Applications of flow models to the generation of correlated lattice QCD ensembles

Ryan Abbott,[1,2] Aleksandar Botev,[3] Denis Boyda,[1,2] Daniel C. Hackett,[4,1,2] Gurtej Kanwar,[5] Sébastien Racanière,[3] Danilo J. Rezende,[3] Fernando Romero-López,[1,2] Phiala E. Shanahan,[1,2] and Julian M. Urban[1,2]

[1] *Center for Theoretical Physics, Massachusetts Institute of Technology,*
Cambridge, Massachusetts 02139, USA
[2] *The NSF AI Institute for Artificial Intelligence and Fundamental Interactions*
[3] *Google DeepMind, London, UK*
[4] *Fermi National Accelerator Laboratory, Batavia, IL 60510, U.S.A.*
[5] *Albert Einstein Center, Institute for Theoretical Physics, University of Bern, 3012 Bern, Switzerland*

(Received 30 January 2024; accepted 22 April 2024; published 24 May 2024)

Machine-learned normalizing flows can be used in the context of lattice quantum field theory to generate statistically correlated ensembles of lattice gauge fields at different action parameters. This work demonstrates how these correlations can be exploited for variance reduction in the computation of observables. Three different proof-of-concept applications are demonstrated using a novel residual flow architecture: continuum limits of gauge theories, the mass dependence of QCD observables, and hadronic matrix elements based on the Feynman–Hellmann approach. In all three cases, it is shown that statistical uncertainties are significantly reduced when machine-learned flows are incorporated as compared with the same calculations performed with uncorrelated ensembles or direct reweighting.

DOI: 10.1103/PhysRevD.109.094514

I. INTRODUCTION

Understanding the strongly interacting sector of the Standard Model of particle physics, described by the theory of quantum chromodynamics (QCD), is essential for advancing particle and nuclear physics. The numerical framework of lattice QCD is a systematically improvable tool to explore the dynamics of the strong nuclear force. This approach has enabled precise calculations across applications spanning from hadron structure to high-temperature QCD and nuclear physics [1,2]. Nevertheless, there is great potential to extend the reach of lattice QCD beyond the current state of the art if computational challenges such as critical slowing down, topological freezing, and signal-to-noise problems can be overcome. In this context, emerging machine learning techniques offer a promising avenue towards mitigating these computational obstacles [3,4].

A growing community effort is developing at the intersection of machine learning and lattice QCD—see, e.g., Refs. [5–9] for a selection of applications. In particular, generative flow models [10–12] are one of several promising pathways which show potential to accelerate the sampling of lattice field configurations. This line of investigation is developing, with demonstrations in 2D theories [9,13–40] and first applications to 4D gauge theories with and without fermions [41–43]. While the field is progressing rapidly, achieving high-quality models that can be applied at the scale of state-of-the-art calculations still requires further engineering [44]. In addition to their promise in the context of sampling, flow models—functioning as approximate maps between distributions—can be used to accelerate lattice QCD calculations in qualitatively different ways. For example, flow models provide a promising new approach to determining thermodynamic observables [9,30,39,45].

In this work, we explore applications which utilize flows to map gauge field configurations between distributions defined by different Euclidean lattice action parameters. Such flows can be used to generate multiple statistically correlated ensembles at different parameters. As we explore in this work, this may be particularly valuable when the variation of some quantity with respect to the action parameter is of physical or computational interest—see also Refs. [46,47]. The advantage of flows in this context originates from correlated cancellations of uncertainties between expectation values evaluated at different action parameters, which leads to reductions in the number of configurations needed to achieve a fixed statistical error.

