

Stochastic normalizing flows as non-equilibrium transformations

Michele Caselle,^{a,b} Elia Cellini,^{a,b} Alessandro Nada^a and Marco Panero^{a,b}

^a*Department of Physics, University of Turin,
Via Pietro Giuria 1, Turin I-10125, Italy*

^b*INFN, Turin,
Via Pietro Giuria 1, Turin I-10125, Italy*

E-mail: caselle@to.infn.it, elia.cellini@unito.it,
alessandro.nada@unito.it, marco.panero@unito.it

ABSTRACT: Normalizing flows are a class of deep generative models that provide a promising route to sample lattice field theories more efficiently than conventional Monte Carlo simulations. In this work we show that the theoretical framework of stochastic normalizing flows, in which neural-network layers are combined with Monte Carlo updates, is the same that underlies out-of-equilibrium simulations based on Jarzynski's equality, which have been recently deployed to compute free-energy differences in lattice gauge theories. We lay out a strategy to optimize the efficiency of this extended class of generative models and present examples of applications.

KEYWORDS: Algorithms and Theoretical Developments, Lattice QCD, Other Lattice Field Theories

ARXIV EPRINT: [2201.08862](https://arxiv.org/abs/2201.08862)

Contents

1	Introduction	1
2	Jarzynski's equality	3
3	Connection with normalizing flows	6
4	Application in lattice ϕ^4 field theory	10
4.1	Results for stochastic evolutions	13
4.2	Including stochastic layers in normalizing flows	14
4.3	Scaling with the volume	17
4.4	Exploring the parameter space	19
5	Conclusions and future prospects	20
A	Derivation of Jarzynski's equality for Monte Carlo algorithms	23

1 Introduction

The free energy F is a quantity of central relevance in the description of physical, chemical, or biological systems with a very large (possibly infinite) number of degrees of freedom. For a system in thermal equilibrium at a temperature T , its definition¹ through the equality $F = -T \ln Z$ provides a direct connection between the microscopic physics encoded in the sum of states in the partition function Z and the equation of state describing the macroscopic properties of the system, since in the thermodynamic limit the pressure p equals minus the free-energy density per unit volume. In addition to its importance for low-energy physics systems that traditionally fall into the domain of thermodynamics, the free energy also has applications in high-energy physics, including in elementary particle physics and cosmology. For example, the existence and the nature of phase transitions at very high energies may have an impact on various aspects relevant for the evolution of the early Universe [1, 2]. While the Standard Model predicts smooth cross-overs both in the electro-weak- and in the strong-interaction sector [3–6], the existence of first-order phase transitions in various types of extensions of the Standard Model may leave an imprint in gravitational waves [7–9].

In quantum chromodynamics (QCD) the theoretical determination of the free energy at temperatures of the order of the hadronic scale necessarily relies on numerical methods, which are based on Wilson's lattice regularization [10] and amount to estimating *ratios* of Feynman path integrals by Markov-chain Monte Carlo calculations — that is, *differences*

¹Throughout this article we work in natural units: $\hbar = c = k_B = 1$.

in free energies. It is well known, however, that typically this involves significant computational costs: commonly used methods to perform such calculations are based on the numerical integration of a derivative with respect to some parameter [11] or on reweighting the field configurations of a simulated ensemble to a target ensemble, specified by different parameter values [12]. The former of these methods, however, introduces a systematic uncertainty due to the discretization of the integration interval; the latter, on the other hand, is often hampered by the fact that the overlap between the most typical configurations in the simulated and in the target ensemble becomes exceedingly small in the thermodynamic limit. The challenging nature of this problem continues to motivate a search for alternative techniques to estimate the free energies in Monte Carlo lattice QCD [13–21], or, more generally, Feynman path integrals in high-dimensional spaces [22].

A different computational strategy to evaluate free-energy or effective-action differences has been recently pursued in a series of works [23–28] that are based on an exact equality in non-equilibrium statistical mechanics discovered by C. Jarzynski more than 25 years ago [29, 30]. It expresses the free-energy difference between two equilibrium states of a statistical system in terms of the exponential average of the work done on the system to push it out of equilibrium. Jarzynski’s equality is part of a series of works that, during the last decade of the past century, studied in depth the connection between deviations from thermodynamic equilibrium and entropy production [31–36] (and which are reviewed in refs. [37, 38]): it extends and generalizes earlier results [39–41] (for a discussion of the connection between those previous works and Jarzynski’s equality, see refs. [42, 43]), entailing a number of implications, in particular, for the scrambling of quantum information and quantum chaos [44–50], while it reduces to known identities in particular limits. Among the implications of Jarzynski’s equality we mention the fact that, simply using the mathematical properties of convex functions [51], it is possible to derive from it the *inequality* that expresses the second law of thermodynamics. In the context of Markov Chain Monte Carlo simulations, a closely-related implementation of the same idea is Annealed Importance Sampling (AIS) [52, 53], which has seen widespread use in several research fields; we observe that the connection with Jarzynski’s equality was already made explicit in the original AIS paper.

Recently, the explosive growth of machine-learning applications in virtually all fields of human activity has triggered an avalanche of novel implementations of these techniques also in the physical sciences [54] and in elementary particle physics [55–57], broadly extending their previous domains of usage [58–65]. This also includes applications in lattice field theory: as examples of recent works in this area of research, we mention refs. [66–81], but this list is likely to grow much longer in the next few years, as the lattice community is developing approaches that are expected to make machine-learning techniques part of the standard lattice-QCD toolbox [82]. A class of deep generative models called *normalizing flows* [83–85] represents one of the most active and interesting developments in this area of research [86–95]. Normalizing flows can be thought of as an invertible map between a latent (easy) distribution and the target probability distribution, whose complexity is encoded in the trainable parameters of the neural networks that compose the flow itself. The fact that the configurations sampled using this kind of generative models are statistically

independent is a promising feature, as it represents a completely new way to approach a typical problem that plagues Monte Carlo simulations close to the continuum limit, namely the so-called critical slowing down. On the other hand, training times seem to grow very quickly when approaching the continuum limit of the theory of interest [91] so more work is needed to improve the scalability of training procedures. Interestingly, it is crucial to note that normalizing flows do not simply provide a new sampling technique for lattice field theories, but represent also a natural tool for the determination of the partition function Z . Namely, the importance sampling technique introduced in ref. [89] allows for the direct estimation of the free energy at certain values of the theory parameters, and not just of its difference with respect to another point in parameter space, thus representing a conceptual evolution with respect to the aforementioned Monte Carlo free-energy estimation techniques.

Normalizing flows and non-equilibrium Monte Carlo calculations based on Jarzynski's equality are the two main topics that we study in the present work: despite the obvious differences between these two computational approaches, in the following we point out the existence of a direct connection between them, and show how this relation can be fully exploited using a more general class of generative models called *Stochastic Normalizing Flows* [96]. Interestingly, the idea of combining deterministic mappings and out-of-equilibrium transformations used in Jarzynski's equality goes back to ref. [97], albeit for fixed, non-trainable mappings.

The structure of our manuscript is the following: after reviewing the basic aspects of Jarzynski's equality, and discussing its practical use for Monte Carlo integration in section 2, in section 3 we reformulate the equality in a framework suitable both for normalizing flows and stochastic processes, starting from the formulation of ref. [89] and reprising previous work on ref. [96]. Next, in section 4 we present an example of physical application of these flows in the evaluation of the free energy of the ϕ^4 two-dimensional lattice field theory, comparing the effectiveness of different types of flows. Finally, in section 5 we recapitulate our findings and discuss possible future extensions of this work.

2 Jarzynski's equality

Consider a statistical-mechanics system, with degrees of freedom ϕ , whose dynamics depends on a set of parameters collectively denoted as η (which can be the couplings appearing in the Hamiltonian H , etc.). Jarzynski's equality [29, 30] states that the ratio of the partition functions corresponding to equilibrium states of the system corresponding to two different values of η , to be denoted as η_{fin} and η_{in} , is equal to the *exponential average* of the work W , in units of the temperature T , that is done on the system, when, starting from thermodynamic equilibrium with parameters η_{in} , it is driven out of equilibrium by a change of its coupling from η_{in} to η_{fin} during a time interval $[t_{\text{in}}, t_{\text{fin}}]$, according to a protocol $\eta(t)$:

$$\frac{Z_{\eta_{\text{fin}}}}{Z_{\eta_{\text{in}}}} = \overline{\exp(-W/T)}. \quad (2.1)$$

The average (denoted by the bar) appearing on the right-hand side of eq. (2.1) is taken over all possible trajectories in the space of configurations that the system can follow, when

its parameters are modified according to $\eta(t)$. While the starting configurations of these trajectories are equilibrium ones, this is no longer the case for all $t > t_{\text{in}}$, as during each trajectory the system is driven out of equilibrium and never allowed to relax to equilibrium anymore. As a consequence, eq. (2.1) describes a non-trivial relation between equilibrium quantities (on the left-hand side) and non-equilibrium ones (on the right-hand side). The time t on which the parameters η depend can be either real time or Monte Carlo time. Eq. (2.1) can be proven in different ways. In appendix A, we review a “constructive” proof, that is closest to the implementation of Monte Carlo algorithms to compute free-energy differences by evaluating the right-hand side of the equation above. In this case, the time t is discretized and identified with the Monte Carlo time, while the work W can be written as

$$W = \sum_{n=0}^{N-1} \{H_{\eta_{n+1}}[\phi_n] - H_{\eta_n}[\phi_n]\} \quad (2.2)$$

where $\eta_i = \eta(t_i)$.

It may be surprising that the $\eta(t)$ protocol, which describes “how” the parameters of the system are let evolve in time, is fixed and arbitrary, and is not averaged over. As the proof in appendix A shows, the result of the $\overline{\exp(-W/T)}$ average appearing on the right-hand side of eq. (2.1) is independent from η . From the point of view of a Monte Carlo implementation, however, the choice of η has a strong effect on the efficiency with which the algorithm can produce accurate and precise numerical estimates of the $Z_{\eta_{\text{fin}}}/Z_{\eta_{\text{in}}}$ ratio with a finite number of trajectories.

To understand the meaning of eq. (2.1), it is interesting to consider it in two particular limits.

In the limit when the $[t_{\text{in}}, t_{\text{fin}}]$ time interval becomes infinitely long, the η parameters evolve infinitely slowly, so that the configurations remain arbitrarily close to thermodynamic equilibrium along each trajectory. Then, the evolution of the system is dissipationless and the work along each trajectory is equal to the free-energy difference between the final and the initial statistical ensembles. In that case, the right-hand side of eq. (2.1) trivially reduces to $\exp(-\Delta F/T)$.

In the opposite limit, when the switching process from η_{in} to η_{fin} becomes instantaneous, eq. (2.1) can be written as

$$\overline{\exp(-W/T)} = \sum_{\phi_0} \sum_{\phi_1} \pi_{\eta_{\text{in}}}[\phi_0] \exp\left(-\frac{H_{\eta_{\text{fin}}}[\phi_0] - H_{\eta_{\text{in}}}[\phi_0]}{T}\right) P_{\eta_{\text{fin}}}[\phi_0 \rightarrow \phi_1] \quad (2.3)$$

where $\pi_{\eta_{\text{in}}}$ stands for the probability distribution of the system with $\eta = \eta_{\text{in}}$. The sum over ϕ_1 is trivial and one is left with:

$$\overline{\exp(-W/T)} = \sum_{\phi_0} \pi_{\eta_{\text{in}}}[\phi_0] \exp\left(-\frac{H_{\eta_{\text{fin}}}[\phi_0] - H_{\eta_{\text{in}}}[\phi_0]}{T}\right). \quad (2.4)$$

The sum on the right-hand side of eq. (2.4) can be interpreted as an expression for $Z_{\eta_{\text{fin}}}$ (divided by $Z_{\eta_{\text{in}}}$) as a weighted sum over the configurations that contribute to $Z_{\eta_{\text{in}}}$, where the weight of each configuration ϕ_0 is $\exp(-\{H_{\eta_{\text{fin}}}[\phi_0] - H_{\eta_{\text{in}}}[\phi_0]\}/T)$: this means that

in this limit Jarzynski's equality (2.1) simply reduces to the equation describing statistical reweighting [12].

We also wish to point out that Jarzynski's equality (2.1) is closely related to another important result in non-equilibrium statistical mechanics, Crooks' theorem [35, 36]. The latter states that the ratio between the probability density $\mathcal{P}_f(W)$ that a “forward” non-equilibrium transformation cost work W , and the probability density $\mathcal{P}_r(-W)$ that the opposite (“reverse”) transformation cost work $-W$ is given by

$$\frac{\mathcal{P}_f(W)}{\mathcal{P}_r(-W)} = \exp\left(-\frac{\Delta F - W}{T}\right). \quad (2.5)$$

The connection between eq. (2.5) and Jarzynski's equality is obvious, as eq. (2.1) can be obtained by multiplying eq. (2.5) by $\mathcal{P}_r(-W)$ and integrating over W . Note that an interesting implication of Crooks' theorem is that the free-energy difference ΔF is the value of W for which $\mathcal{P}_f(W)$ and $\mathcal{P}_r(-W)$ are equal.

