXC 2046/1, Project-ID 390685689). P.S. thanks the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS) for funding. Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Economic Development, Job Creation and Trade. We want to express our gratitude for valuable feedback by Klaus-Robert Müller, Wojciech Samek, Nils Strodthoff, Alessandro Nada, and Stefan Kühn.

[1] W. Busza, K. Rajagopal, and W. Van Der Schee, Heavy ion collisions: the big picture and the big questions, Annual Review of Nuclear and Particle Science **68**, 339 (2018).

[2] O. Philipsen, The qcd equation of state from the lattice, Progress in Particle and Nuclear Physics **70**, 55 (2013).

[3] P. De Forcrand, M. D'Elia, and M. Pepe, 't hooft loop in su(2) yang-mills theory, Physical Review Letters **86**, 1438 (2001).

[4] L. Giusti and M. Pepe, Equation of state of the su (3) yang–mills theory: A precise determination from a moving frame, Physics Letters B **769**, 385 (2017).

[5] M. Caselle, A. Nada, and M. Panero, Qcd thermodynamics from lattice calculations with nonequilibrium methods: The su(3) equation of state, Physical Review D **98**, 054513 (2018).

[6] C. Jarzynski, Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach, Physical Review E **56**, 5018 (1997).

[7] C. Jarzynski, Nonequilibrium equality for free energy differences, Physical Review Letters **78**, 2690 (1997).

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (MIT press, 2016).

[9] P. E. Shanahan, D. Trewartha, and W. Detmold, Machine learning action parameters in lattice quantum chromodynamics, Physical Review D **97**, 094506 (2018).

[10] D. Bachtis, G. Aarts, and B. Lucini, A mapping of distinct phase transitions to a neural network, arXiv preprint arXiv:2007.00355 (2020).

[11] J.-W. Lee, Quantum fields as deep learning, Journal of the Korean Physical Society **76**, 684 (2020).

[12] K. Hashimoto, S. Sugishita, A. Tanaka, and A. Tomiya, Deep learning and the ads/cft correspondence, Physical Review D **98**, 046019 (2018).

[13] K. Hashimoto, S. Sugishita, A. Tanaka, and A. Tomiya, Deep learning and holographic qcd, Physical Review D **98**, 106014 (2018).

[14] K. Hashimoto, Ads/cft correspondence as a deep boltzmann machine, Physical Review D **99**, 106017 (2019).

[15] T. Akutagawa, K. Hashimoto, and T. Sumimoto, Deep learning and ads/qcd, arXiv preprint arXiv:2005.02636 (2020).

[16] 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).

[17] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Equivariant flow-based sampling for lattice gauge theory, Phys. Rev. Lett. **125**, 121601 (2020).

[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, arXiv preprint arXiv:2008.05456 (2020).

[19] J. M. Pawlowski and J. M. Urban, Reducing autocorrelation times in lattice simulations with generative adversarial networks, Machine Learning: Science and Technology **1**, 045011 (2020).

[20] A. Tanaka and A. Tomiya, Towards reduction of autocorrelation in hmc by machine learning, arXiv preprint arXiv:1712.03893 (2017).

[21] P. Wirnsberger, A. J. Ballard, G. Papamakarios, S. Abercrombie, S. Racanière, A. Pritzel, D. J. Rezende, and C. Blundell, Targeted free energy estimation via learned mappings, arXiv preprint arXiv:2002.04913 (2020).

[22] R. W. Zwanzig, High-temperature equation of state by a perturbation method. ii. polar gases, The Journal of Chemical Physics **23**, 1915 (1955).

[23] S. Akiyama, D. Kadoh, Y. Kuramashi, T. Yamashita, and Y. Yoshimura, Tensor renormalization group approach to four-dimensional complex $\phi^4$ theory at finite density, arXiv preprint arXiv:2005.04645 (2020).

[24] M. Asakawa, T. Hatsuda, E. Itou, M. Kitazawa, H. Suzuki, F. Collaboration, *et al.*, Thermodynamics of s u (3) gauge theory from gradient flow on the lattice, Physical Review D **90**, 011501 (2014).

[25] L. Giusti and M. Pepe, Equation of state of a relativistic theory from a moving frame, Physical review letters **113**, 031601 (2014).

[26] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, arXiv preprint arXiv:1912.02762 (2019).

[27] L. Dinh, D. Krueger, and Y. Bengio, Nice: Non-linear independent components estimation, arXiv preprint arXiv:1410.8516 (2014).

[28] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms* (Cambridge university press, 2003).

[29] More precisely, the distributions $p$ and $q$ have to be identically only almost everywhere, i.e. up to a set of measure zero.

[30] D. P. Kingma and M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[31] D. J. Rezende, S. Mohamed, and D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, arXiv preprint arXiv:1401.4082 (2014).

[32] D. J. Rezende and S. Mohamed, Variational inference with normalizing flows, arXiv preprint arXiv:1505.05770 (2015).

[33] T. Müller, B. Mcwilliams, F. Rousselle, M. Gross, and J. Novák, Neural importance sampling, ACM Transactions on Graphics (TOG) **38**, 1 (2019).

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

[35] D. Wu, L. Wang, and P. Zhang, Solving statistical mechanics using variational autoregressive networks, Physical Review Letters **122**, 080602 (2019).

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

[37] K. Nicoli, P. Kessel, N. Strodthoff, W. Samek, K.-R. Müller, and S. Nakajima, Comment on "solving statistical mechanics using vans": Introducing savant-vans enhanced by importance and mcmc sampling, arXiv preprint arXiv:1903.11048 (2019).

[38] S. L. Adler, Over-relaxation method for the monte carlo evaluation of the partition function for multiquadratic actions, Physical Review D **23**, 2901 (1981).