Examples of physically relevant applications of derivatives with respect to action parameters include continuum and chiral extrapolations as well as the computation of matrix elements such as the chiral condensate, the nucleon sigma term, or other observables, using Feynman–Hellmann techniques. Another is derivatives with respect to the electromagnetic coupling for scale setting or to compute isospin breaking corrections in QCD + QED [48,49]. One may also consider applications in theories with a sign problem, e.g., to derivatives with respect to the baryon chemical potential or the QCD θ-term. In all of these cases, the distributions to be related by a flow transformation are much more similar than in applications intended to accelerate sampling, and current flow methods can already be applied at the scale of typical lattice QCD calculations. Three selected applications are investigated, namely the continuum extrapolation of gradient flow scales, the computation of the gluon momentum fraction of the pion in quenched lattice QCD using the Feynman–Hellmann approach, and the mass dependence of observables in $N_f = 2$ QCD.

This paper is organized as follows. In Sec. II, we discuss preliminaries on flows, their applicability in the context of correlated ensembles, and the residual flow architectures used in this work. The three numerical demonstrations are presented in Sec. III. We conclude in Sec. IV. Appendix provides further details of the flow models used in this work.

II. FLOWS FOR THE GENERATION OF CORRELATED ENSEMBLES

A. Flows for lattice QCD

This section presents an introduction to normalizing flows [10–12], reviewing the key ideas relevant for the present work.

A “flow” is defined as a diffeomorphism f between probability distributions that maps samples from a base (or prior) distribution, $r(U)$, to a model distribution with density

$$q(V) = r(U)\left|\det\frac{\partial f(U)}{\partial U}\right|^{-1}, \tag{1}$$

where $V = f(U)$. Flows can be constructed such that they have many free, trainable parameters. These parameters may be optimized such that the model distribution approximates some target distribution p, i.e., $q(V) \simeq p(V)$.

For the applications explored in this work, flow models are constructed in which the samples U are lattice gauge-field configurations, and the probability distributions $p(U)$ and $r(U)$ are defined in terms of Euclidean lattice actions such that $r(U) \propto \exp(-S_0(U))$, and $p(V) \propto \exp(-S_1(V))$. In most cases, it is not necessary to know the normalization of p or r (the exception being thermodynamic observables [9]).

Expressive flow transformations can be constructed in a variety of ways, for example as the composition of n invertible layers

$$f = g_1 \circ g_2 \circ \ldots \circ g_n. \tag{2}$$

Architectures for invertible layers g_i which act on lattice gauge fields have been discussed in Ref. [43]. The particular constructions used in this work are detailed in Sec. II C. Given a model, its trainable parameters may be optimized in various ways. One choice is to minimize the Kullback–Leibler (KL) divergence [50] between the model and target distributions. Approaches such as path gradients [51], related control variate methods [43], as well as the “REINFORCE” algorithm [52], may be be used to improve and accelerate training dynamics by reducing the variance associated with stochastic gradient estimates. After optimization, model quality can be characterized using the Effective Sample Size per configuration (ESS),

$$\mathrm{ESS} = \frac{1}{N}\frac{[\sum_{i=1}^{N} w(V_i)]^2}{\sum_{i=1}^{N}[w(V_i)]^2}, \tag{3}$$

estimated using N gauge field configurations generated from $q(V)$, and where $w(V_i) = p(V_i)/q(V_i)$ is the reweighting factor of the ith configuration. The values of the ESS lie in the interval $\mathrm{ESS} \in [1/N, 1]$, with $\mathrm{ESS} = 1$ corresponding to a perfect model.

In practice, a learned flow is not perfect, but may function as an approximate map between distributions. To ensure correctness of expectation values computed on the flowed configurations, one may use the independence Metropolis algorithm [53–55] or simply reweighting, with the weight of each configuration given by $w(U)$. Expectation values of observables such as plaquettes, hadronic correlation functions, or the topological charge can be directly reweighted as

$$\langle \mathcal{O} \rangle_p = \langle w\mathcal{O} \rangle_q, \tag{4}$$

where the notation $\langle\rangle_q$ is used to refer to expectation values with respect to the probability distribution q, and we assume the reweighting factors have been properly normalized such that $\langle w \rangle_q = 1$. Derived quantities, such as gradient flow scales or hadron masses, can be computed from reweighted correlation functions. Statistical uncertainties in reweighted quantities are typically larger than those before reweighting. A rough estimate of the increase in the variance is a factor of $\simeq 1/\mathrm{ESS}$.