Finally, we note that the theoretical results presented in this section for a statistical mechanics system are immediately translatable in the language of quantum field theory by substituting $H[\phi]/T$ with the Euclidean action $S[\phi]$ and W/T with the generalized work

$$w(\phi_0, \phi_1, \dots, \phi_N) = \sum_{n=0}^{N-1} \{S_{\eta_{n+1}}[\phi_n] - S_{\eta_n}[\phi_n]\} \quad (2.6)$$

$$= S_{\eta_N}[\phi_N] - S_{\eta_0}[\phi_0] - Q(\phi_0, \phi_1, \dots, \phi_N) \quad (2.7)$$

where in the second line we introduced the quantity

$$Q(\phi_0, \phi_1, \dots, \phi_N) = \sum_{n=0}^{N-1} \{S_{\eta_{n+1}}[\phi_{n+1}] - S_{\eta_{n+1}}[\phi_n]\} \quad (2.8)$$

that is the equivalent of the heat exchanged with the environment during the transformation defined by the protocol $\eta(t)$. At the end of the next section we will see how the definition of eq. (2.6) can be generalized in a framework that includes also normalizing flows. Now eq. (2.1) can be written as

$$\begin{aligned} \frac{Z_{\eta_{\text{fin}}}}{Z_{\eta_{\text{in}}}} &= \langle \exp(-w(\phi_0, \phi_1, \dots, \phi_N)) \rangle_f \\ &= \int d\phi_0 d\phi_1 \dots d\phi_N \pi_{\eta_{\text{in}}}(\phi_0) P_f[\phi_0, \phi_1, \dots, \phi_N] \exp(-w(\phi_0, \phi_1, \dots, \phi_N)), \end{aligned} \quad (2.9)$$

where the average over all possible paths $\eta_{\text{in}} \rightarrow \eta_{\text{fin}}$ has been expressed through the probability P_f of going through a given set of configurations $\phi_0 \rightarrow \phi_1 \rightarrow \dots \rightarrow \phi_N$, having used the distribution $\pi_{\eta_{\text{in}}}(\phi_0) = e^{-S_{\eta_{\text{in}}}(\phi_0)}/Z_{\eta_{\text{in}}}$ to sample ϕ_0 . Let us add that it is inside P_f where it lies the dependence of the calculation on crucial details of the transformation, such as the protocol $\eta(t)$ or the Monte Carlo algorithm chosen to update the system in the intermediate steps. In the following, we will refer to a transformation defined by a given protocol $\eta(t)$ as a “stochastic evolution”.

We end this section by pointing out that the same equality can be used to compute the expectation value of a generic observable \mathcal{O} at $\eta = \eta_{\text{fin}}$:

$$\langle \mathcal{O} \rangle_{\eta=\eta_{\text{fin}}} = \frac{\langle \mathcal{O}(\phi_N) \exp(-w(\phi_0, \phi_1, \dots, \phi_N)) \rangle_{\text{f}}}{\langle \exp(-w(\phi_0, \phi_1, \dots, \phi_N)) \rangle_{\text{f}}}, \quad (2.10)$$

whose derivation follows closely the one of eq. (2.9) in the appendix A.

3 Connection with normalizing flows

In this section, we first review the basics about normalizing flows (mostly following the presentation in ref. [96]), before exposing the relation between normalizing flows and non-equilibrium Monte Carlo simulations based on Jarzynski’s equality.

Normalizing flows [83–85, 98] can be interpreted as (a discrete collection of) bijective and differentiable functions interpolating between two different statistical distributions,² and provide a natural tool to construct scalable, arbitrarily complex approximations of unknown posterior distributions in variational-inference problems. Starting from functions that map a base (or “prior”) distribution, which is sufficiently simple to be mathematically tractable, to a target distribution, the density of a statistical sample from the latter can be obtained by constructing its counter-image, and multiplying its density by the product of the Jacobians encoding the volume change along the transformation, i.e. the target distribution is the push-forward of the base distribution. The function from the base distribution to the target distribution can be described as a “generative” map, as it transforms “noise” into the feature-rich, physical target distribution. Conversely, the inverse function is a “normalizing” one, mapping the target distribution into the simpler base distribution.

Normalizing flows can be implemented as neural networks by “discretizing” the functions that interpolate between the base distribution q_0 and the target distribution p through a composition of invertible layers, labeled by a natural number $0 \leq n \leq N$. If z denotes a variable from the base distribution and g_θ is the generative map, one can write:

$$g_\theta(z) = (g_N \circ \dots \circ g_1 \circ g_0)(z). \quad (3.1)$$

We denote the distributions of the intermediate variables $y_{n+1} = g_n(y_n)$ as

$$q_{n+1}(y_{n+1}) = q_n(g_n(y_n)) = q_n(y_n) |\det J_n(y_n)|^{-1}, \quad (3.2)$$

where J_n denotes the Jacobian matrix associated with the change of variables between the layers with labels n and $n + 1$. The training of the network can be done by minimizing the Kullback-Leibler (KL) divergence between the generated distribution and the target distribution [99], which is a measure of the similarity between the two probability distributions and can be written as

$$\tilde{D}_{\text{KL}}(q_N \| p) = \int d\phi q_N(\phi) [\ln q_N(\phi) - \ln p(\phi)]. \quad (3.3)$$

²More precisely, the differentiability must hold at least almost everywhere in the measurable spaces on which the distributions are defined.

In a similar fashion with respect to ref. [89], we introduce a weight function

$$\tilde{w}(\phi) = \frac{\exp(-S[\phi])}{Z_0 q_N(\phi)}, \quad (3.4)$$

where we included the normalization constant of the q_0 distribution in the denominator for reasons that will be clear in the following. The partition function associated with the target probability distribution $p(\phi)$ can be simply expressed as

$$Z = Z_0 \int d\phi q_N(\phi) \tilde{w}(\phi) = Z_0 \langle \tilde{w}(\phi) \rangle_{\phi \sim q_N}, \quad (3.5)$$

where $\langle \dots \rangle_{\phi \sim q_N}$ denotes the average over the ensemble described by the probability density distribution q_N . Then, we are able to write the expectation value of a generic observable \mathcal{O} as

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int d\phi q_N(\phi) \mathcal{O}(\phi) \tilde{w}(\phi) = \frac{\langle \mathcal{O}(\phi) \tilde{w}(\phi) \rangle_{\phi \sim q_N}}{\langle \tilde{w}(\phi) \rangle_{\phi \sim q_N}}. \quad (3.6)$$

Note that the right-hand side of eq. (3.6) expresses the expectation value of \mathcal{O} in the target ensemble through a reweighting from the q_N ensemble, which, in turn, is obtained combining the sampling from the base distribution with a deterministic flow g_θ . In particular, the weight function can be rewritten (as a function of y_0) in the form

$$\begin{aligned} \tilde{w}(y_0) &= \exp(-\{S[g_\theta(y_0)] + \ln Z_0 + \ln q_0[y_0] - Q\}) \\ &= \exp(-\{S[g_\theta(y_0)] - S_0[y_0] - Q\}), \end{aligned} \quad (3.7)$$

where in the second equation we inserted $q_0[y_0] = \exp(-S_0[y_0])/Z_0$. We also defined the quantity Q , which encodes the variation in phase-space volume accumulated along the flow:

$$Q = \sum_{n=0}^{N-1} \ln |\det J_n(y_n)|. \quad (3.8)$$

In practical implementations, Q depends on the network architecture: for example, it is identically zero in frameworks like NICE [100], while it is generally non-zero for networks based on Real NVP [101]. Let us note also that eq. (3.6) is not the only way to compute expectation values with normalizing flows: a popular alternative consists in generating the configurations using $q_N(\phi)$ and applying an independent Metropolis-Hastings algorithm to correct for the difference between $q_N(\phi)$ and $p(\phi)$ [86, 102]. The acceptance rate of the Metropolis step provides a measure of the quality of the flow.³

Finally, we obtain for eq. (3.5) the following form

$$\frac{Z}{Z_0} = \langle \exp(-\{S[g_\theta(y_0)] - S_0[y_0] - Q\}) \rangle_{y_0 \sim q_0}. \quad (3.9)$$

The reader may have noticed the strong similarities between eq. (3.9) and eq. (2.9), in particular when $S_{\eta_{\text{in}}}$ is identified with S_0 : this symmetry is not obvious, as the flows used

³In this work we do not pursue this method, but general considerations on the effectiveness of stochastic normalizing flows presented in the following are valid independently of the formula used to compute expectation values.

in these computations are purely deterministic in the one case and stochastic in the other. The aim of this section is to generalize both concepts under a common framework, that can describe both deterministic and stochastic transformations and eventually use them together in a single flow.

Following the work of ref. [96], we start by considering a configuration y_0 sampled from the base distribution and we define a forward path as a sequence of configurations (y_0, y_1, \dots, y_t) , with $t \leq N$. The probability of going through this path can be expressed as the product of the transition probabilities at all intermediate steps:

$$P_f[y_0, y_1, \dots, y_t] = \prod_{n=0}^{t-1} P[y_n \rightarrow y_{n+1}], \quad (3.10)$$

so that the probability of reaching a given configuration y_t at a generic step t can be expressed by integrating over the initial configuration y_0 (sampled from the base distribution q_0) and over all intermediate configurations:

$$q_t(y_t) = \int dy_0 dy_1 \dots dy_{t-1} q_0(y_0) P_f[y_0, y_1, \dots, y_t]. \quad (3.11)$$

Another useful quantity is the probability of going through the reverse path $(y_t, y_{t-1}, \dots, y_0)$

$$P_r[y_t, y_{t-1}, \dots, y_0] = \prod_{n=0}^{t-1} P[y_{t-n} \rightarrow y_{t-n-1}], \quad (3.12)$$

that allows for the definition of the weight function

$$\tilde{w}(y_0, y_1, \dots, y_N) = \frac{Z}{Z_0} \frac{p(y_N) P_r[y_N, y_{N-1}, \dots, y_0]}{q_0(y_0) P_f[y_0, y_1, \dots, y_N]} \quad (3.13)$$

$$= \exp(-\{S[y_N] - S_0[y_0] - Q\}) \quad (3.14)$$

with

$$Q(y_0, y_1, \dots, y_N) = \ln \frac{P_r[y_N, y_{N-1}, \dots, y_0]}{P_f[y_0, y_1, \dots, y_N]} = \sum_{n=0}^{N-1} (\ln P[y_{n+1} \rightarrow y_n] - \ln P[y_n \rightarrow y_{n+1}]). \quad (3.15)$$

Note that, in our discussion above, the quantity Q defined in eq. (3.8) was a function of y_0 only — a consequence of the deterministic nature of the flow. For flows containing stochastic steps, Q depends on all y_n , for $0 \leq n \leq N$, i.e. is a function of a particular trajectory, not only of its starting (or of its final) point. The KL divergence can then be interpreted as a “distance” between the forward and reverse paths that go through the same configurations,⁴ i.e.

$$\begin{aligned} \tilde{D}_{\text{KL}}(q_0 P_f || p P_r) &= \int dy_0 dy_1 \dots dy_N q_0(y_0) P_f[y_0, y_1, \dots, y_N] \ln \frac{q_0(y_0) P_f[y_0, y_1, \dots, y_N]}{p(y_N) P_r[y_N, y_{N-1}, \dots, y_0]} \\ &= -\langle \ln \tilde{w}(y_0, y_1, \dots, y_N) \rangle_f + \ln \frac{Z}{Z_0} \end{aligned} \quad (3.16)$$

⁴Strictly speaking, eq. (3.16) does not define an actual metric; in particular, it does not necessarily satisfy the triangle inequality.

while the ratio of partition functions simply becomes

$$\frac{Z}{Z_0} = \langle \tilde{w}(y_0, y_1, \dots, y_N) \rangle_f. \quad (3.17)$$

Both normalizing flows and the stochastic procedure described in section 2 emerge naturally within this framework. Normalizing flows are easily recovered by setting

$$P[y_n \rightarrow y_{n+1}] = \delta(y_{n+1} - g_n(y_n)) \quad (3.18)$$

for every n , with $g_n(y_n)$ being the transformation of layer n . In that case, using the fact that

$$q_n(y_n)P[y_n \rightarrow y_{n+1}] = q_{n+1}(y_{n+1})P[y_{n+1} \rightarrow y_n], \quad (3.19)$$

the right-hand side of eq. (3.15) reduces to

$$\sum_{n=0}^{N-1} \ln [q_n(y_n)/q_{n+1}(y_{n+1})] = \sum_{n=0}^{N-1} \ln |\det J_n(y_n)|, \quad (3.20)$$

i.e. to eq. (3.8). Similarly, the quantity $P_f[y_0, y_1, \dots, y_N]$ appearing in eq. (3.16) reduces to a product of Dirac distributions, and after integration this definition of $\tilde{D}_{\text{KL}}(q_0 P_f || p P_f)$ leads to eq. (3.3).