[39] C. Whitmer, Over-relaxation methods for monte carlo simulations of quadratic and multiquadratic actions, Physical Review D **29**, 306 (1984).

[40] D. J. Callaway and A. Rahman, Lattice gauge theory in the microcanonical ensemble, Physical Review D **28**, 1506 (1983).

[41] Z. Fodor and K. Jansen, Overrelaxation algorithm for coupled gauge-higgs systems, Physics Letters B **331**, 119 (1994).

[42] U. Wolff, A. Collaboration, *et al.*, Monte carlo errors with less errors, Computer Physics Communications **156**, 143 (2004).

[43] Y. Burda, R. Grosse, and R. Salakhutdinov, Importance weighted autoencoders, arXiv preprint arXiv:1509.00519 (2015).

[44] S. Nowozin, Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference, ICLR 2018 (2018).

[45] Y. W. Teh, D. Newman, and M. Welling, A collapsed variational bayesian inference algorithm for latent dirichlet allocation, in *Advances in neural information processing systems* (2007) pp. 1353–1360.

[46] C. Gattringer and C. Lang, *Quantum chromodynamics on the lattice: an introductory presentation*, Vol. 788 (Springer Science & Business Media, 2009).

[47] C. M. Bishop, *Pattern recognition and machine learning* (springer, 2006) pp. 468–469.

[48] P. J. Bickel and K. A. Doksum, *Mathematical statistics: basic ideas and selected topics, volume I*, Vol. 117 (CRC Press, 2015).

[49] G. Montavon, G. Orr, and K.-R. Müller, Neural networks-tricks of the trade second edition, Springer, DOI **10**, 978 (2012).

## Conventions for Action

The form of the action $S$ used in the main text is

$$S(\phi) = \sum_{x\in\Lambda} -2\kappa \sum_{\hat{\mu}=1}^{2} \phi(x)\phi(x+\hat{\mu}) + (1-2\lambda)\phi(x)^2 \\ + \lambda\, \phi(x)^4 \,. \tag{11}$$

It can be obtained by starting from the (more standard) action

$$S(\varphi) = \sum_{x\in\Lambda} a^2 \frac{1}{2} \sum_{\hat{\mu}=1}^{2} \frac{(\varphi(x+a\hat{\mu}) - \varphi(x))^2}{a^2} \\ + \frac{m_0^2}{2}\varphi^2(x) + \frac{g_0}{4!}\varphi^4(x) \tag{12}$$

and performing the following re-definitions

$$\varphi = (2\kappa)^{\frac{1}{2}}\phi \,, \tag{13}$$

$$(am_0)^2 = \frac{1-2\lambda}{\kappa} - 4 \,, \tag{14}$$

$$a^2 g_0 = \frac{6\lambda}{\kappa^2} \,. \tag{15}$$

## A brief overview of Deep Learning

*Neural Networks* Neural networks are a machine learning algorithm which has proven to be particularly powerful. A neural network is build of layers which are defined by

$$y^{(l)}(x) = \sigma(W^l x + b^l) \,,$$

where $x \in \mathbb{R}^n$, $y^l \in \mathbb{R}^m$ are input and output of the layer. The output of the layer is also often called the activation of the layer. The weights $W^l \in \mathbb{R}^{m,n}$ and the bias $b^l \in \mathbb{R}^n$ are the learnable parameters of the neural network. The non-linearity $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-linear function which is applied element-wise to the components of $W^l x + b^l$. Widely-used activation function are $\sigma(x) = \max(x, 0)$ or $\sigma(x) = \tanh(x)$.

A neural network consists of $L$ such layers, i.e.

$$g(x) = (y^{(L)} \circ \cdots \circ y^{(1)})(x) \,.$$

It is important to note that the weights $W^l$ and biases $b^l$ do not have to be of the same dimensionality for each layer (although we did not make this explicit in our notation). It is also important to note that we merely described the most simple type of neural network, namely a fully-connected neural network. There is a zoo of other neural networks but we will refrain from a more detailed discussion as it is not needed for our purposes (see [8] for an overview).

*Learning Parameters with Backpropagation* The parameters of the neural networks,

$$\mathcal{W} = \{(W^l,\, b^l)\,, \;\; i = 1, \dots, L\} \,,$$

are determined by minimizing a certain loss function $\mathcal{L}$ by gradient descent (see (7) for the particular loss function used in this work). It is important to emphasize that the number of parameters are typically large (of order $10^3 - 10^6$ for typical modern neural networks). It is therefore clear that one cannot determine the gradient by finite-difference (as we would need calculate the finite difference ratio for each of these parameters which is prohibitively expensive).

The basic idea for calculating the gradient $\nabla_{\mathcal{W}}\mathcal{L}$ is to use the fact that we know the functional form of the neural network: the gradient of the loss is given by

$$\nabla_{\mathcal{W}}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial y^L} \frac{\partial y^L}{\partial y^{L-1}} \cdots \frac{\partial y^1}{\partial x} \,.$$

Each term in this expression is known, for example

$$\frac{\partial y^{l+1}}{\partial y^l} = \sigma'(y^l) W^l \,.$$

For a fixed non-linearity, we know the analytical form of the derivative $\sigma'$. This observation leads to the following algorithm: we first perform a *forward pass* of the neural network, i.e. starting from the input $x$, we calculate the activations $y^l$ for each layer and store them in memory. This process ends with the final activation $y^L$ which is, by definition, the output of the neural network. The gradient of each layer $\frac{\partial y^{l+1}}{\partial y^l} = \sigma'(y^l) W^l$ can then directly be calculated (as we have stored the activation $y^l$). Crucially, we only need the matrix product of these Jacobians and it is efficient to start by calculating the gradient with respect to the output layer $L$, then the layer $L-1$ and so forth. This is because the loss function has a scalar output value and therefore the matrix product of the Jacobians

$$\frac{\partial \mathcal{L}}{\partial y^L} \frac{\partial y^L}{\partial y^{L-1}} \cdots \frac{\partial y^{l+1}}{\partial y^l}$$

is a vector with the same number of components as $y^l$. We can therefore save memory by simply overwriting the stored activation $y^l$. This algorithm is called *backpropagation* and allows us to calculate the gradient $\nabla_{\mathcal{W}}\mathcal{L}$ for roughly the same cost as a forward pass of the neural network.