B. Correlated ensembles and flows

While applications of flows to accelerate the generation of field configurations continue to advance, here we describe another avenue for flow models to improve lattice QCD calculations by reducing the variance of observables that can be computed from differences between quantities at different action parameters. The key idea is the following. Consider a generic parameter of the action, α. The goal is to compute some observable $\mathcal{O}$ as a function of α, and in particular the derivative

$$\frac{d\langle\mathcal{O}\rangle}{d\alpha} \simeq \frac{\langle\mathcal{O}\rangle_{\alpha_1} - \langle\mathcal{O}\rangle_{\alpha_2}}{\Delta\alpha}, \tag{5}$$

where the right-hand side is a finite-difference approximation of the derivative using $\Delta\alpha = \alpha_1 - \alpha_2$, with $\langle\rangle_\alpha$ denoting the expectation under the distribution defined by the action parameter α, i.e., p_α. Higher order derivatives, or derivatives of one observable with respect to another, may be computed in a similar way.

In this work, we consider three qualitatively different approaches to the computation of the quantity in Eq. (5). The first two are standard tools in common use:

(1) Use a very small step $\Delta\alpha = \epsilon$, and compute the numerator in Eq. (5) with ϵ *reweighting*:

$$\langle\mathcal{O}\rangle_{\alpha_1} - \langle\mathcal{O}\rangle_{\alpha_1+\epsilon} = \langle\mathcal{O} - w_\epsilon\mathcal{O}\rangle_{\alpha_1}, \tag{6}$$

where $w_\epsilon = p_{\alpha_1+\epsilon}/p_{\alpha_1}$. For this approach, the ESS generically degrades as $\mathrm{ESS} = 1 - k(\Delta\alpha)^2 + \cdots$, where k is a problem-specific constant. The separation ϵ may be made small without compromising signal to noise due to correlated noise cancellations between the two expectation values. As $\epsilon \to 0$ it becomes exact, recovering an estimate statistically identical to that obtained by applying the derivative analytically.

(2) Generate *independent ensembles* to separately compute expectation values at α_1 and α_2 in Eq. (5). This enables use of much more widely separated α_1 and α_2 than accessible with reweighting, thereby allowing exploitation of the bias-variance tradeoff to reduce statistical uncertainties while accepting additional discretization artifacts from the finite-difference approximation in order to improve signal-to-noise. However, this effect must be sufficiently large to compensate for the lack of correlated noise cancellations.

These two methods each have different capabilities, with each useful for different applications. Incorporating flows provides an additional approach that combines some of the advantages of both:

(3) Use a trained *flow model* to map configurations between the distributions given by α_1 and α_2. Including flow reweighting factors, correlated differences can be calculated as:

$$\langle\mathcal{O}(U) - w(f(U))\mathcal{O}(f(U))\rangle_{\alpha_1}, \tag{7}$$

where $w(f(U)) = p_{\alpha_2}(f(U))/q(f(U))$, such that a perfect flow would remove the reweighting factors entirely. This approach benefits from the same correlated cancellation of uncertainties as does ϵ reweighting, while allowing for larger steps in $\Delta\alpha$ to exploit the bias-variance tradeoff as does the approach using independent ensembles.

In Sec. III below, we provide numerical demonstrations of the advantages of this flow-based approach.

Note that the latter two approaches, with finite separation in α, can be combined with improved finite-difference estimators of derivatives to reduce the $O(\Delta\alpha)$ bias, or by fitting the α dependence at the cost of introducing model dependence.

C. Architecture based on residual flows

The flow architecture used in this work is based on that introduced in Ref. [43], with a series of improvements that are detailed below. The flow transformation is defined as the composition of trainable gauge-equivariant layers that act directly on the gauge links. The transformation of a gauge field $U \to U'$ through an $\mathrm{SU}(N)$-residual layer can be expressed as

$$U'_\mu(x) = e^{g_x(U)}U_\mu(x), \tag{8}$$

where $g_x(U)$ is an algebra-valued matrix which can in principle have an arbitrary dependence on the entire gauge-field configuration, as long as it transforms locally under gauge transformations, $g_x(U) \to \Omega_x^\dagger g_x(U)\Omega_x$; here Ω_x denotes a gauge transformation and the subscript labels the spacetime dependence. This transformation can be inverted by fixed point iteration, with a unique solution guaranteed if the Lipschitz continuity condition is satisfied [43].