For the stochastic procedure used in Jarzynski's equality, we first introduce the protocol $\eta(t)$ to interpolate between the base distribution and the target distribution. As we remarked above, the $\eta(t)$ function (or its discretization on the layers specified by an integer-valued label) is largely arbitrary, provided it satisfies the requirements of yielding the parameters of the base and target distributions at the initial and final times, respectively. Given a (discretized) protocol η , one can construct the sequence of Boltzmann distributions at the n -th step:

$$\pi_n[\phi] = \pi_{\eta_n}[\phi] = \frac{1}{Z_{\eta_n}} \exp(-S_{\eta_n}[\phi]) \quad (3.21)$$

and construct transition probabilities $P(\phi \rightarrow \phi')$ satisfying detailed balance (a condition analogous to eq. (3.19)):

$$\frac{P[\phi \rightarrow \phi']}{P[\phi' \rightarrow \phi]} = \frac{\pi_{n+1}[\phi']}{\pi_{n+1}[\phi]}. \quad (3.22)$$

Using eq. (3.22), eq. (3.15) can be rewritten in the form

$$Q(y_0, y_1, \dots, y_N) = \sum_{n=0}^{N-1} \ln \frac{\pi_{n+1}[y_n]}{\pi_{n+1}[y_{n+1}]}, \quad (3.23)$$

which, when combined with eq. (3.21) brings us back to eq. (2.8). Similarly, the weight \tilde{w} defined in eq. (3.13) can be rewritten as

$$\tilde{w}(y_0, y_1, \dots, y_N) = \exp\left(-\sum_{n=0}^{N-1} S_{\eta_{n+1}}[y_n] - S_{\eta_n}[y_N]\right) = \exp(-w(y_0, y_1, \dots, y_N)), \quad (3.24)$$

where in the argument of the exponential we have recognized the work w done on the system during the (y_0, y_1, \dots, y_N) trajectory, as defined in eq. (2.6). Thus, eq. (3.17) is just Jarzynski's equality (2.1) in a form that can be easily translated both in the language of (deterministic) normalizing flows and of Markov Chain Monte Carlo simulations.

It is clear now that nothing prevents us from creating a “stochastic” normalizing flow that contains both deterministic coupling layers and stochastic updates. In the following section we review some possible applications and advantages of such a choice.

4 Application in lattice ϕ^4 field theory

We have performed a series of tests in the two-dimensional ϕ^4 interacting field theory defined on a lattice Λ of size $L_t \times L_s$, with lattice spacing a . We denote with $N_t = L_t/a$ and $N_s = L_s/a$ the number of sites in the temporal and spatial directions respectively and we impose periodic boundary conditions along both of them. The Euclidean action of the theory is defined as

$$S(\phi) = \sum_{x \in \Lambda} -2\kappa \sum_{\mu=0,1} \phi(x)\phi(x + \hat{\mu}) + (1 - 2\lambda)\phi(x)^2 + \lambda\phi(x)^4 \quad (4.1)$$

and the (target) probability distribution is

$$p(\phi) = \frac{1}{Z} e^{-S(\phi)}, \quad (4.2)$$

where Z denotes the partition function:

$$Z = \int \prod_{x \in \Lambda} d\phi(x) e^{-S(\phi)}. \quad (4.3)$$

We use three different approaches to generate asymptotically correct configurations: purely stochastic protocols (as described in section 2), standard normalizing flows and stochastic normalizing flows (SNF), in which the affine layers that compose a typical normalizing flow are combined with stochastic layers where Monte Carlo updates are performed.

In each kind of flow, we sample the latent variables z from a normal distribution

$$q_0(z) = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^{|\Lambda|} e^{-S_0(z)} \quad (4.4)$$

with a Gaussian action

$$S_0(z) = \sum_{x \in \Lambda} \frac{z(x)^2}{2\sigma^2}. \quad (4.5)$$

We set $\sigma = 0.5$ so that we exactly recover eq. (4.1) with $\kappa = 0$ and $\lambda = 0$. This simplifies the protocol that is needed for purely stochastic evolutions when interpolating between $q_0(z)$ and $p(\phi)$.

The main observable of interest is the free-energy density of the system $f = F/L_s = F/(aN_s)$. Since

$$F = -T \ln Z = -\frac{1}{N_t a} \ln Z \quad (4.6)$$

we can look at the dimensionless quantity

$$a^2 f = -\frac{\ln Z}{N_t N_s}. \tag{4.7}$$

Recalling the definition of a normalizing flow in eq. (3.1), we use as building blocks of the flow the coupling layers g_i . In order to ensure invertibility and an easy evaluation of the Jacobian, we define the coupling layers by splitting the lattice into two different partitions. A given layer g_i leaves one partition unchanged while acting on the other one. More precisely, we use an even-odd (or “checkerboard”) partitioning, so that each subsequent configuration $y^{i+1} = g_i(y^i)$ can be written as

$$g_i : \begin{cases} y_A^{i+1} = y_A^i \\ y_B^{i+1} = e^{-s(y_A^i)} y_B^i + t(y_A^i) \end{cases} \tag{4.8}$$

so that even sites are left unchanged when A = even and B = odd, and vice versa for odd sites. We observe that s and t are two different neural networks that take as input a configuration and release as output an equally sized configuration. This setup is commonly referred to as an affine layer and is part of the Real NVP architecture [101].

When building a normalizing flow, it is a desirable feature for it to be equivariant under the symmetries of the probability distribution it is going to approximate. In this case, the target probability distribution $p(\phi)$ is invariant under $\phi \rightarrow -\phi$ transformations: to enforce this \mathbb{Z}_2 symmetry in the flow we require the mapping g_θ to be an odd function with respect to z . We do so by choosing neural networks with a hyperbolic-tangent activation function for both s and t in eq. (4.8), and also by taking the absolute value of the output of s . The resulting distribution $q_N(\phi)$ is then invariant under $z \rightarrow -z$ transformations.

The networks used in this work are shallow, i.e. with a single hidden layer between input and output. We obtained results with two types of networks: fully connected networks with $N_s \times N_t$ neurons in the hidden layer and convolutional networks with kernel size 3×3 and one feature map.

The stochastic evolutions described in section 2 can be thought of as a composition of subsequent “layers” as well, whose structure can be written very similarly to that of affine layers. In this case, the n -th layer is defined by the protocol parameters $\eta_n = \eta(t_n)$ that are used to update the system with the action S_{η_n} . Exploiting the locality of the action (4.1), we perform an even-odd partitioning: a stochastic layer acts on an intermediate configuration by updating odd (or even) sites using an algorithm that uses as input only even (or odd) sites, which in turn are kept fixed, in a similar fashion as the affine layer of eq. (4.8). While in this work a highly-efficient heatbath algorithm customized for the target distribution of the ϕ^4 action has been used, we observe that the Metropolis-Hastings algorithm can be used in this approach as well.

Finally, a protocol $\eta(t)$ has to be set in order to interpolate between the initial and the final action, in this case eqs. (4.5) and (4.1) respectively. In practice, one has to gradually change the values of the parameters of the theory to interpolate from the prior distribution ($\kappa = 0$ and $\lambda = 0$) to the target distribution at the desired values of κ and λ . In this

work we always followed a linear protocol in all parameters and for each layer we applied only one heatbath update. We stress, however, that different protocols, such as non-linear ones or with multiple Markov-chain Monte Carlo updates in the same layer, are possible, as well as the possibility to let the intermediate parameters be tunable whenever the flow undergoes a training procedure.

Having fixed the details of affine and stochastic parameters, any flow used in this work is simply characterized by the number of stochastic and deterministic layers: using even-odd partitioning in both cases, we have always used “blocks” of two subsequent layers where both even and odd sites are updated once. In the following we denote the number of affine blocks as n_{ab} and the number of stochastic blocks as n_{sb} . In the case of stochastic normalizing flows, where both $n_{\text{ab}} \neq 0$ and $n_{\text{sb}} \neq 0$, stochastic blocks are always inserted equally distanced between affine blocks so to maximize the number of deterministic layers between them. For example, in the case $n_{\text{ab}} = 2n_{\text{sb}}$, the flow is built alternating two affine blocks and one stochastic block.

In the case of normalizing flows and SNFs, we perform the training procedure needed to tune the parameters of the neural networks contained in the affine layers of eq. (4.8) by minimizing the loss function $-\langle \ln \tilde{w} \rangle_{\text{f}}$, which equals the KL divergence (3.16) minus the ratio Z/Z_0 . In order to evaluate the convergence of the training (i.e. the ability of the latent distribution $q_N(\phi)$ to describe the target distribution $p(\phi)$), we also monitor the variance of the loss

$$\text{Var}_{\text{f}}(\mathcal{L}) = \text{Var}_{\text{f}}(-\ln \tilde{w}), \tag{4.9}$$

and the effective sample size (ESS)

$$\text{ESS} = \frac{\langle \tilde{w} \rangle_{\text{f}}^2}{\langle \tilde{w}^2 \rangle_{\text{f}}} \tag{4.10}$$

which is always in the range $[0, 1]$ and tends to 1 for a “perfect” training. These quantities can be calculated for a stochastic flow as well, using the same definitions, and determine the quality of the protocol chosen for the flow.

A few comments are in order regarding the training of SNFs and the Monte Carlo update algorithm of choice. In this work, the gradients of the tunable parameters of the neural networks, which are computed during the training procedure using the backpropagation algorithm, are propagated also through the stochastic blocks: thus, Monte Carlo updates are performed also during the training. In general, the question whether the graph used for the computation of the gradients is continuous or not through the stochastic blocks can be answered only by looking at the specific update algorithm. The accept-reject step is a non-differentiable function, so the gradients cannot propagate through it: however the graph will depend on the type of proposal used to generate the updated variable. In the case of the heatbath algorithm, the computational graph of the gradients is generated in a non-trivial manner, as every new (accepted) variable is proposed using the values of the variables on the nearest-neighbour sites.

The training procedure for normalizing flows and SNFs was performed by applying 10^4 steps (5×10^4 for fully-connected architectures) of the ADAM algorithm [103] (with

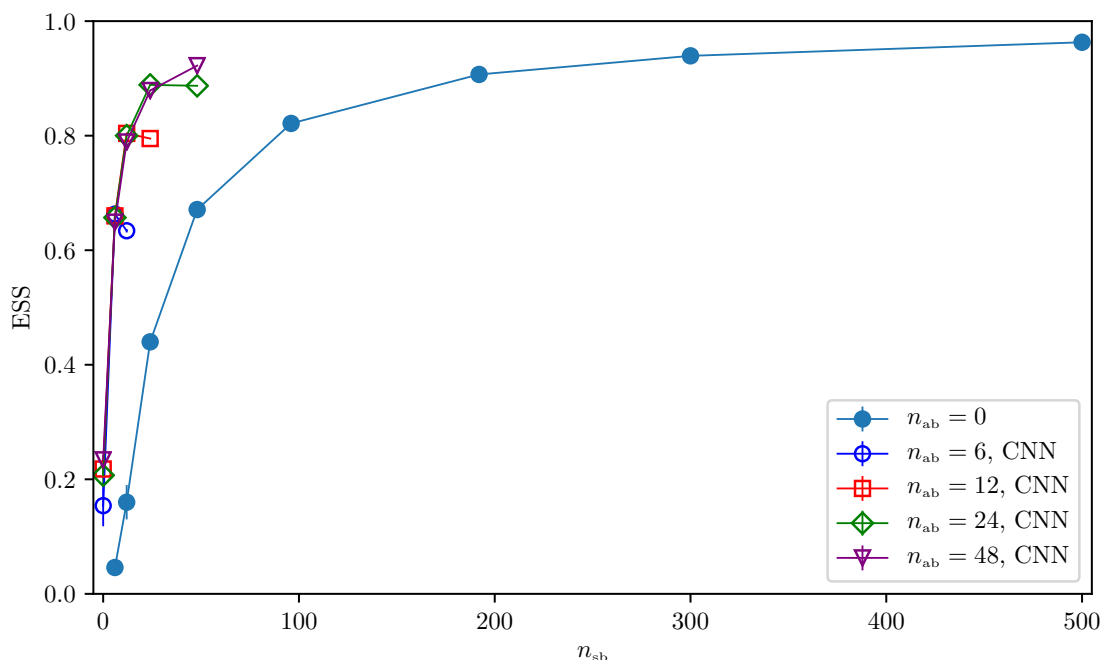


Figure 1. Effective sample size for different flows, for varying number of affine blocks n_{ab} and stochastic blocks n_{sb} , for 16×8 lattices at $\kappa = 0.2$, $\lambda = 0.022$. The $n_{ab} = 0$ points are purely stochastic evolutions, where no training is required, while the $n_{sb} = 0$ data represent standard normalizing flows. The remaining points in the plot represent SNFs, where stochastic blocks are placed between affine blocks. Error bars are not visible for most of the flows and stochastic evolutions under consideration due to the very small statistical errors.

batches of 8000 configurations) to update the parameters of the neural networks; the code used for the training is based on the PyTorch library. We used the ReduceLROnPlateau scheduler with an initial learning rate set to 0.0005 and a patience of 500 steps. All numerical results for the free energy density $a^2 f$ defined in eq. (4.7) are obtained always taking $N_{\text{meas}} = 2 \times 10^5$ independent measurements for the average of eq. (3.17), for any of the three methods used in this work (normalizing flows, SNFs and stochastic evolutions); at the same time also the ESS is calculated using eq. (4.10). The errors on these two quantities have been computed using a jackknife procedure. All the numerical experiments discussed in this section (both training procedures and measurements) have been performed on a NVIDIA Volta V100 GPU with 16GB of memory.