## Relation between Var($C$) and KL divergence

**Theorem.** *Let $C(\phi) = S(\phi) + \ln q(\phi)$. The following relation between the KL divergence and the variance of $C$ holds:*

$$KL(q_\theta || p) = \tfrac{1}{2}\; Var_q(C) + \mathcal{O}(\mathbb{E}_q[|w-1|^3]) \,,$$

*where* $w(\phi) = \frac{p(\phi)}{q(\phi)}$ *is the normalized importance weight.*

*Proof.* The expectation value of the normalized importance weight is

$$\mathbb{E}_q\left[\frac{p}{q}\right] = \int \mathcal{D}[\phi]\, p(\phi) = 1\,.$$

The Kullback-Leibler divergence can be rewritten in terms of the normalized importance weight

$$\mathrm{KL}(q|p) = \mathbb{E}_q\left[\ln\frac{q}{p}\right] = -\mathbb{E}_q\left[\ln w\right]\,.$$

We now expand the KL divergence around the expectation value of the normalized importance weight

$$\begin{aligned}\mathrm{KL}(q|p) &= -\mathbb{E}_q\left[\ln\left(1-(w-1)\right)\right]\\ &= \mathbb{E}_q\left[\sum_{j=0}^{\infty}\frac{(-1)^j}{j}(w-1)^j\right]\\ &= \underbrace{\mathbb{E}_q[w-1]}_{=0}+\frac{1}{2}\mathbb{E}_q[(w-1)^2]+\mathcal{O}(\mathbb{E}_q[|w-1|^3])\,.\end{aligned}$$

We now relate this expression to the variance of $C$. To this end, we first observe that $C = -\ln\tilde{w}$, where $\tilde{w} = \exp(-S)/q$ is the unnormalized importance weight. We then rewrite the expectation value of $C$ as

$$\begin{aligned}\mathbb{E}_q[C] &= -\mathbb{E}_q[\ln\tilde{w}]\\ &= -\mathbb{E}_q\left[\ln w + \ln Z\right]\\ &= -\ln Z - \mathbb{E}_q[\ln w]\\ &= -\ln Z + \mathrm{KL}(q|p)\\ &= -\ln Z + \mathcal{O}(\mathbb{E}_q[(w-1)^2])\,,\end{aligned}$$

where the last step uses the expansion for the KL divergence derived above. It then follows its variance is given by

$$\begin{aligned}\mathrm{Var}_q(C) &= \mathbb{E}_q\left[(C-\mathbb{E}_q[C])^2\right]\\ &= \mathbb{E}_q\left[(-\ln\tilde{w}+\mathbb{E}_q[\ln\tilde{w}])^2\right]\\ &= \mathbb{E}_q\left[\left(\underbrace{-\ln\tilde{w}+\ln Z}_{=-\ln w}+\mathcal{O}\left(\mathbb{E}_q[(w-1)^2]\right)\right)^2\right]\end{aligned}$$

Expanding the logarithm around $\mathbb{E}_q[w] = 1$ again, we obtain

$$\mathrm{Var}_q[C] = \mathbb{E}_q[(w-1)^2] + \mathcal{O}\left(\mathbb{E}_q[|w-1|^3]\right)\,.$$

Combining this expression with the expansion derived for the KL divergence, we obtain the claim of the theorem. □

The higher-order moments will be small towards the end of the training process for which $q \approx p$ and thus $w \approx 1$. Thus, the variance of $C$ will become small. We indeed observe this behaviour in our numerical experiments, see Figure 3 for an example.

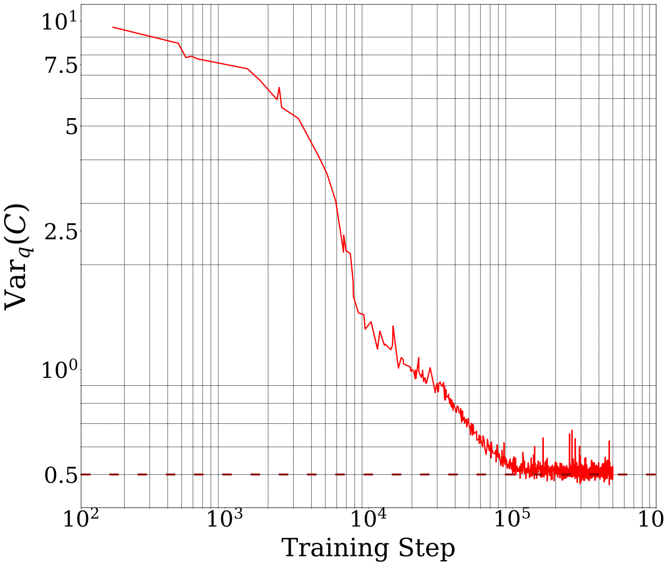

FIG. 3. The variance $\mathrm{Var}_q(C)$ decreases during training. We use hopping parameter $\kappa = 0.3$ and bare coupling $\lambda = 0.022$ for a $16 \times 8$ lattice.