For numerical tractability, each layer partitions the gauge field and transforms only the *active links*, defined as those with fixed direction μ on a subset of lattice sites $\{x_a\}$, conditioned on the values of the remaining *frozen links* U_f. Each layer acts as

$$U'_\mu(x_a) = e^{g_x(U_f,U_\mu(x_a))}U_\mu(x_a), \tag{9}$$

that is, g_x for any given active link depends on all frozen links but only the same active link. This separation of variables allows efficient computation of the Jacobian of the transformation using automatic differentiation as described in Eq. (26) of Ref. [43]. In the present work, we use two partitioning schemes (also referred to as "masking patterns") for the site index:

(1) A checkerboard or "mod 2" masking pattern, where the active links are those with direction μ in the positions that satisfy $(p + \sum_\mu x_\mu) = 0\,(\mathrm{mod}\,2)$ for $p \in 0,\ 1$. A stack of eight layers is needed to transform all links, i.e., two complementary checkerboards in each of the four directions μ. This is a

simple nontrivial choice that updates all variables within a small number of layers.

(2) A "mod 4" masking pattern, where the positions of active links satisfy $(p + \sum_\mu x_\mu) = 0\,(\mathrm{mod}\,4)$, for $p \in 0,\ 1,\ 2,\ 3$. Sixteen layers are thus needed to transform every link on the lattice. This choice is more expensive than the "mod 2" pattern described above, but it can also be more expressive by allowing a more complicated dependence of the transformation on the frozen links.

The function $g_x(U_f, U_\mu(x_a))$ must be constructed in a way that is expressive but simple to evaluate. One simple construction utilizes 1×1 staples (depicted in Fig. 1),

$$\begin{aligned} S^R_{x,\mu\nu}(U) &= U_\nu(x+\mu)U^\dagger_\mu(x+\nu)U^\dagger_\nu(x) \quad \text{and} \\ S^L_{x,\mu\nu}(U) &= U^\dagger_\nu(x+\mu-\nu)U^\dagger_\mu(x-\nu)U_\nu(x-\nu), \end{aligned} \tag{10}$$

such that the 1×1 loops,

$$\begin{aligned} W^R_{x,\mu\nu}(U) &= U_\mu(x)S^R_{x,\mu\nu}(U_f) \quad \text{and} \\ W^L_{x,\mu\nu}(U) &= U_\mu(x)S^L_{x,\mu\nu}(U_f), \end{aligned} \tag{11}$$

have the same gauge transformation as g_x. One can then define a covariant algebra-valued object as, e.g.,

$$\begin{aligned} G_{x,\mu} = &\sum_{\nu\neq\mu}\alpha^{(1)}_{\mu\nu}\mathcal{P}(W_{x,\mu\nu}(U)) \\ &+ \sum_{\nu,\rho\neq\mu}\alpha^{(2)}_{\mu\nu\rho}\mathcal{P}(W_{x,\mu\nu}(U)W_{x,\mu\rho}(U)), \end{aligned} \tag{12}$$

where $W_{x,\mu\nu} = W^R_{x,\mu\nu} + W^L_{x,\mu\nu}$, and $\mathcal{P}(W)$ is the gauge-covariant traceless anti-Hermitian projection of W. Moreover, $\alpha^{(1)}_{\mu\nu}$ and $\alpha^{(2)}_{\mu\nu\rho}$ are $d-1$ and $(d-1)^2$ trainable parameters in d spacetime dimensions for fixed μ, respectively. Any polynomial function of $G_{x,\mu}$ with coefficients that are arbitrary function of $\mathrm{Tr}[G_{x,\mu}G^\dagger_{x,\mu}]$ is thus gauge covariant and can be used to construct $g_x(U)$. One choice of such a construction is:

$$g_x(U_f, U_\mu(x_a)) = G_{x,\mu} \times f(\mathrm{Tr}[G_{x,\mu}G^\dagger_{x,\mu}]), \tag{13}$$