4.1 Results for stochastic evolutions

Let us first describe our results for the free-energy density with stochastic evolutions, obtained by computing the average of eq. (3.24) with N_{meas} independent measurements. We stress that in this case no training is strictly needed, as we already fixed all the parameters by choosing a linear protocol.

In figure 1 we report some values obtained for the ESS (4.10) by performing measurements for different flows. The only difference between the various protocols is the number

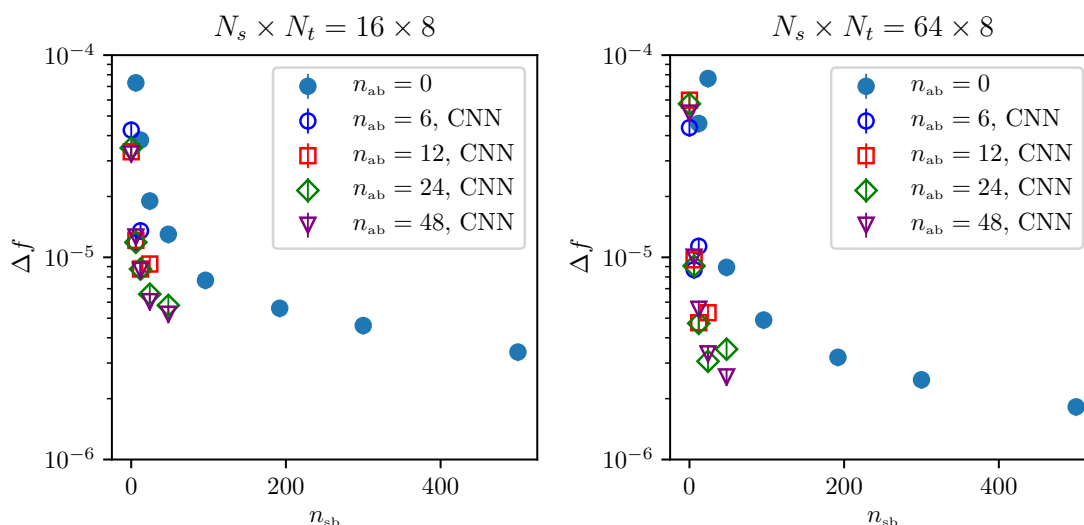


Figure 2. Error on the free energy density $a^2 f$ for various flow architectures from 2×10^5 independent measurements. Results for 16×8 lattices (left-hand-side panel) and 64×8 lattices (right-hand-side panel) at $\kappa = 0.2$, $\lambda = 0.022$. The error on the error is calculated with a jackknife procedure.

of intermediate steps (or, equivalently, the number of stochastic blocks n_{sb}) between the initial and final points. An effective strategy with this kind of out-of-equilibrium transformations is to increase the number of intermediate Monte Carlo updates performed during the transformation while keeping the protocol $\eta(t)$ and the number of measurements N_{meas} fixed. In this way, each measurement becomes more expensive from a computational point of view, but the distribution $q_N(\phi)$ is also more effective at describing the target distribution $p(\phi)$, as shown by the very high values of ESS obtained for the largest values of n_{sb} and by the steadily decreasing error in both panels of figure 2.

To determine the optimal number of stochastic blocks at fixed computational effort, we combined the error on the free-energy density f and an estimate of the computational cost of each measurement. In this case the latter is simply given by n_{sb} , as just a single Monte Carlo update is performed in each layer.⁵ The results of this comparison are shown in figure 3: one can observe that on the smaller lattice increasing the number of intermediate steps above 100 does not appear to be particularly cost-effective in sampling the target distribution, while it is still slightly advantageous for the larger volume.

4.2 Including stochastic layers in normalizing flows

Let us proceed to the analysis of results for stochastic normalizing flows, where stochastic blocks are inserted between “deterministic” affine blocks. Firstly, we observe that for standard normalizing flows, a larger n_{ab} does not necessarily provide a more efficient way

⁵Increasing the number of updates in each layer is possible, but we found it not to be helpful in this setup.

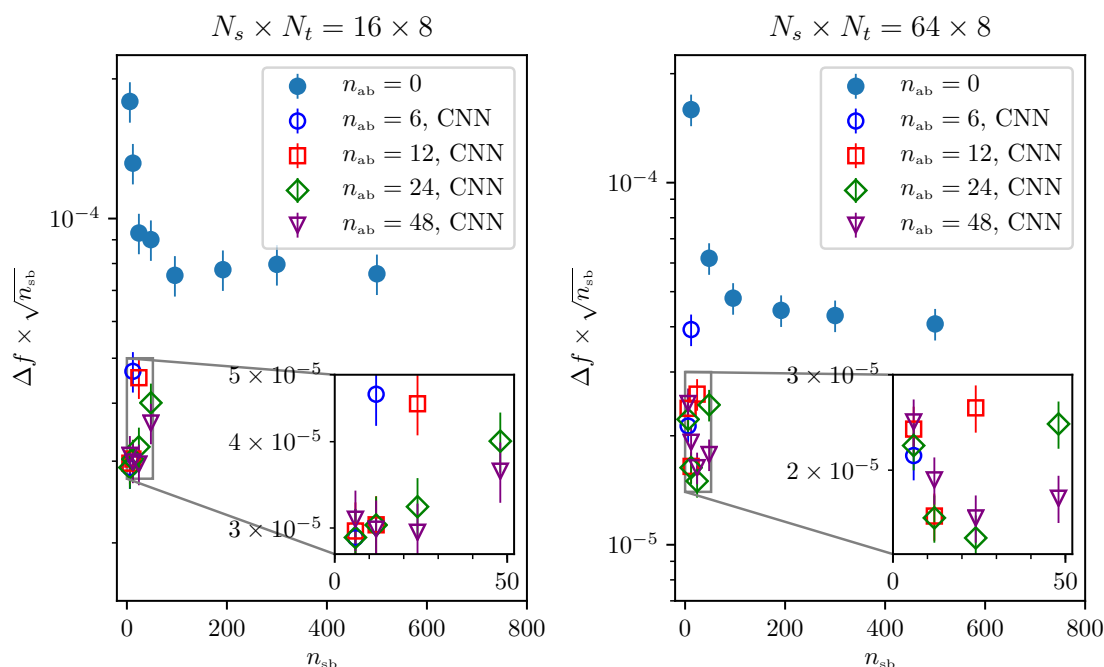


Figure 3. Efficiency of various SNF architectures, determined as the error Δf times the square root of the number of stochastic blocks, for 16×8 lattices (left-hand-side panel) and 64×8 lattices (right-hand-side panel). Training time and measurements cost for the deterministic layers in SNFs are not taken into account.

of sampling $p(\phi)$, possibly because of a more difficult training. This occurs both for convolutional and fully connected neural networks, as shown in figure 4 for the data at $n_{sb} = 0$: we also add that with the shallow representations used in this setup, the latter perform much better than the former.

Whenever stochastic layers are inserted between affine layers, the effectiveness of the flows at fixed number of training steps⁶ improves in a decisive manner for flows based on convolutional neural networks (CNN), but only slightly for fully connected networks, so that the latter are quickly surpassed for $n_{ab} > 6$ by the convolutional architectures. Interestingly, in both cases flows with larger n_{ab} seem to perform better when also n_{sb} grows, until a plateau is reached. For fully connected networks this plateau is reached very quickly, as the improvement for $n_{ab} > 6$ is very small. On the other hand, for CNNs the performance keeps improving even more when n_{ab} increases, especially for larger volumes: it is also interesting to note that flows with $n_{ab} = n_{sb}$ seem to be more efficient than flows with $n_{ab} \neq n_{sb}$. Looking at the two insets of figure 3, this behaviour is mostly absent for the 16×8 lattice, but it can be observed for the largest one (64×8), at least for $0 < n_{ab} < 48$.

The difference in behavior for the two types of neural networks is even more striking when the spatial size aN_s is increased, as shown in figure 5. Fully connected networks show

⁶We note that increasing n_{sb} also increases the time required for each training step.

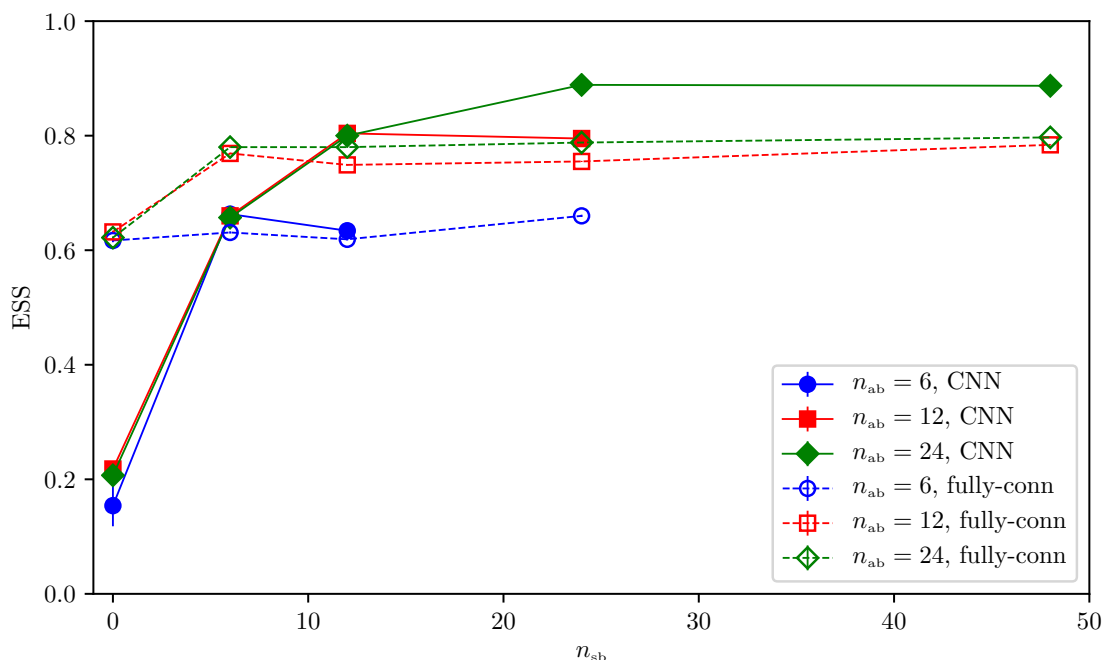


Figure 4. Comparison of ESS for stochastic normalizing flows, with fully connected neural networks (empty symbols) and with CNN (full symbols), for varying number of affine blocks n_{ab} and stochastic blocks n_{sb} . Results obtained for 16×8 lattices at $\kappa = 0.2$, $\lambda = 0.022$. Training duration is 10^4 epochs for SNFs with CNN and 5×10^4 for SNFs with fully connected networks.

little to no improvement when more stochastic blocks are inserted, while CNN-based flows improve with increasing n_{sb} at a similar rate even for larger volumes.

We stress that the saturation effect that we observe by increasing n_{ab} for SNFs with fully-connected networks in figures 4 and 5 is not an artifact of training procedures cut too short: we checked that all the combinations of n_{ab} and n_{sb} under investigation were well in the slowly improving regime and that the ESS had already reached a plateau, both for fully-connected networks and CNNs. As a further check, we performed longer, independent training procedures for the smallest and largest volumes for SNFs with fully-connected networks with $n_{ab} = 24$ and $n_{sb} = 6, 12, 24, 48$: for a training twice as long (i.e. after 10^5 epochs) we obtained values for the ESS around 1% better than those from shorter trainings.

The neural networks s and t of eq. (4.8) used in this work can be considered shallow, since a single hidden layer is present. However, in this setup the overall architectures are “deep” in the sense that the number of affine blocks is relatively large ($n_{ab} \geq 6$). What we found through careful experimentation is that flows with large n_{ab} appear to be the easiest to integrate whenever many stochastic layers are inserted. For example, flows with deeper neural networks or less affine blocks were generally less amenable to improvements in training when trained with many stochastic layers.