### Analytic Solution for Partition Function

The action of the scalar field theory is given by

$$S = \sum_{x\in\Lambda} -2\kappa\sum_{\hat{\mu}=1}^{2}\varphi(x)\varphi(x+\hat{\mu}) + (1-2\lambda)\varphi(x)^2 + \lambda\,\varphi(x)^4\,.$$

We want to calculate the partion function

$$Z = \int \mathcal{D}[\phi]\,\exp(-S(\phi))\,,$$

which for vanishing hopping parameter $\kappa$ decouples in independent integrals of each lattice site of the lattice $\Lambda$:

$$Z = \prod_{x\in\Lambda}\left(\int \mathrm{d}\phi(x)\,\exp(-\lambda\,\phi(x)^4 - (1-2\lambda)\,\phi(x)^2)\right)$$

The partition function can then be calculated analytically using the integral

$$\int \exp(-ax^4 - bx^2)\,\mathrm{d}x = \sqrt{\frac{b}{4a}}\,\exp\left(\frac{b^2}{8a}\right)\,K_{\frac{1}{4}}\left(\frac{b^2}{8a}\right)\,,$$

where $K_n$ is the modified Bessel function of the second kind. Using this formula, we obtain the following analytic form of the free energy

$$F = -T\,|\Lambda|\,\ln(z)\,,$$

where we have defined

$$z(\lambda) = \sqrt{\frac{1-2\lambda}{4\lambda}}\,\exp\left(\frac{(1-2\lambda)^2}{8\lambda}\right)\,K_{\frac{1}{4}}\left(\frac{(1-2\lambda)^2}{8\lambda}\right)\,.$$

This result corresponds to the zeroth order of the hopping expansion [46] of the partition function $Z$ and one may, in principle, also calculate higher-order corrections. However, they are not needed for our purposes.

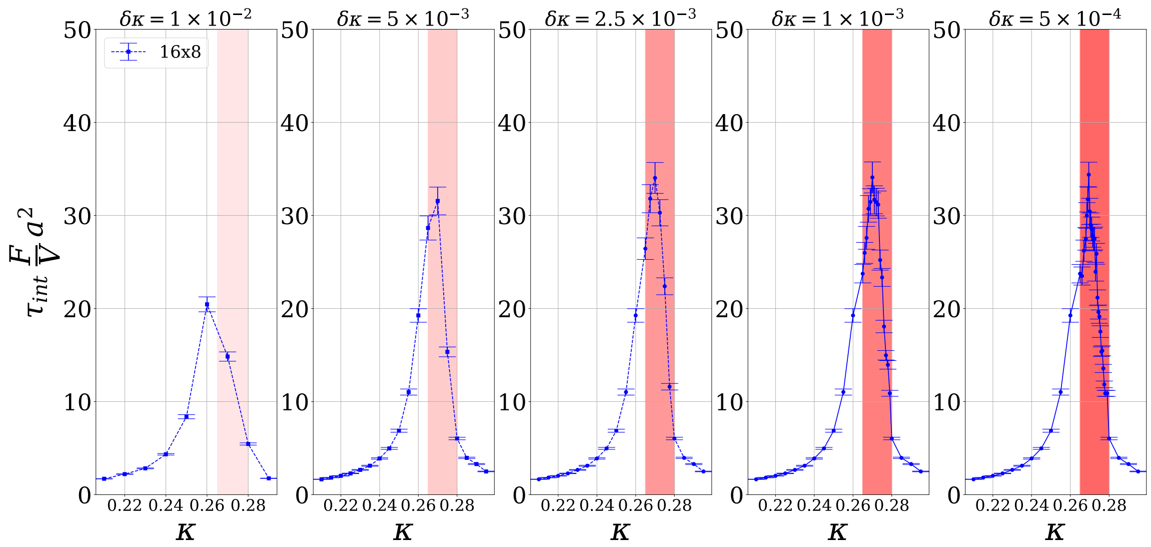

FIG. 4. Integrated autocorrelation time of the free energy during refinement of the step size. The shaded red areas refers to the interval in hopping parameter $\kappa \in [0.265, 0.28]$ for which the refinement is applied. Darker shading indicates narrower step size. The experiments were performed using the overrelaxed HCM algorithm.

### Error Analysis for Free Energy Estimator

Our discussion is based on [44] which discussed the same results in the context of variational inference. We provide a review here since these results may be hard to extract for physicist not familiar with variational inference. We also point out a subtlety that was not discussed in the previous work.

The estimator for the free energy is given by

$$\hat{F} = -T\,\ln \frac{1}{N}\sum_{i=0}^{N} \tilde{w}(\phi_i)\,, \qquad \phi_i \sim q_\theta\,. \tag{16}$$

Theorems for the variance and bias of this estimator are discussed in the following. For this, we use the *delta method of moments* which is summarized in the following theorem.

**Theorem.** *Let $\hat{X}_N = \frac{1}{N}\sum_{i=1}^{N} X_i$ be the sample mean of independent and identically distributed random variables $X_i$ with $\mathbb{E}\left[X_i^{2k+2}\right] < \infty$ for $k \in \{0, 1\}$. Let $h$ be a real-valued function with uniformly bounded derivatives. It then holds that*

$$\mathbb{E}\left[h(\hat{X}_N)\right] = c_0 + \frac{c_1}{N} + \mathcal{O}\left(\frac{1}{N^2}\right)\,,$$

*where*

$$c_0 = h(\mu)\,, \qquad c_1 = h''(\mu)\frac{\sigma^2}{2}\,,$$

*with $\sigma^2 = \mathbb{E}\left[(X - \mathbb{E}X)^2\right]$ and $\mu = \mathbb{E}\left[X\right]$.*

We refer to Chapter 5.3 of [48] for a proof and more details.

The application of the delta method to the free energy estimator $\hat{F}$ is, in practise, subject to a subtlety regarding the bounded differentiablity of the function $h$. We will ignore this subtlety in the following and return to it at the end of the section.