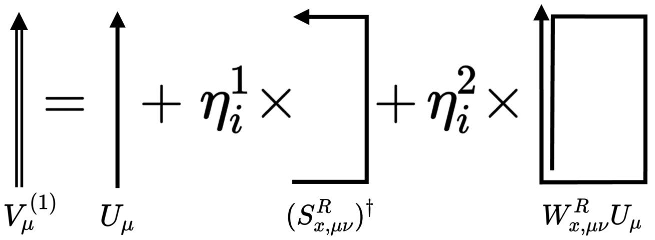


FIG. 1. Sketch of the recursive transformation, Eq. (14), to build generic Wilson loops in the residual layers.

where $f(x)$ is, e.g., a ratio of polynomials—see the Appendix for an example.

A useful modification to this construction is to consider Wilson loops that are larger than 1×1. Sums of such loops can be constructed iteratively, by repeatedly adding together links and staples which transform in the same way, and finally computing a 1×1 loop. This is inspired by similar transformations used in Refs. [41,56] and resembles the learned smearing of Ref. [57]. This gauge-equivariant "convolution" can be written explicitly as the recursion

$$V^{(i+1)}_\mu = V^{(i)}_\mu + \sum_{\substack{\rho\neq\mu,\\ \ell}}\eta^\ell_{i,\rho}\left(R^\ell_{\mu\rho}(V^{(i)}) + L^\ell_{\mu\rho}(V^{(i)})\right), \tag{14}$$

where

$$V^{(0)}_\mu(x) = \begin{cases} U_\mu(x) & U_\mu(x)\,\text{is frozen}, \\ 0 & U_\mu(x)\,\text{is active}, \end{cases} \tag{15}$$

$\eta^\ell_{i,\rho}$ are trainable coefficients, and L^ℓ and R^ℓ label generic staplelike objects that transform in the same way as the gauge links. Here we use two explicit choices, $R^1_{\mu\nu} = (S^R_{x,\mu\nu})^\dagger$ in Eq. (10) and $R^2_{\mu\nu} = W^R_{x,\mu\nu}U_\mu$, and similarly for $L^\ell_{\mu\nu}$; see Fig. 1. Note that in Eq. (14), these objects are computed using the variables $V^{(i)}$. After iterating, $V^{(i)}$ is not an element of the gauge group, but this is not important since ultimately there is a projection to the algebra to construct G_μ in Eq. (12).

The iterative procedure in Eq. (14) can be used to construct expressive residual layers. After applying n_{pt} iterations of Eqs. (14) and (15), the resulting values of $V^{(n_{\mathrm{pt}})}$ can be used to construct the quantity $g_x(V^{(n_{\mathrm{pt}})}, U_\mu(x_a))$ that enters in the transformation of the residual layer defined in Eq. (9). Specifically, the convoluted frozen links, $V^{(n_{\mathrm{pt}})}$, are used to construct the staples in Eq. (11) instead of U_f.

III. EXAMPLE APPLICATIONS

Physics contexts in which derivatives of the form of Eq. (5) arise are ubiquitous; here we discuss three examples. First, derivatives with respect to the gauge coupling β can be used to constrain continuum extrapolations. Second, matrix elements may be computed using Feynman–Hellmann techniques, where derivatives with respect to action parameters correspond to single insertions of the corresponding operator. Second-order derivatives using Feynman–Hellmann also access physically relevant processes, e.g., Compton scattering. Third, derivatives with respect to the quark mass can be employed to constrain chiral extrapolations or in calculations of, e.g., sigma terms. This section presents numerical demonstrations using flows to improve estimates of these three kinds of derivatives.