At this point we would like to use a word of caution concerning the generality of the results obtained in this work. A complete discussion would take into account the

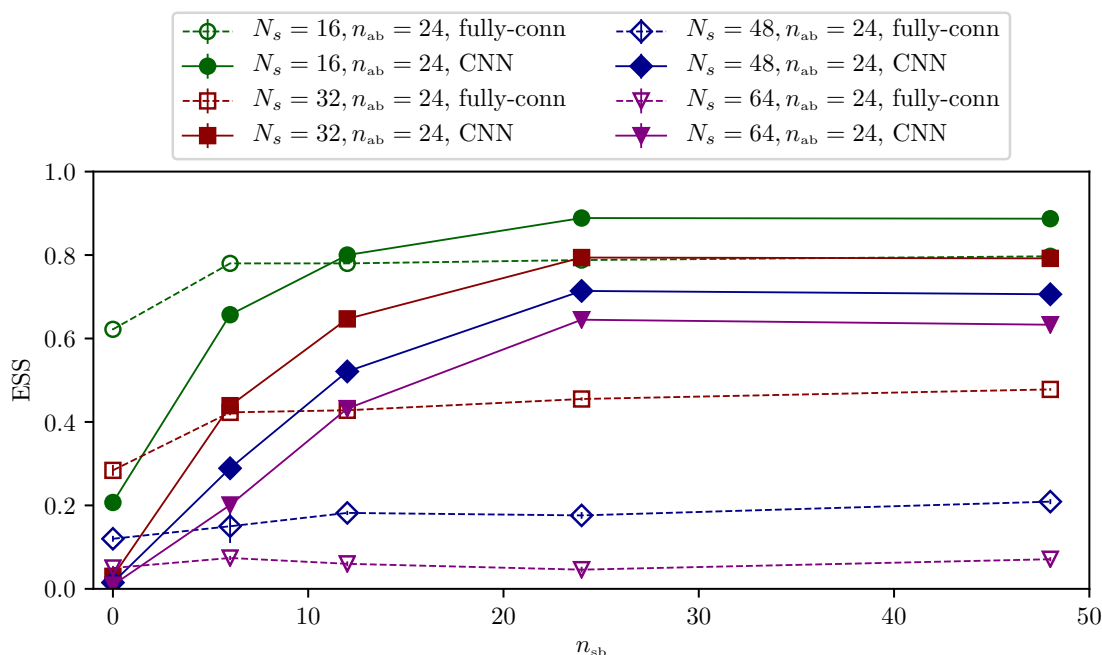


Figure 5. Comparison of effective sample size for $N_s \times 8$ lattices between SNFs with fully connected networks (empty points) and CNNs (full points), at $\kappa = 0.2$, $\lambda = 0.022$. Training duration is the same as in figure 4.

dependence on a wide variety of factors: the architecture of affine layers (e.g. the type of neural networks), the characteristics of stochastic layers (e.g. their position with respect to affine layers or the Monte Carlo algorithm of choice) and the tuning of the hyperparameters. Hardware, too, plays a role: GPUs allow for greater parallelization but, for example, the dependence on the batch size is not obvious when both Markov-chain updates and forward passes on affine coupling layers are performed. For these reasons, a full quantitative comparison between normalizing flows (or SNFs) and purely stochastic evolutions, while extremely interesting for future practical applications, is beyond the scope of this work. Furthermore, including the training time in the overall effort to reach a given error on f is not completely straightforward. Only neural networks strictly need a training procedure to work properly, while for purely stochastic evolutions a reasonably efficient protocol can be set manually. We limit ourselves to observe that in the standard and stochastic normalizing flows used in this work the training time was of the order of hours. This is not negligible at all when compared with the time needed to perform $N_{\text{meas}} = 2 \times 10^5$ measurements, which ranges from seconds for flows containing zero or few stochastic layers, to at most a few minutes for flows with $n_{sb} > 100$.

4.3 Scaling with the volume

It is interesting to investigate the effects of an increase in the spatial volume of the lattice: heuristically, the target distribution becomes more sharply peaked and thus a better train-

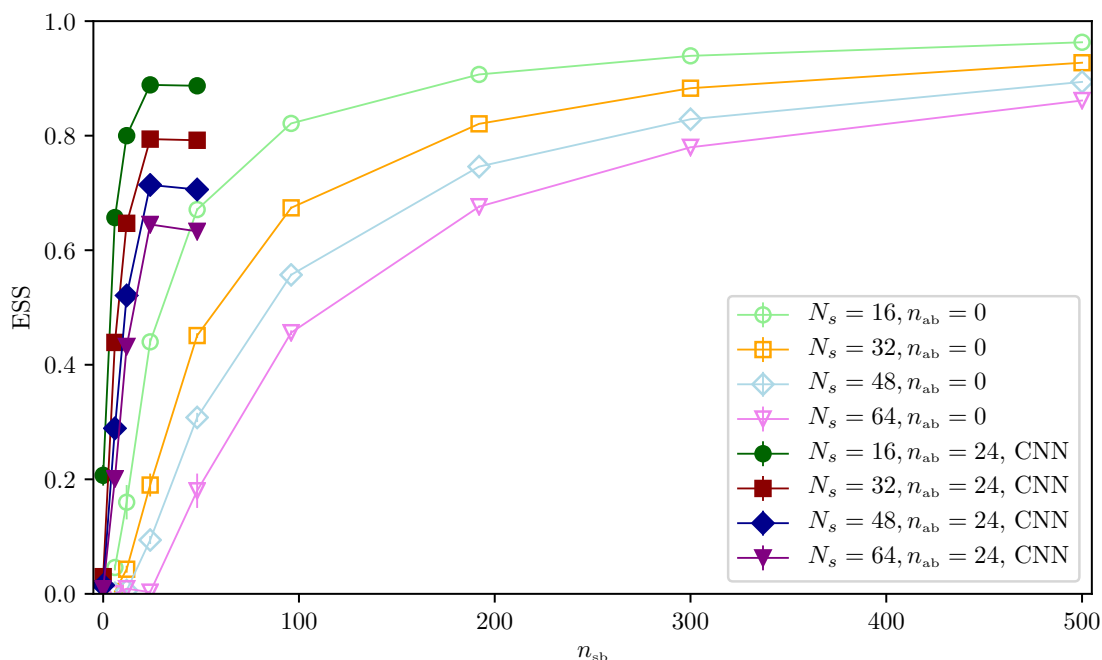


Figure 6. Comparison of effective sample size for $N_t = 8$ lattices between purely stochastic evolutions (light colors) and SNFs with CNNs and $n_{ab} = 24$ (dark colors) as a function of n_{sb} , at $\kappa = 0.2$, $\lambda = 0.022$.

ing and/or a more expressive flow are needed in order to reach the same effectiveness in sampling $p(\phi)$. Before looking at the results, we remind that the number of training steps for SNFs with CNNs is equal to 10^4 for all the volumes investigated in this work.

For what concerns purely stochastic evolutions, a simple strategy allowing to sample larger volumes as effectively as smaller ones consists in increasing the number of Monte Carlo updates of the protocol. Intuitively, the corresponding transformation becomes closer and closer to a reversible one, i.e. a transformation in which every intermediate point is (almost) at equilibrium. This can be seen in figure 6, where for all volumes the effective sample size grows with n_{sb} .

The question is whether this strategy can be implemented structurally for stochastic normalizing flows as well. As we already pointed out, an improvement in the effectiveness of SNFs can be obtained simply adding deterministic and stochastic blocks in an equal manner, roughly keeping $n_{ab} = n_{sb}$ and alternating one block of even-odd affine layers with one block of even-odd Monte Carlo updates. This can be easily seen for all the volumes under study by looking at figure 6: for $n_{sb} = 0$ the ESS is essentially zero except for the smallest volume, but for $n_{sb} > 0$ it grows quickly for all values of N_s (with N_t left fixed) until reaching the point $n_{ab} = n_{sb} = 24$; after that, no improvement is observed if n_{sb} increases. In order to further improve a flow then, n_{ab} must be increased as well: when $n_{ab} = n_{sb}$ the value of the ESS for all volumes steadily increases, as shown in figure 7.

We would like to note again that these results are obtained when the training of all architectures under consideration was already well inside the slowly-improving regime.

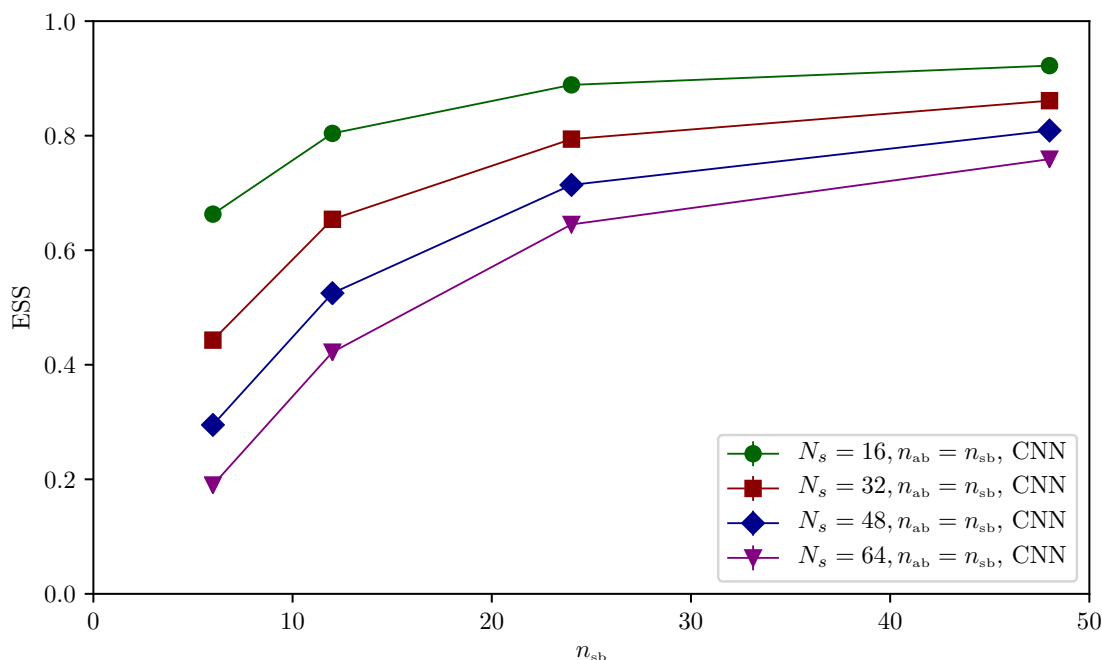


Figure 7. ESS comparison for $N_s \times 8$ lattices between CNN-based SNFs with $n_{sb} = n_{ab}$, at $\kappa = 0.2$, $\lambda = 0.022$.

Even increasing the training duration by a factor 5 (i.e. to 5×10^4 epochs) the improvement in the quality of the training is absent (for smaller values of n_{ab}) or very limited (around 1% increase in the ESS for $n_{ab} = 48$) both for the smallest and largest volumes under study: most importantly, the general behaviour previously observed when increasing n_{sb} and n_{ab} still holds.

Naively, larger lattices allow for smaller errors, as averaging over bigger volumes is akin to having increased statistics. However, as we remarked above, the target distribution $p(\phi)$ is more difficult to sample on larger volumes, so it is interesting to investigate the effort required to obtain the same effectiveness when changing the lattice spatial size. In figure 8 we study the error reduction with respect to $N_s = 16$, when increasing n_{sb} either for stochastic evolutions, or for SNFs with $n_{ab} = 24$ affine blocks. In the first case, the naive error reduction is reached around $n_{sb} = 300$ for $N_s = 32$ and 48, while for $N_s = 64$ a protocol with $n_{sb} = 500$ might not be sufficient yet. For SNFs the situation is rather different though, as the expected gain is already reached for $n_{sb} = 24$, which, as discussed above, is the most efficient setup for $n_{ab} = 24$. This could indicate that for these flows the increase in volume can be compensated more easily than by simply increasing n_{sb} in purely stochastic evolutions.