**Theorem.** *The bias of $\hat{F}$ is given by*

$$\mathbb{B}[-\beta\hat{F}] = -\frac{1}{2\,N}\frac{\mathbb{E}_q\left[(\tilde{w} - \mathbb{E}_q[\tilde{w}])^2\right]}{\mathbb{E}_q[\tilde{w}]^2} + \mathcal{O}(N^{-2})\,, \tag{17}$$

*assuming that $\mathbb{E}_q[\tilde{w}^{2k+2}] < \infty$ for $k = 0, 1$.*

*Proof.* The bias of $-\beta\hat{F} = \ln\hat{Z}$ is given by

$$\mathbb{B}[-\beta\hat{F}] = \mathbb{E}_q[\ln\hat{Z}] - \ln Z\,.$$

Using the delta method for moments, we derive that

$$\mathbb{E}_q[\ln\hat{Z}] = \ln Z - \frac{1}{2NZ^2}\mathbb{E}_q\left[(\tilde{w} - \mathbb{E}_q[\tilde{w}])^2\right] + \mathcal{O}\left(N^{-2}\right)\,, \tag{18}$$

where we have used that $h(x) = \ln(x)$ has second derivative $h''(x) = -\frac{1}{x^2}$. The proof then concludes by observing that

$$\mathbb{E}_q\left[\tilde{w}\right] = Z\,.$$

□

**Theorem.** *The variance of $\hat{F}$ is given by*

$$Var\left[-\beta\hat{F}\right] = \frac{1}{N}\frac{\mathbb{E}_q\left(\tilde{w}-\mathbb{E}_q\left[\tilde{w}\right]\right)^2}{(\mathbb{E}_q[\tilde{w}])^2} + \mathcal{O}\left(\frac{1}{N^2}\right), \quad (19)$$

*assuming that $\mathbb{E}_q[\tilde{w}^{2k+2}] < \infty$ for $k = 0, 1$.*

*Proof.* The variance can be written as

$$\mathrm{Var}[-\beta\hat{F}] = \mathbb{E}_q[(\ln\hat{Z})^2] - \mathbb{E}_q[\ln\hat{Z}]^2\,.$$

We now evaluate both terms on the right-hand-side individually using the delta method. For the first term, we use the delta method with $h(x) = (\ln x)^2$ which has second derivative

$$h''(x) = \frac{2}{x^2} - 2\frac{\ln(x)}{x^2}\,.$$

Using this expression, we then obtain that $\mathbb{E}_q[(\ln\hat{Z})^2]$ is equal to

$$(\ln Z)^2 + \left(\frac{1}{Z^2} - \frac{\log Z}{Z^2}\right)\frac{\mathbb{E}_q\left[(\tilde{w}-\mathbb{E}_q[\tilde{w}])^2\right]}{N} + \mathcal{O}(N^{-2})$$

For the squared expectation value, we use the expansion (18) derived in the proof for the bias. This gives that $(\mathbb{E}_q\ln\hat{Z})^2$ is equal to

$$\left(\ln Z - \frac{1}{2NZ^2}\mathbb{E}_q\left[(\tilde{w}-\mathbb{E}_q[\tilde{w}])^2\right] + \mathcal{O}\left(N^{-2}\right)\right)^2$$
$$= (\ln Z)^2 - \frac{1}{NZ^2}\mathbb{E}_q\left[(\tilde{w}-\mathbb{E}_q[\tilde{w}])^2\right] + \mathcal{O}\left(N^{-2}\right)\,.$$

Subtracting these two expressions, it then follows

$$\mathrm{Var}[-\beta\hat{F}] = \frac{1}{Z^2}\frac{\mathbb{E}_q\left[(\tilde{w}-\mathbb{E}_q[\tilde{w}])^2\right]}{N} + \mathcal{O}(N^{-2})\,,$$

and the proof concludes by observing that $Z = \mathbb{E}_q[\tilde{w}]$. □

A few remarks are in order: from the theorems, it follows that the standard deviation of the estimator $\hat{F}$ is of order $\mathcal{O}(1/\sqrt{N})$. In the large $N$ limit, we can therefore neglect the bias correction as it is of order $\mathcal{O}(N^{-1})$. Furthermore, we can replace the expectation values in the theorems by the sample mean up to (negligible) higher-order corrections. In practise, we therefore use these results to estimate the variance and bias of $\hat{F}$. Alternatively, one can use a standard Jackknife analysis to estimate variance and bias (see for example [46]). In our experiments, we use both methods to estimate the errors and check that they lead to consistent results. Lastly, we remark that error estimators for general observables involving the partition function can be derived, see [36].

As mentioned above, the delta method requires that the derivatives of the function $h$ are (uniformly) bounded. For a generic LQFT, this will not be the case for $h(x) = \ln(x)$ since its derivatives diverge for $x \to 0^+$. To the best of our knowledge, the same problem will generically arise in the context of variational inference but seems to have not been discussed in the literature.

To address this subtlety, one could require that the action of the lattice quantum field theory is bounded. For example, this can be ensured by putting the field theory in a box potential. Since only very high energy configurations are affected by this (for suitably large choice of the box potential) and since these configurations are extremely unlikely to be sampled, this modification will have no practical effect on the numerical experiments. After this modification, $\hat{Z}$ is bounded from below and $h^{(n)}(\hat{Z})$ is also bounded as a result.