The flow models used in these applications are summarized in Table I. All flow models have been optimized using

TABLE I. Summary of flow models used in this work. All flow models have been trained on a hypercubic lattice volume of size 4^4, while the evaluation lattice volume at which the flows are used (Eval. vol.) is given explicitly in the table.

Model	Prior type	Parameters	Target type	Parameters	Train ESS	Eval vol	ESS
A	Pure Gauge SU(3)	$\beta = 6.02$	Pure Gauge SU(3)	$\beta = 6.03$	99.72%	16^4	67%
B1	Pure Gauge SU(3)	$\beta = 6.00$	Feynman–Hellmann	$\beta = 6.00, \lambda = +0.01$	99.4%	16×8^3	84%
B2	Pure Gauge SU(3)	$\beta = 6.00$	Feynman–Hellmann	$\beta = 6.00, \lambda = -0.01$	99.4%	16×8^3	84%
C	$N_f = 2$ QCD	$\beta = 5.60$, $\kappa = 0.153$	$N_f = 2$ QCD	$\beta = 5.60$, $\kappa = 0.1545$	99.2%	8^4	48%

path gradients [51] as described in Ref. [43]. Gauge field samples for both training and evaluation are obtained using standard Markov Chain Monte Carlo methods, specifically the (pseudo-)heatbath algorithm with overrelaxation [58–62] for Yang–Mills theory and the Hybrid/Hamiltonian Monte Carlo [63] (HMC) algorithm for QCD.

A. Continuum limit of gauge theories

One application in lattice QCD for flow-correlated ensembles is in taking the continuum limit. For a numerical demonstration, we consider gradient flow scales.

We use the pure-gauge SU(3) theory, with action

$$S_g(U) = -\frac{\beta}{N_c} \mathrm{Tr}\,\mathrm{Re} \sum_{\mu>\nu} U_{\mu\nu}, \tag{16}$$

where β is the inverse squared bare gauge coupling and $U_{\mu\nu}$ is the plaquette. The continuum limit of lattice spacing $a \to 0$ corresponds to $\beta \to \infty$.

One class of observables is obtained by using the gradient flow; in particular, a scale t_c can be defined implicitly from

$$\langle t^2 E(t) \rangle|_{t=t_c} = c, \tag{17}$$

where c is a numerical constant, and $E(t)$ is the energy density at flow time t, for which we use the plaquette definition; see Eq. (3.1) in Ref. [64]. The choice $c = 0.3$ defines the scale $t_{0.3}$, often referred to as "t_0." One can compute the ratio of two gradient flow scales $t_{0.3}/t_{0.35}$, which can be related to the ratio of the strong coupling at two different energy scales [64]. The continuum limit of this quantity takes the form

$$\left.\frac{t_{0.3}}{t_{0.35}}\right|_{\mathrm{lat}} = \left.\frac{t_{0.3}}{t_{0.35}}\right|_{\mathrm{cont}} + k_1 \frac{a^2}{t_{0.3}} + \cdots, \tag{18}$$

where k_1 is a dimensionless constant, the ellipsis indicates higher orders in a^2, the subscripts "lat" and "cont" refer to finite-a and continuum values, and discretization effects are parameterized by powers of $a^2/t_{0.3}$.

The standard approach for performing a continuum extrapolation in lattice QCD relies on computing the desired quantity at several different lattice spacings using independent ensembles and extrapolating. This method can be improved by additional constraints on such an extrapolation in the form of derivatives

$$k(a^2) = \frac{d(t_{0.3}/t_{0.35})}{d(a^2/t_{0.3})} = k_1 + O(a^2). \tag{19}$$

Without generating more ensembles, this derivative can be computed using finite differences combined with ϵ reweighting or with flows to nearby values of the lattice spacing, or equivalently, values of the bare gauge coupling β:

$$k(a^2) \simeq \frac{\frac{t_{0.3}}{t_{0.35}}|_{\beta+\Delta\beta} - \frac{t_{0.3}}{t_{0.35}}|_{\beta}}{\frac{a^2}{t_{0.3}}|_{\beta+\Delta\beta} - \frac{a^2}{t_{0.3}}|_{\beta}}. \tag{20}$$

Note that the gradient flow scales t_c are derived quantities, so we use the notation "$|_\beta$" to indicate that they have been computed in a theory with the given β.