4.4 Exploring the parameter space

In order to consolidate the results obtained for $\kappa = 0.2$ and $\lambda = 0.022$, we explore further the unbroken symmetry phase of the model by changing both target parameters. The trends

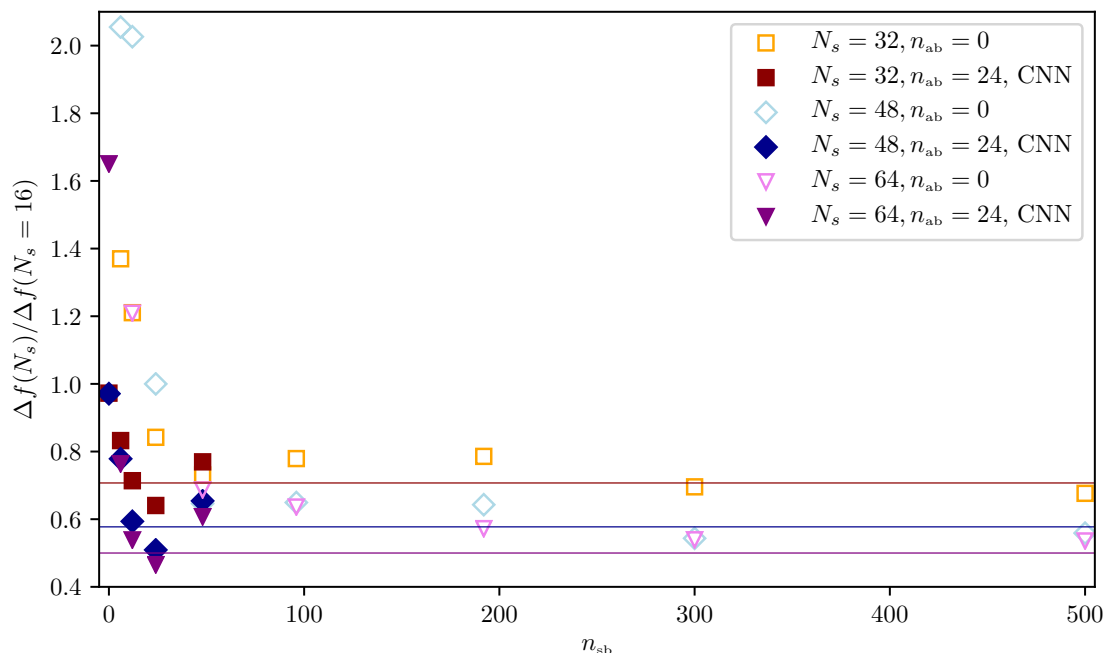


Figure 8. Error Δf of the free-energy density for $N_s \times 8$ lattices normalized to the error for a 16×8 lattice, for different flows at $\kappa = 0.2$, $\lambda = 0.022$ and with $N_{\text{meas}} = 2 \times 10^5$. Horizontal lines show the error decrease that is expected considering only the volume averaging.

that we observe in figure 9 are fully compatible with those discussed previously in this section, namely the saturation of the ESS whenever $n_{sb} > n_{ab}$ and the clear improvement in the quality of the flows if both the number of affine and stochastic blocks are increased while being kept equal.

A word of caution is in order concerning the extrapolation of these results in any point of the parameter space, in particular in the vicinity of the transition to the broken symmetry phase. A different kind of SNF architectures might be needed to reproduce the same trends observed in this section.

5 Conclusions and future prospects

In this manuscript we have made explicit the connection between normalizing flows and non-equilibrium Monte Carlo calculations based on Jarzynski’s equality (“stochastic evolutions”), which becomes apparent when both are used for the determination of the free energy F . In general, stochastic evolutions by themselves represent an efficient method to sample a target distribution and to directly evaluate the partition function Z : they provide a novel framework to compute expectation values in lattice field theory from first principles and represent a viable alternative to the traditional Monte Carlo approach. In Markov-chain Monte Carlo simulations, measurements are performed on subsequent equilibrium configurations that are part of the same Markov chain, thermalized according to the target

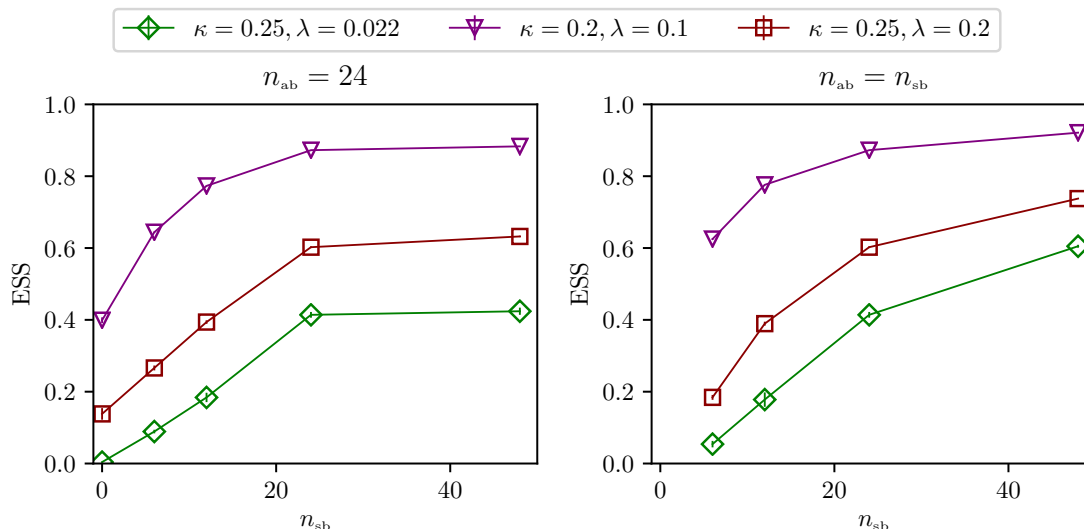


Figure 9. ESS comparison for 16×8 lattices between CNN-based SNFs with $n_{sb} = 24$ (left panel) and $n_{sb} = n_{ab}$ (right panel), for different values of the target parameters.

coupling(s). The cost of generating a new configuration is as low as the application of a single update on a lattice. However, configurations belonging to the same Markov chain are correlated with each other, reducing the number of effectively independent configurations, and this problem can become potentially severe in the proximity of critical points; in addition, every new set of couplings requires a new chain to be thermalized again. By contrast, in stochastic evolutions, each measurement is *independent* from the others, but involves several Monte Carlo updates that “interpolate” from the prior distribution to the distribution defined by the target coupling(s). The computational cost of this might seem very high at first, due to the relatively large number of Monte Carlo updates needed for convergence. However, it is crucial to note that, as a byproduct of this procedure, all intermediate couplings can be sampled at the same time. It is then clear that this method is particularly suitable when a fine scan of the parameter space of a theory is required. A typical example is the study of thermodynamics in strongly interacting quantum field theories. Indeed, a full-scale numerical study using this approach has already been reported: it is the high-precision computation of the equation of state in the $SU(3)$ Yang-Mills theory in $3 + 1$ dimensions [27]. In that case, the only parameter changed by the protocol is the inverse coupling β , which in turn also controls the temperature of the system. The major difference with respect to the stochastic evolutions used in this work is that the generator of starting configurations was not a treatable (e.g. normal) prior distribution, but a distribution obtained from a Markov chain thermalized at a certain value β_0 of the inverse coupling.

We also showed that a common framework can describe in a natural way both stochastic and normalizing flows, following the work of ref. [96] and expanding it so to explicitly include stochastic transformations based on Jarzynski’s equality. The construction of stochastic normalizing flows is a direct consequence of this connection: a “hybrid” flow

combining deterministic and stochastic transformations proved to be highly efficient at sampling the target distribution, with relatively simple and short training procedures and a limited number of Monte Carlo updates. Our previous work on stochastic evolutions suggested that increasing the number of intermediate steps, while expensive per se, is an efficient way of improving the effectiveness of the flow, and this is confirmed by the numerical tests performed in this work. However, it was not clear whether this strategy would work at all for stochastic normalizing flows. Interestingly, this proved to be true also in the latter case and with surprisingly high efficiency. We stress though, that this was observed in a rather specific setup in which a) affine blocks based on CNNs are used, b) stochastic and affine blocks are placed in an alternating order, and c) the number of stochastic and affine blocks is increased roughly in the same manner.

The strong similarities between purely stochastic evolutions and SNFs suggest that the success of the former in full-scale numerical calculations can be replicated with the latter using similar strategies. Moreover, due to the existence of an explicit protocol determined by the stochastic updates, the role of intermediate affine layers can be interpreted straightforwardly. As pointed out above, each stochastic layer can be used to sample the intermediate parameters defined by the protocol $\eta(t)$: the (deterministic) affine layers inserted between them are then trained to “glue” together the various steps of the protocol without resorting to further (generally more expensive) intermediate Monte Carlo updates.

Before concluding, we would like to point out that the training of SNFs is not necessarily more efficient in general. A standard normalizing flow is intuitively more “free” to seek the best possible path between the prior and the target distributions. On the other hand, stochastic normalizing flows are “constrained” by the protocol chosen for Monte Carlo updates; these intermediate steps happen at fixed values in the parameter space of the theory, which the training is “forced” to go through. Naively, one could expect that a standard normalizing flow will *eventually* outperform a stochastic one, given the same neural-network architecture; however, this might not happen in a reasonable training time, and a fixed protocol might lead in some instances to a faster training. More work is needed, for example, to understand how the training times needed by SNFs to reach a plateau in the loss function behave when changing the volume of the system.

Among possible directions of future work, our primary interest is to study the effectiveness of SNFs in systems close to criticality, in order to develop the most suitable strategy for SNFs in this region of the parameter space of a theory. More generally, an analysis of the interplay between Monte Carlo updates and different types of neural-network architectures would be highly insightful and could help one understand what exactly the neural networks are learning when “coupled” to Monte Carlo algorithms in this way. Natural extensions of this work include the use of convolutional architectures for gauge equivariant flows [87, 88], rational quadratic splines [91] and continuous equivariant flows [94].

Acknowledgments

We thank Kim Nicoli and Paolo Stornati for helpful discussions. The numerical simulations were run on machines of the Consorzio Interuniversitario per il Calcolo Automatico

dell'Italia Nord Orientale (CINECA). We acknowledge support from the SFT Scientific Initiative of INFN. This work was partially supported by the “Departments of Excellence 2018–2022” Grant awarded by the Italian Ministry of Education, University and Research (MIUR) (L.232/2016). Part of the numerical functions used in the present work are based on ref. [90].

A Derivation of Jarzynski’s equality for Monte Carlo algorithms

We first set our notation. For a system in thermodynamic equilibrium at temperature T , the statistical distribution of the ϕ configurations is the Boltzmann distribution π :

$$\pi[\phi] = \frac{1}{Z} \exp(-H[\phi]/T). \tag{A.1}$$

The partition function Z is related to the free energy F via

$$Z = \exp(-F/T). \tag{A.2}$$

Let $P[\phi \rightarrow \phi']$ denote the normalized conditional probability of a transition from a configuration ϕ to a configuration ϕ' which defines the Markov-chain algorithm of the Monte Carlo simulation under consideration. At equilibrium, the Boltzmann distribution has to be stationary: the probability that the system evolves from some configuration ϕ to a given configuration ϕ' must be equal to the probability that it evolves from ϕ' to some other configuration, i.e.

$$\sum_{\phi} \pi[\phi] P[\phi \rightarrow \phi'] = \sum_{\phi} \pi[\phi'] P[\phi' \rightarrow \phi]. \tag{A.3}$$

A sufficient (albeit not necessary) condition to enforce the validity of eq. (A.3) is to assume that the summands, not only the sums, are equal:

$$\pi[\phi] P[\phi \rightarrow \phi'] = \pi[\phi'] P[\phi' \rightarrow \phi], \tag{A.4}$$

i.e. the detailed-balance condition.

Let us consider an out-of-equilibrium evolution of the system during the time interval from t_{in} to t_{fin} , denoting the $t_{\text{fin}} - t_{\text{in}}$ difference as Δt , and assuming that this time interval is divided into N subintervals (which we take to be of equal width $\tau = \Delta t/N$, for the sake of simplicity), setting $t_n = t_{\text{in}} + n\tau$ for integer $0 \leq n \leq N$. We identify the discrete time steps t_n with the steps in Monte Carlo time in a Markov-chain algorithm.

Finally, let us introduce the quantity $\mathcal{R}_N[\phi]$ defined as

$$\mathcal{R}_N[\phi] = \exp\left(-\frac{1}{T} \sum_{n=0}^{N-1} \{H_{\eta_{n+1}}[\phi_n] - H_{\eta_n}[\phi_n]\}\right), \tag{A.5}$$

which represents the sum of the exponentiated work (divided by T) done on the system during each of the time intervals of width τ , when the couplings are switched from η_n to

η_{m+1} . In the $N \rightarrow \infty$ limit, $\mathcal{R}_N[\phi]$ tends to the quantity that is averaged over on the right-hand side of eq. (2.1). Using eq. (A.1), $\mathcal{R}_N[\phi]$ can be rewritten in terms of the Boltzmann distribution as

$$\mathcal{R}_N[\phi] = \prod_{n=0}^{N-1} \frac{Z_{\eta_{n+1}} \pi_{\eta_{n+1}}[\phi_n]}{Z_{\eta_n} \pi_{\eta_n}[\phi_n]}, \quad (\text{A.6})$$

so that the average of eq. (A.6) over all possible trajectories from t_{in} to t_{fin} can be written as

$$\overline{\exp(-W/T)} = \lim_{N \rightarrow \infty} \sum_{\{\phi_n\}_{n=0}^N} \pi_{\eta_{\text{in}}}[\phi_0] \prod_{n=0}^{N-1} \left\{ \frac{Z_{\eta_{n+1}}}{Z_{\eta_n}} \cdot \frac{\pi_{\eta_{n+1}}[\phi_n]}{\pi_{\eta_n}[\phi_n]} \cdot P_{\eta_{n+1}}[\phi_n \rightarrow \phi_{n+1}] \right\}, \quad (\text{A.7})$$

having used the fact that the system is initially in thermal equilibrium, hence the probability distribution for the configurations at $t = t_{\text{in}}$ is given by eq. (A.1), and having denoted the $N + 1$ sums over configurations at $t_{\text{in}}, t_1, t_2, \dots, t_{N-1}, t_{\text{fin}}$ as

$$\sum_{\{\phi_n\}_{n=0}^N} = \sum_{\phi_0} \sum_{\phi_1} \sum_{\phi_2} \cdots \sum_{\phi_{N-1}} \sum_{\phi_N}. \quad (\text{A.8})$$