More rigorously, the result for the variance can be derived without assumptions on a bound for the derivatives by using the *delta method for in law approximation* which takes the following form

**Theorem.** *Let $\hat{X}_N = \frac{1}{N}\sum_{i=1}^{N} X_i$ be the sample mean of independent and identically distributed random variables $X_i$ with $\mathbb{E}\left[X_i^k\right] < \infty$ for $k \in \{1, 2\}$. Let $h$ be a differential function at $\mu = \mathbb{E}_q[X]$. Then*

$$\sqrt{n}\left(h(\hat{X}_N) - h(\mu)\right) \xrightarrow{D} \mathcal{N}(0, \sigma^2(h))\,,$$

*where $\sigma^2(h) = h'(\mu)\,Var(X)$.*

For a proof, we again refer to [48], see Theorem 5.3.3. Applying this theorem to the free energy estimator $-\beta\hat{F} = \ln\hat{Z}$, we obtain the same expression for its variance as derived above. However, the theorem does not require any bound on the derivatives of $h(x) = \ln(x)$.

## Systematic Errors

In this section, we discuss the various sources of systematic error relevant for both the flow-based and MCMC-based estimation methods and discuss how they can be assessed.

*Mode-dropping for Generative Model:* We ensure that no mode-dropping for the generative model takes place, i.e. all modes of the target distribution $p$ are captured by the generative model $q$. Mode-dropping can cause underestimation of the partition function $Z$. To this end, we use the relation

$$\mathbb{E}_{\phi\sim p}[\tilde{w}^{-1}(\phi)] = \frac{1}{Z}\int\mathcal{D}[\phi]\,e^{-S(\phi)}\,\frac{q(\phi)}{e^{-S(\phi)}} = Z^{-1}\,.$$

The left-hand-side can be estimated using a single Markov chain. Note that we sample from the target distribution $p$ as opposed to the generative model $q$. Therefore, the estimator is consistent even if $q$ is mode-dropping. The resulting estimate $\hat{Z}$ for the partition function can then be plugged into (10) to obtain an estimate for the free energy. It is then checked that the

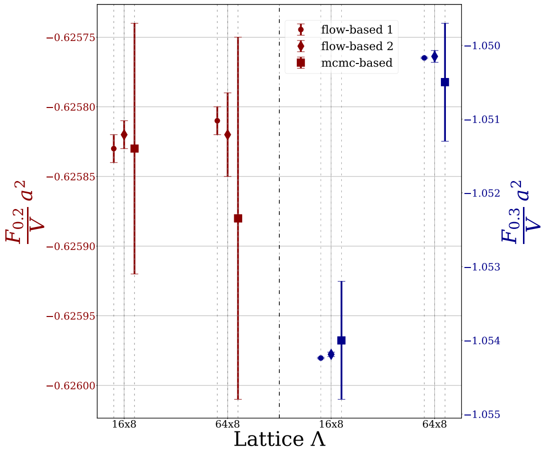

FIG. 5. The left-hand and right-hand sides show the estimates for the free energy at hopping parameters $\kappa = 0.2$ and $\kappa = 0.3$ for fixed bare coupling $\lambda = 0.022$ respectively. We used the same procedure as for Figure 2 on both the smallest and largest lattice for illustration. The first and second flow-based method use as a sampler $p$ and $q$ respectively and lead to compatible results. We use the uwerr method for MCMC error analysis.

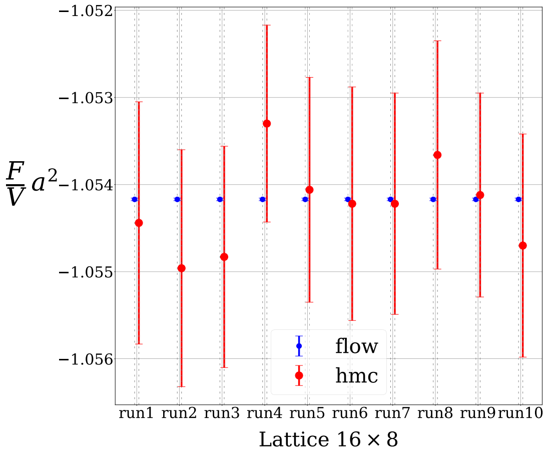

FIG. 6. Free energy estimation as for Figure 2 repeated ten times for the 16 × 8 lattice.

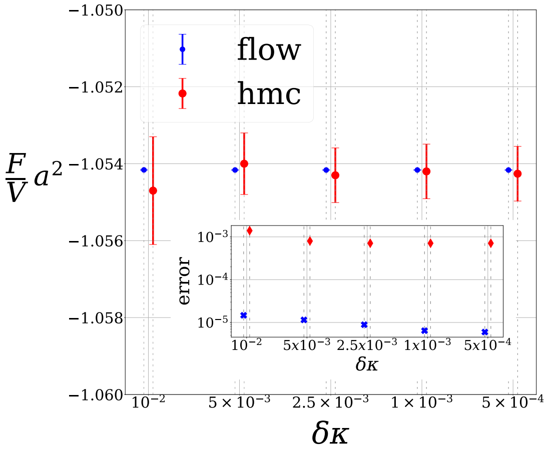

FIG. 7. Free energy at $\kappa = 0.3$ obtained by both MCMC-based and flow-based method. For the MCMC-based method, different step size $\delta\kappa$ of the hopping parameter were used. Details on the step sizes $\delta\kappa$ are summarized in Table I. For the flow-based method, we use the same number of samples as for the corresponding refined MCMC method. As a result, also the error of the flow's estimate decreases.

result is consistent with the one obtained from (9), see Figure 5. We stress that this consistency check only requires a single Markov chain and is therefore equal in cost to the overlap consistency check of the MCMC method discussed below.

*Bias due to imperfect training:* From the theoretical analysis of the variance of the estimator (19), it is expected that it has a standard deviation with the typical $N^{-\frac{1}{2}}$ fall-off in the number of samples $N$. On the other hand, the bias of the estimator (17) due to imperfect training has a subleading $N^{-1}$ fall-off. We check carefully that our error estimates indeed show the theoretically predicted $N^{-\frac{1}{2}}$ behaviour in our numerical experiments.