To demonstrate the advantage gained by using flows, we compute Eq. (20) using ϵ reweighting [Eq. (6)] and the flowed approach [Eq. (7)] and compare. For this test, we use 96k configurations at $\beta = 6.02$ on volume $L^4 = 16^4$. For ϵ reweighting, we use a step of $\Delta\beta = 0.001$, leading to an ESS of 96% on this ensemble. For the flowed approach, we use Model A of Table I, which maps from $\beta = 6.02$ to $\beta = 6.03$, that is $\Delta\beta = 0.01$. This model achieves an ESS of 67%, which is significantly higher than direct reweighting, which has an ESS of 2% at the same target parameters. Using these approaches, we find

$$\begin{aligned} \text{Flow:}\ k(a^2) &= -0.0167(41), \\ \epsilon\,\text{reweighting:}\ k(a^2) &= -0.0208(63), \end{aligned} \tag{21}$$

that is, the statistical uncertainly using ϵ reweighting is 50% larger than that obtained with flows. In other words, one needs about 2.4× fewer samples using the flow method as compared with ϵ reweighting to achieve the same statistical uncertainty.

Assuming that cutoff effects are already in the linear regime at this value of the lattice spacing, one can use this procedure to perform a simple continuum extrapolation of the ratio of flow scales. The continuum-extrapolated

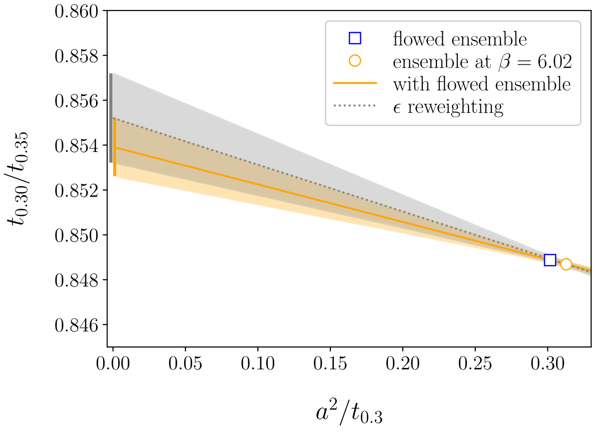


FIG. 2. Continuum extrapolation of the ratio of two gradient flow scales $t_{0.3}/t_{0.35}$, using the quantity in the numerator to set the scale. Two methods are shown: ϵ reweighting (dotted grey line), and using a flowed ensemble (solid orange line). Statistical uncertainties are displayed as bands.

results show the same hierarchy of uncertainties as in Eq. (21):

$$\begin{aligned} \text{Flow:}\ t_{0.3}/t_{0.35}|_{\text{cont}} &= 0.8539(13), \\ \epsilon\,\text{reweighting:}\ t_{0.3}/t_{0.35}|_{\text{cont}} &= 0.8552(20). \end{aligned} \tag{22}$$

These results are shown in Fig. 2 for the two methods.

B. Hadron structure with Feynman–Hellmann techniques

Another promising application of machine-learned flows is in the calculation of matrix elements via the Feynman–Hellmann (FH) approach—see Refs. [65–68] for recent applications. In this framework, a matrix element

$$T_h = \langle h|\mathcal{O}|h\rangle, \tag{23}$$

where h is a stable hadron at rest and $\mathcal{O}$ is the operator of interest projected to zero momentum, is computed by taking derivatives with respect to a parameter in the action. Specifically, adding the operator to the action as

$$S \to S_\lambda = S + \lambda\mathcal{O}, \tag{24}$$

the matrix element can be obtained as

$$T_h = \frac{1}{2M_h}\frac{dM_h}{d\lambda}\bigg|_{\lambda\to 0}, \tag{25}$$

where M_h is the hadron mass. In practice, this can be estimated using a finite-difference approximation of the derivative, e.g.,

$$T_h = \frac{1}{2M_h(0)}\frac{M_h(+\lambda) - M_h(-\lambda)}{2\lambda} + O(\lambda^2). \tag{26}$$