The product of ratios of partition functions in eq. (A.7) simplifies to $Z_{\eta_{\text{fin}}}/Z_{\eta_{\text{in}}}$. Moreover, using eq. (A.4), the sum appearing on the right-hand side of eq. (A.7) can be rewritten as

$$\overline{\exp(-W/T)} = \frac{Z_{\eta_{\text{fin}}}}{Z_{\eta_{\text{in}}}} \lim_{N \rightarrow \infty} \sum_{\{\phi_n\}_{n=0}^N} \pi_{\eta_{\text{in}}}[\phi_0] \prod_{n=0}^{N-1} \left\{ \frac{\pi_{\eta_{n+1}}[\phi_{n+1}]}{\pi_{\eta_n}[\phi_n]} \cdot P_{\eta_{n+1}}[\phi_{n+1} \rightarrow \phi_n] \right\}. \quad (\text{A.9})$$

In the latter expression, the ratios of Boltzmann distributions simplify to $\pi_{\eta_{\text{fin}}}[\phi_N]/\pi_{\eta_{\text{in}}}[\phi_0]$, which, in turn, simplifies against the $\pi_{\eta_{\text{in}}}[\phi_0]$ factor:

$$\overline{\exp(-W/T)} = \frac{Z_{\eta_{\text{fin}}}}{Z_{\eta_{\text{in}}}} \lim_{N \rightarrow \infty} \sum_{\{\phi_n\}_{n=0}^N} \pi_{\eta_{\text{fin}}}[\phi_N] \prod_{n=0}^{N-1} P_{\eta_{n+1}}[\phi_{n+1} \rightarrow \phi_n]. \quad (\text{A.10})$$

The sum over the initial configurations can be performed explicitly, as ϕ_0 appears only in the $P_{\eta_1}[\phi_1 \rightarrow \phi_0]$ term, and the result is 1, due to the normalization of the conditional transition probability. Next, the same argument can be repeated to sum over the $\phi_1, \phi_2, \dots, \phi_{N-1}$ configurations. Finally, noting that also $\pi_{\eta_{\text{fin}}}[\phi_N]$ is normalized to 1, one obtains

$$\overline{\exp(-W/T)} = \frac{Z_{\eta_{\text{fin}}}}{Z_{\eta_{\text{in}}}}, \quad (\text{A.11})$$

which is eq. (2.1).

We remark that, although in this proof we used the Boltzmann distributions at all times $t_{\text{in}} \leq t \leq t_{\text{fin}}$, we did this only to re-express the $\exp(-H/T)$ terms appearing in eq. (A.5). The configurations ϕ_n at $t > t_{\text{in}}$ are *not* in thermal equilibrium. Moreover, for simplicity, we assumed the temperature T to be constant throughout the evolution of the system along each trajectory, but this does not necessarily have to be the case [24].

Open Access. This article is distributed under the terms of the Creative Commons Attribution License ([CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited. SCOAP³ supports the goals of the International Year of Basic Sciences for Sustainable Development.

References

- [1] D.J. Schwarz, *The first second of the universe*, *Annalen Phys.* **12** (2003) 220 [[astro-ph/0303574](#)] [[INSPIRE](#)].
- [2] C. Caprini et al., *Detecting gravitational waves from cosmological phase transitions with LISA: an update*, *JCAP* **03** (2020) 024 [[arXiv:1910.13125](#)] [[INSPIRE](#)].
- [3] K. Kajantie, M. Laine, K. Rummukainen and M.E. Shaposhnikov, *Is there a hot electroweak phase transition at $m_H \gtrsim m_W$?*, *Phys. Rev. Lett.* **77** (1996) 2887 [[hep-ph/9605288](#)] [[INSPIRE](#)].
- [4] K. Kajantie, M. Laine, K. Rummukainen and M.E. Shaposhnikov, *The Electroweak phase transition: A Nonperturbative analysis*, *Nucl. Phys. B* **466** (1996) 189 [[hep-lat/9510020](#)] [[INSPIRE](#)].
- [5] Y. Aoki, G. Endrődi, Z. Fodor, S.D. Katz and K.K. Szabó, *The Order of the quantum chromodynamics transition predicted by the standard model of particle physics*, *Nature* **443** (2006) 675 [[hep-lat/0611014](#)] [[INSPIRE](#)].
- [6] A. Bazavov et al., *The chiral and deconfinement aspects of the QCD transition*, *Phys. Rev. D* **85** (2012) 054503 [[arXiv:1111.1710](#)] [[INSPIRE](#)].
- [7] E. Witten, *Cosmic Separation of Phases*, *Phys. Rev. D* **30** (1984) 272 [[INSPIRE](#)].
- [8] C.J. Hogan, *Gravitational radiation from cosmological phase transitions*, *Mon. Not. Roy. Astron. Soc.* **218** (1986) 629 [[INSPIRE](#)].
- [9] A. Kosowsky, M.S. Turner and R. Watkins, *Gravitational waves from first order cosmological phase transitions*, *Phys. Rev. Lett.* **69** (1992) 2026 [[INSPIRE](#)].
- [10] K.G. Wilson, *Confinement of Quarks*, *Phys. Rev. D* **10** (1974) 2445 [[INSPIRE](#)].
- [11] J. Engels, J. Fingberg, F. Karsch, D. Miller and M. Weber, *Nonperturbative thermodynamics of SU(N) gauge theories*, *Phys. Lett. B* **252** (1990) 625 [[INSPIRE](#)].
- [12] A.M. Ferrenberg and R.H. Swendsen, *New Monte Carlo Technique for Studying Phase Transitions*, *Phys. Rev. Lett.* **61** (1988) 2635 [[INSPIRE](#)].
- [13] K. Langfeld, B. Lucini and A. Rago, *The density of states in gauge theories*, *Phys. Rev. Lett.* **109** (2012) 111601 [[arXiv:1204.3243](#)] [[INSPIRE](#)].
- [14] FLOWQCD collaboration, *Thermodynamics of SU(3) gauge theory from gradient flow on the lattice*, *Phys. Rev. D* **90** (2014) 011501 [*Erratum ibid.* **92** (2015) 059902] [[arXiv:1312.7492](#)] [[INSPIRE](#)].
- [15] K. Langfeld and B. Lucini, *Density of states approach to dense quantum systems*, *Phys. Rev. D* **90** (2014) 094502 [[arXiv:1404.7187](#)] [[INSPIRE](#)].
- [16] M. Kitazawa, T. Iritani, M. Asakawa, T. Hatsuda and H. Suzuki, *Equation of State for SU(3) Gauge Theory via the Energy-Momentum Tensor under Gradient Flow*, *Phys. Rev. D* **94** (2016) 114512 [[arXiv:1610.07810](#)] [[INSPIRE](#)].
- [17] L. Giusti and M. Pepe, *Equation of state of the SU(3) Yang-Mills theory: A precise determination from a moving frame*, *Phys. Lett. B* **769** (2017) 385 [[arXiv:1612.00265](#)] [[INSPIRE](#)].
- [18] K. Kanaya et al., *Equation of state in (2 + 1)-flavor QCD with gradient flow*, *PoS LATTICE2016* (2016) 063 [[arXiv:1610.09518](#)] [[INSPIRE](#)].

- [19] M. Dalla Brida, L. Giusti and M. Pepe, *QCD in a moving frame: an exploratory study*, *EPJ Web Conf.* **175** (2018) 14012 [[arXiv:1710.09219](#)] [[INSPIRE](#)].
- [20] T. Hirakida, E. Itou and H. Kouno, *Thermodynamics for pure SU(2) gauge theory using gradient flow*, *Prog. Theor. Exp. Phys.* **2019** (2019) 033B01 [[arXiv:1805.07106](#)] [[INSPIRE](#)].
- [21] T. Iritani, M. Kitazawa, H. Suzuki and H. Takaura, *Thermodynamics in quenched QCD: energy-momentum tensor with two-loop order coefficients in the gradient-flow formalism*, *Prog. Theor. Exp. Phys.* **2019** (2019) 023B02 [[arXiv:1812.06444](#)] [[INSPIRE](#)].
- [22] Y. Che, C. Gneiting and F. Nori, *Estimating the Euclidean quantum propagator with deep generative modeling of Feynman paths*, *Phys. Rev. B* **105** (2022) 214205 [[arXiv:2202.02750](#)] [[INSPIRE](#)].
- [23] C. Chatelain and D. Karevski, *Probability distributions of the work in the two-dimensional Ising model*, *J. Stat. Mech.* **0606** (2006) P06005 [[cond-mat/0602580](#)].
- [24] C. Chatelain, *Temperature-extended Jarzynski relation: Application to the numerical calculation of the surface tension*, *J. Stat. Mech.* **0704** (2007) P04011 [[cond-mat/0702044](#)] [[INSPIRE](#)].
- [25] H. Híjar, J. Quintana Hinojosa and G. Sutmann, *Non-equilibrium work theorems for the two-dimensional Ising model*, *J. Stat. Mech.* **0704** (2007) P04010.
- [26] M. Caselle, G. Costagliola, A. Nada, M. Panero and A. Toniato, *Jarzynski's theorem for lattice gauge theory*, *Phys. Rev. D* **94** (2016) 034503 [[arXiv:1604.05544](#)] [[INSPIRE](#)].
- [27] M. Caselle, A. Nada and M. Panero, *QCD thermodynamics from lattice calculations with nonequilibrium methods: The SU(3) equation of state*, *Phys. Rev. D* **98** (2018) 054513 [[arXiv:1801.03110](#)] [[INSPIRE](#)].
- [28] O. Francesconi, M. Panero and D. Preti, *Strong coupling from non-equilibrium Monte Carlo simulations*, *JHEP* **07** (2020) 233 [[arXiv:2003.13734](#)] [[INSPIRE](#)].
- [29] C. Jarzynski, *Nonequilibrium Equality for Free Energy Differences*, *Phys. Rev. Lett.* **78** (1997) 2690 [[cond-mat/9610209](#)] [[INSPIRE](#)].
- [30] C. Jarzynski, *Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach*, *Phys. Rev. E* **56** (1997) 5018 [[cond-mat/9707325](#)].
- [31] D.J. Evans, E.G.D. Cohen and G.P. Morriss, *Probability of second law violations in shearing steady states*, *Phys. Rev. Lett.* **71** (1993) 2401.
- [32] D.J. Evans and D.J. Searles, *Equilibrium microstates which generate second law violating steady states*, *Phys. Rev. E* **50** (1994) 1645.
- [33] G. Gallavotti and E.G.D. Cohen, *Dynamical Ensembles in Nonequilibrium Statistical Mechanics*, *Phys. Rev. Lett.* **74** (1995) 2694 [[chao-dyn/9410007](#)] [[INSPIRE](#)].
- [34] G. Gallavotti and E.G.D. Cohen, *Dynamical ensembles in stationary states*, *J. Stat. Phys.* **80** (1995) 931 [[chao-dyn/9501015](#)].
- [35] G.E. Crooks, *Nonequilibrium Measurements of Free Energy Differences for Microscopically Reversible Markovian Systems*, *J. Stat. Phys.* **90** (1998) 1481.
- [36] G.E. Crooks, *Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences*, *Phys. Rev. E* **60** (1999) 2721 [[cond-mat/9901352](#)].