*Repeated runs:* We repeated the estimate for the smallest lattice ten times. The resulting estimates are shown in Figure 6. The sample standard deviation is consistent with our error estimates. Furthermore, the MCMC-based method does not systematically over- or underestimate with respect to the flow.

*Step size for MCMC:* As explained in the main text, the free energy difference $\Delta F_{e\,b}$ is calculated in steps

$$\Delta F_{e\,b} = \Delta F_{e,i_K} + \Delta F_{i_K\,i_{K-1}} + \cdots + \Delta F_{i_1\,b}\,.$$

In the following, we will analyze the dependency of our results on the chosen steps.

We start from an initial step size corresponding to a change in hopping parameter $\kappa$ of $\delta\kappa = 0.05$. Between $\kappa = 0.2$ and $\kappa = 0.3$, we however take a finer step size of $\delta\kappa = 0.01$. Since we are interested in the free energy difference $\Delta F_{e\,b}$ between $\kappa_b = 0.0$ and $\kappa_e = 0.3$, this corresponds to running $K = 14$ Markov chains. We focus on the 16 × 8 lattice and use an overelaxed HMC algorithm to sample $400k$ configurations for each chain. The overrelaxation is performed every 10 steps.

We then repeatedly refine the step size in a certain subregion around the critical $\kappa$ value. The details can be found in Table I.

The results of this analysis are shown in Figure 7. We observe that the error of the estimator does not significantly decrease. We note that the error of the flow decreases during refinement because its free energy estimation uses the same number of samples as all Markov chains combined (and this number increases by the additional refinement steps).

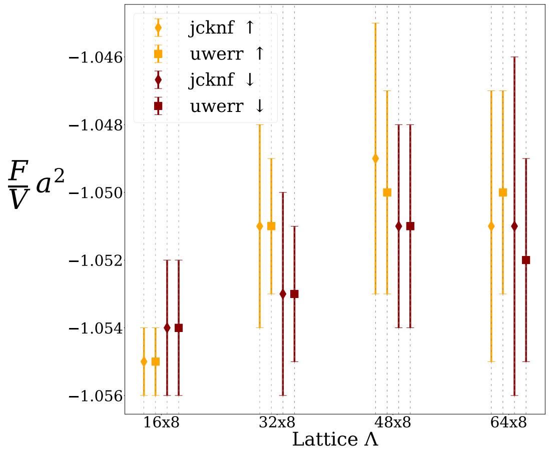

FIG. 8. Free energy at $\kappa = 0.3$ and $\lambda = 0.022$ using both $Z_i/Z_{i+1}$ (down) and $Z_{i+1}/Z_i$ (up). We use the same setup (i.e. number of steps, hopping parameter change $\delta\kappa$, etc) as for Figure 2.

TABLE I. Details on the refinement analysis. In each refinement stage, we take smaller steps $\delta\kappa$ in a certain subregion of the hopping parameter $\kappa$ trajectory (see last column). The step size taken in this region is shown in the first column. Outside of the most refined region, the same step sizes as in the previous refinement stage are taken. Since each chain is taken to be of the same length, the total number of samples (third column) grows proportional to the number of chains (second column).

| $\delta\kappa$ | # chains | # samples | refined $\kappa$ region |
|---|---|---|---|
| 0.01 | 14 | 5.6 M | 0.20-0.30 |
| 0.005 | 24 | 9.6 M | 0.20-0.30 |
| 0.0025 | 40 | 16 M | 0.22-0.30 |
| 0.001 | 76 | 30.4 M | 0.24-0.30 |
| 0.0005 | 88 | 35.2 M | 0.267-0.279 |

*Mode-dropping for MCMC:* In order to ensure that the distributions $p_i$ and $p_{i+1}$ in

$$\mathbb{E}_{p_i}\left[\frac{\exp(-S_{i+1})}{\exp(-S_i)}\right] = \frac{1}{Z_i}\int \mathcal{D}[\phi]\, e^{-S_i(\phi)}\,\frac{e^{-S_{i+1}(\phi)}}{e^{-S_i(\phi)}} = \frac{Z_{i+1}}{Z_i}$$

have sufficient overlap, we also estimate $\frac{Z_i}{Z_{i+1}}$ by exchanging $p_i$ with $p_{i+1}$ in the relation above. We then check that this leads to compatible results, see Figure 8. We note that this consistency check is relatively cheap as it requires running one additional Markov chain. We also study the dependence of the integrated autocorrelation of the free energy on the refinement of $\delta\kappa$, see Figure 4.

## Additional point in parameter space

In the main text, we demonstrated our method in a case where the MCMC-based method had to cross the critical region to calculate the absolute value of the free energy because we consider this one of the main applications of our method. Since this leads to large integrated autocorrelation times for the MCMC-based method, it has errors which are significantly larger than the generative model estimate. In the following however, we will focus on a point in phase space for which the MCMC approach does not need to cross a critical region. Namely, we use the same value for the bare coupling $\lambda = 0.022$ as in the main text but set the hopping parameter $\kappa = 0.2$. As can be seen from Figure 1, this corresponds to a point before the critical regime. The resulting estimates for the free energy are shown in Figure 9. The MCMC-based estimate has a variance of the same order of magnitude as the flow-based one in this regime.

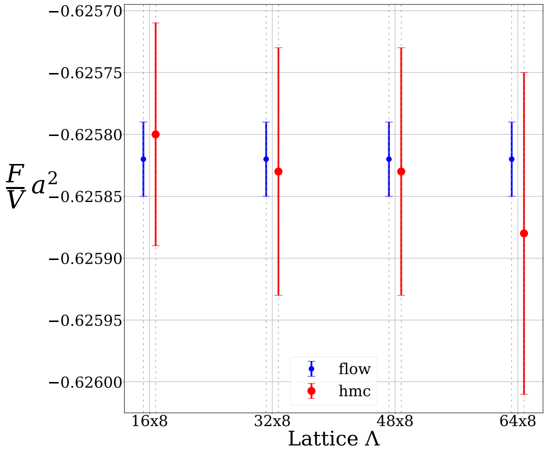

FIG. 9. Free energy estimation using the same setup as for Figure 2 but at $\kappa = 0.2$.

## Details on Numerical Experiments

*HMC:* We use a HMC algorithm with overrelaxation. Each Markov chain has 5k thermalization steps followed by 400k estimation steps. The sign of the field configuration is flipped every ten steps.

*Training of flow:* For every lattice, we use a normalizing flow with six coupling layers. Each coupling layer (5) has neural network $m$ with five fully-connected layers with no bias and Tanh non-linearities. The hidden layers of $m$ consist of 1000 neurons each. For each coupling layer, we split the input in half to obtain $y^{(u)}$ and $y^{(d)}$, see (5), using a checkerboard-type partitioning. Consecutive coupling blocks use alternating checkerboard partioning in order to ensure that all lattice sites are updated. We train the flow for 1M steps using an 8k mini-batch. We use ReduceLROnPlateau learning rate scheduler of PyTorch with an initial learning rate of $5 \times 10^{-4}$ and patience of 3k steps. The minimum learning rate was set to $1 \times 10^{-7}$.

*Estimation:* As described in the main text, for HMC-based estimation we use a step size of $\delta\kappa = 0.01$ for $\kappa \in [0.2, 0.3]$ and a step size of $\delta\kappa = 0.05$ for all other values of the hopping parameter. As a result 14 Markov chains are run. In total, the HMC-based method therefore uses $14 \times 400k = 5.6M$ configurations. We use the same number of samples for the flow-based estimation. For efficiency, we sample these configurations in mini-batches of 3k samples.

*Error estimation:* we use both the uwerr [42] and jackknife method to estimate uncertainties for HMC. In order to deal with autocorrelation for jackknife, we perform binning with a 1k bin size. Error estimation for flow is performed by the delta method and also by jackknife, see Figure 10.

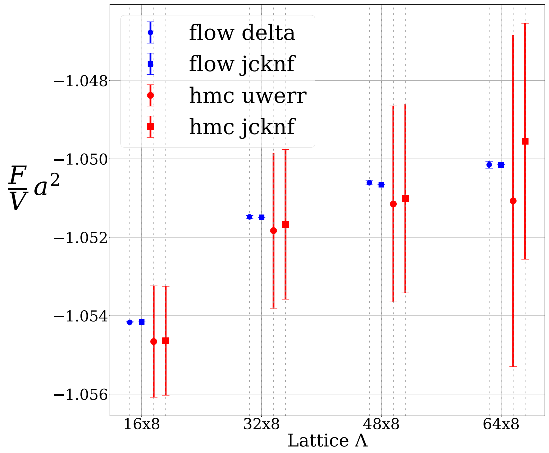

FIG. 10. Free energy estimation with error analysis by both Jackknife and delta method. Both lead to compatible results. We use the same data as in Figure 2.

From the free energy estimates, one can then derive other thermodynamic observables such as the entropy. We refer to the main text for a discussion of this. Figure 11 shows estimation of entropy. Errors were estimated using both the Jackknife and uwerr method. Both error analysis methods lead to consistent results.

*Runtime:* Normalizing Flows allow for very efficient sampling. Specifically, we can sample the total of 5.6M samples in under a minute for all lattices considered in the main text. Furthermore, this sampling procedure can be perfectly parallelized over multiple GPUs as the flow can generate each configuration independently. On the other hand, flows require a substantial up-front training cost which is independent of the number of samples used for free energy estimation and is therefore amortized over the entire estimation procedure. As a result, the relative runtime comparison between the flow-based and HMC-based algorithm strongly depends on the number of samples (as well as, of course, on the used implementation and hardware setup). In our numerical experiments, we use a Intel Xenon 2.4 GHZ CPU with NVidia P100 graphics card with 16GB memory. Both our implementation for the HMC as well as for the generative model were not optimized. The training time of the generative models takes about 20 hours to converge. Generating the samples with 14 different Markov chains takes about 25 hours. It is likely that both the MCMC runtime and training time could be substantially reduced by using more efficient implementations and by tuning the hyperparameters of the training process, such as initial learning rate, its decay schedule, choice of optimizer, initialization of weights, as well as early stopping (see [49] for an overview). Since the goal of this work was to introduce the method and provide a proof of principle, we did not explore such techniques as part of this study.

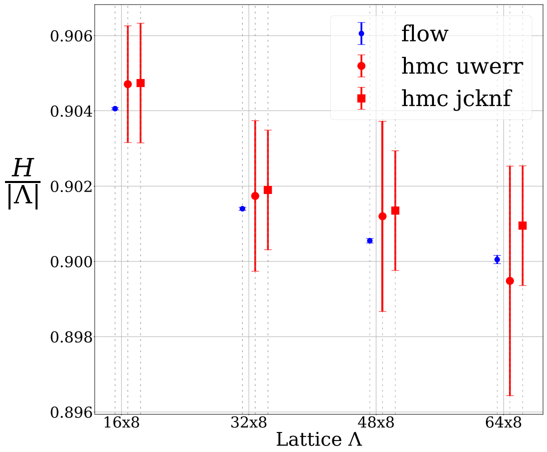

FIG. 11. Entropy density estimation with error analysis by both Jackknife and delta method. Both lead to compatible results. We use the same setup as for Figure 2.