- [37] F. Ritort, *Work fluctuations, transient violations of the second law and free-energy recovery methods: Perspectives in Theory and Experiments*, in *Poincaré Seminar 2*, Birkhäuser Verlag, Basel, Switzerland (2003), pp. 195–229 [[cond-mat/0401311](#)].
- [38] U. Marini Bettolo Marconi, A. Puglisi, L. Rondoni and A. Vulpiani, *Fluctuation dissipation: Response theory in statistical physics*, *Phys. Rept.* **461** (2008) 111 [[arXiv:0803.0719](#)].
- [39] G.N. Bochkov and Y.E. Kuzovlev, *General theory of thermal fluctuations in nonlinear systems*, *Sov. Phys. JETP* **45** (1977) 125.
- [40] G.N. Bochkov and Y.E. Kuzovlev, *Fluctuation-dissipation relations for non-equilibrium processes in open systems*, *Sov. Phys. JETP* **49** (1979) 543.
- [41] G.N. Bochkov and Y.E. Kuzovlev, *Nonlinear fluctuation-dissipation relations and stochastic models in nonequilibrium thermodynamics*, *Physica A* **106** (1981) 443.
- [42] C. Jarzynski, *Comparison of far-from-equilibrium work relations*, *Compt. Rendus Phys.* **8** (2007) 495 [[cond-mat/0612305](#)].
- [43] Y. Kuzovlev, *Short remarks on the so-called fluctuation theorems and related statements*, [arXiv:1106.0589](#).
- [44] M. Esposito, U. Harbola and S. Mukamel, *Nonequilibrium fluctuations, fluctuation theorems, and counting statistics in quantum systems*, *Rev. Mod. Phys.* **81** (2009) 1665 [[arXiv:0811.3717](#)] [[INSPIRE](#)].
- [45] L. D’Alessio, Y. Kafri, A. Polkovnikov and M. Rigol, *From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics*, *Adv. Phys.* **65** (2016) 239 [[arXiv:1509.06411](#)] [[INSPIRE](#)].
- [46] M. Campisi and J. Goold, *Thermodynamics of quantum information scrambling*, *Phys. Rev. E* **95** (2017) 062127 [[arXiv:1609.05848](#)] [[INSPIRE](#)].
- [47] N. Yunger Halpern, *Jarzynski-like equality for the out-of-time-ordered correlator*, *Phys. Rev. A* **95** (2017) 012120 [[arXiv:1609.00015](#)] [[INSPIRE](#)].
- [48] N. Yunger Halpern, B. Swingle and J. Dressel, *Quasiprobability behind the out-of-time-ordered correlator*, *Phys. Rev. A* **97** (2018) 042105 [[arXiv:1704.01971](#)] [[INSPIRE](#)].
- [49] T. Mori, T.N. Ikeda, E. Kaminishi and M. Ueda, *Thermalization and prethermalization in isolated quantum systems: a theoretical overview*, *J. Phys. B* **51** (2018) 112001 [[arXiv:1712.08790](#)] [[INSPIRE](#)].
- [50] A. Chenu, J. Molina-Vilaplana and A. Del Campo, *Work Statistics, Loschmidt Echo and Information Scrambling in Chaotic Quantum Systems*, *Quantum* **3** (2019) 127 [[arXiv:1804.09188](#)] [[INSPIRE](#)].
- [51] J.L.W.V. Jensen, *Sur les fonctions convexes et les inégalités entre les valeurs moyennes*, *Acta Math.* **30** (1906) 175.
- [52] R.M. Neal, *Annealed Importance Sampling*, [physics/9803008](#).
- [53] R.M. Neal, *Annealed importance sampling*, *Stat. Comput.* **11** (2001) 125.
- [54] G. Carleo et al., *Machine learning and the physical sciences*, *Rev. Mod. Phys.* **91** (2019) 045002 [[arXiv:1903.10563](#)] [[INSPIRE](#)].
- [55] D. Guest, K. Cranmer and D. Whiteson, *Deep Learning and its Application to LHC Physics*, *Ann. Rev. Nucl. Part. Sci.* **68** (2018) 161 [[arXiv:1806.11484](#)] [[INSPIRE](#)].

- [56] A. Radovic et al., *Machine learning at the energy and intensity frontiers of particle physics*, *Nature* **560** (2018) 41 [INSPIRE].
- [57] M.D. Schwartz, *Modern Machine Learning and Particle Physics*, [arXiv:2103.12226](#) [INSPIRE].
- [58] B.H. Denby, *Neural Networks and Cellular Automata in Experimental High-energy Physics*, *Comput. Phys. Commun.* **49** (1988) 429 [INSPIRE].
- [59] L. Lönnblad, C. Peterson and T. Rönvaldsson, *Using neural networks to identify jets*, *Nucl. Phys. B* **349** (1991) 675 [INSPIRE].
- [60] B.H. Denby, *The Use of neural networks in high-energy physics*, *Neural Comput.* **5** (1993) 505.
- [61] R. Mankel, *Pattern recognition and event reconstruction in particle physics experiments*, *Rept. Prog. Phys.* **67** (2004) 553 [[physics/0402039](#)] [INSPIRE].
- [62] ATLAS collaboration, *A neural network clustering algorithm for the ATLAS silicon pixel detector*, *2014 JINST* **9** P09009 [[arXiv:1406.7690](#)] [INSPIRE].
- [63] NNPDF collaboration, *Parton distributions for the LHC Run II*, *JHEP* **04** (2015) 040 [[arXiv:1410.8849](#)] [INSPIRE].
- [64] J. Barnard, E.N. Dawe, M.J. Dolan and N. Rajcic, *Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks*, *Phys. Rev. D* **95** (2017) 014018 [[arXiv:1609.00607](#)] [INSPIRE].
- [65] G. Louppe, K. Cho, C. Becot and K. Cranmer, *QCD-Aware Recursive Neural Networks for Jet Physics*, *JHEP* **01** (2019) 057 [[arXiv:1702.00748](#)] [INSPIRE].
- [66] S.J. Wetzel and M. Scherzer, *Machine Learning of Explicit Order Parameters: From the Ising Model to SU(2) Lattice Gauge Theory*, *Phys. Rev. B* **96** (2017) 184410 [[arXiv:1705.05582](#)] [INSPIRE].
- [67] K. Zhou, G. Endrődi, L.-G. Pang and H. Stöcker, *Regressive and generative neural networks for scalar field theory*, *Phys. Rev. D* **100** (2019) 011501 [[arXiv:1810.12879](#)] [INSPIRE].
- [68] P.E. Shanahan, D. Trewartha and W. Detmold, *Machine learning action parameters in lattice quantum chromodynamics*, *Phys. Rev. D* **97** (2018) 094506 [[arXiv:1801.05784](#)] [INSPIRE].
- [69] J.M. Pawłowski and J.M. Urban, *Reducing Autocorrelation Times in Lattice Simulations with Generative Adversarial Networks*, *Mach. Learn. Sci. Tech.* **1** (2020) 045011 [[arXiv:1811.03533](#)] [INSPIRE].
- [70] B. Yoon, T. Bhattacharya and R. Gupta, *Machine Learning Estimators for Lattice QCD Observables*, *Phys. Rev. D* **100** (2019) 014504 [[arXiv:1807.05971](#)] [INSPIRE].
- [71] C. Giannetti, B. Lucini and D. Vadicchino, *Machine Learning as a universal tool for quantitative investigations of phase transitions*, *Nucl. Phys. B* **944** (2019) 114639 [[arXiv:1812.06726](#)] [INSPIRE].
- [72] S.-H. Li and L. Wang, *Neural Network Renormalization Group*, *Phys. Rev. Lett.* **121** (2018) 260601 [[arXiv:1802.02840](#)].
- [73] T. Matsumoto, M. Kitazawa and Y. Kohno, *Classifying topological charge in SU(3) Yang-Mills theory with machine learning*, *Prog. Theor. Exp. Phys.* **2021** (2021) 023D01 [[arXiv:1909.06238](#)] [INSPIRE].

- [74] M.N. Chernodub, H. Erbin, V.A. Goy and A.V. Molochkov, *Topological defects and confinement with machine learning: the case of monopoles in compact electrodynamics*, *Phys. Rev. D* **102** (2020) 054501 [[arXiv:2006.09113](#)] [[INSPIRE](#)].
- [75] S. Blücher, L. Kades, J.M. Pawłowski, N. Strodthoff and J.M. Urban, *Towards novel insights in lattice field theory with explainable machine learning*, *Phys. Rev. D* **101** (2020) 094507 [[arXiv:2003.01504](#)] [[INSPIRE](#)].
- [76] M. Favoni, A. Ipp, D.I. Müller and D. Schuh, *Lattice Gauge Equivariant Convolutional Neural Networks*, *Phys. Rev. Lett.* **128** (2022) 032003 [[arXiv:2012.12901](#)] [[INSPIRE](#)].
- [77] D.L. Boyda, M.N. Chernodub, N.V. Gerasimeniuk, V.A. Goy, S.D. Liubimov and A.V. Molochkov, *Finding the deconfinement temperature in lattice Yang-Mills theories from outside the scaling window with machine learning*, *Phys. Rev. D* **103** (2021) 014509 [[arXiv:2009.10971](#)] [[INSPIRE](#)].
- [78] M. Medvidovic, J. Carrasquilla, L.E. Hayward and B. Kulchytskyy, *Generative models for sampling of lattice field theories*, in proceedings of the *34th Conference on Neural Information Processing Systems*, online conference, Canada, 6–12 December 2020 [[arXiv:2012.01442](#)] [[INSPIRE](#)].
- [79] D. Bachtis, G. Aarts and B. Lucini, *Quantum field-theoretic machine learning*, *Phys. Rev. D* **103** (2021) 074510 [[arXiv:2102.09449](#)] [[INSPIRE](#)].
- [80] M. Gabrié, G.M. Rotskoff and E. Vanden-Eijnden, *Adaptive Monte Carlo augmented with normalizing flows*, [arXiv:2105.12603](#) [[INSPIRE](#)].
- [81] A. Singha, D. Chakrabarti and V. Arora, *Generative learning for the problem of critical slowing down in lattice Gross Neveu model*, [arXiv:2111.00574](#) [[INSPIRE](#)].
- [82] USQCD collaboration, *Status and Future Perspectives for Lattice Gauge Theory Calculations to the Exascale and Beyond*, *Eur. Phys. J. A* **55** (2019) 199 [[arXiv:1904.09725](#)] [[INSPIRE](#)].
- [83] E.G. Tabak and E. Vanden-Eijnden, *Density estimation by dual ascent of the log-likelihood*, *Comm. Math. Sci.* **8** (2010) 217.
- [84] I. Kobyzev, S.J. Prince and M.A. Brubaker, *Normalizing Flows: An Introduction and Review of Current Methods*, *IEEE Trans. Pattern Anal. Mach. Intell.* **43** (2020) 3964 [[arXiv:1908.09257](#)].
- [85] G. Papamakarios, E. Nalisnick, D. Jimenez Rezende, S. Mohamed and B. Lakshminarayanan, *Normalizing Flows for Probabilistic Modeling and Inference*, *J. Mach. Learn. Res.* **22** (2021) 1 [[arXiv:1912.02762](#)].
- [86] 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** (2019) 034515 [[arXiv:1904.12072](#)] [[INSPIRE](#)].
- [87] G. Kanwar et al., *Equivariant flow-based sampling for lattice gauge theory*, *Phys. Rev. Lett.* **125** (2020) 121601 [[arXiv:2003.06413](#)] [[INSPIRE](#)].
- [88] D. Boyda et al., *Sampling using $SU(N)$ gauge equivariant flows*, *Phys. Rev. D* **103** (2021) 074504 [[arXiv:2008.05456](#)] [[INSPIRE](#)].
- [89] K.A. Nicoli et al., *Estimation of Thermodynamic Observables in Lattice Field Theories with Deep Generative Models*, *Phys. Rev. Lett.* **126** (2021) 032001 [[arXiv:2007.07115](#)] [[INSPIRE](#)].

- [90] M.S. Albergo et al., *Introduction to Normalizing Flows for Lattice Field Theory*, [arXiv:2101.08176](#) [INSPIRE].
- [91] L. Del Debbio, J.M. Rossney and M. Wilson, *Efficient modeling of trivializing maps for lattice ϕ^4 theory using normalizing flows: A first look at scalability*, *Phys. Rev. D* **104** (2021) 094507 [[arXiv:2105.12481](#)] [INSPIRE].
- [92] M.S. Albergo et al., *Flow-based sampling for fermionic lattice field theories*, *Phys. Rev. D* **104** (2021) 114507 [[arXiv:2106.05934](#)] [INSPIRE].
- [93] D.C. Hackett et al., *Flow-based sampling for multimodal distributions in lattice field theory*, [arXiv:2107.00734](#) [INSPIRE].
- [94] P. de Haan, C. Rainone, M.C.N. Cheng and R. Bondesan, *Scaling Up Machine Learning For Quantum Field Theory with Equivariant Continuous Flows*, [arXiv:2110.02673](#) [INSPIRE].
- [95] J. Finkenrath, *Tackling critical slowing down using global correction steps with equivariant flows: the case of the Schwinger model*, [arXiv:2201.02216](#) [INSPIRE].
- [96] H. Wu, J. Köhler and F. Noé, *Stochastic Normalizing Flows*, [arXiv:2002.06707](#).
- [97] S. Vaikuntanathan and C. Jarzynski, *Escorted free energy simulations*, *J. Chem. Phys.* **134** (2011) 054107 [[arXiv:1101.2612](#)].
- [98] D. Jimenez Rezende and S. Mohamed, *Variational Inference with Normalizing Flows*, in proceedings of the *32nd International Conference on Machine Learning*, Lille, France, 7–9 July 2015, *Proceedings of Machine Learning Research* **37**, F. Bach and D. Blei eds., PMLR (2015), pp. 1530–1538 [[arXiv:1505.05770](#)] and online at <http://proceedings.mlr.press/v37/rezende15.html>.
- [99] S. Kullback and R.A. Leibler, *On Information and Sufficiency*, *Ann. Math. Stat.* **22** (1951) 79.
- [100] L. Dinh, D. Krueger and Y. Bengio, *NICE: Non-linear Independent Components Estimation*, [arXiv:1410.8516](#).
- [101] L. Dinh, J. Sohl-Dickstein and S. Bengio, *Density estimation using Real NVP*, [arXiv:1605.08803](#).
- [102] L. Tierney, *Markov Chains for Exploring Posterior Distributions*, *Ann. Statist.* **22** (1994) 1701.
- [103] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, [arXiv:1412.6980](#) [INSPIRE].