Other improved finite-difference approximations or modeling-based approaches may also be used to better control the $O(\lambda^2)$ bias. As a numerical demonstration, we consider a Feynman–Hellmann calculation of the gluon momentum fraction of the pion in the quenched approximation of lattice QCD, similar to Ref. [65]. In this case the operator $\mathcal{O}$ may be defined as

$$\mathcal{O} = -\frac{\beta}{N_c}\mathrm{TrRe}\left(\sum_i U_{i0} - \sum_{i<j} U_{ij}\right), \tag{27}$$

where $i, j \in (1, 2, 3)$, which is a discretization of the Energy-Momentum-Tensor (EMT). The matrix element can then be related to the gluon momentum fraction of the hadron $\langle x\rangle_g$ by

$$\frac{dM_h}{d\lambda}\bigg|_{\lambda\to 0} = -\frac{3M_h}{2}\langle x\rangle_g^{\text{latt}}, \tag{28}$$

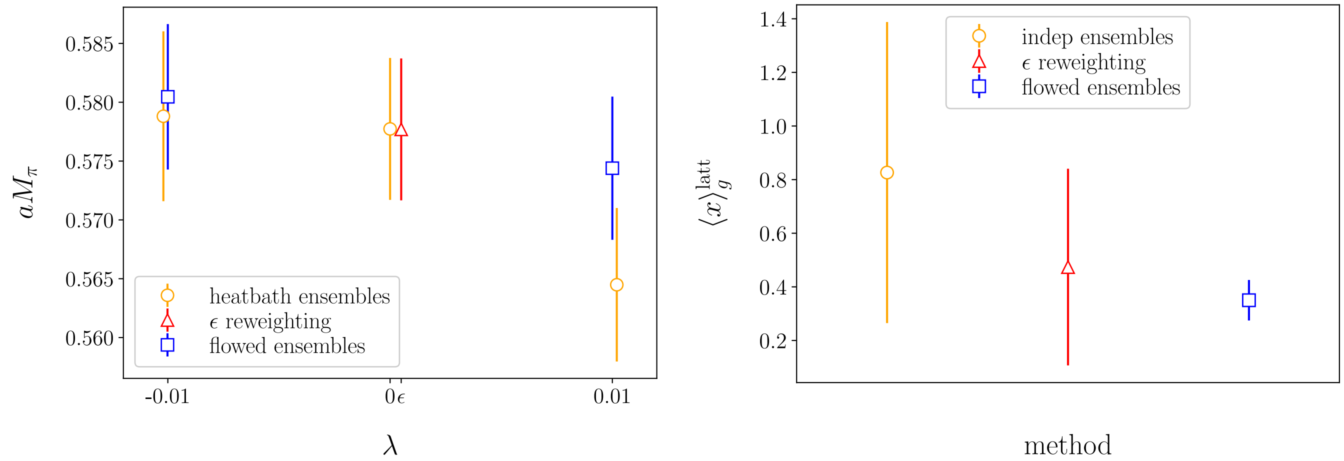
where the superscript "latt" emphasizes that it is a bare matrix element. When adding this operator to the gauge action with a small parameter λ, the full action can be seen as an anisotropic action with different couplings for the temporal and spatial plaquettes:

$$S_\lambda = -\frac{\beta}{N_c}(1+\lambda)\mathrm{Re\,Tr}\sum_i U_{i0} - \frac{\beta}{N_c}(1-\lambda)\mathrm{Re\,Tr}\sum_{i<j} U_{ij}. \tag{29}$$

It is therefore possible to use flow transformations to map from the isotropic pure gauge action at $\lambda = 0$ to nonzero values of λ. This target is referred to as "Feynman–Hellmann" in Table I.

We test the flowed approach by computing the difference in Eq. (26) using an ensemble generated at $\lambda = 0$ and flowed to nonzero ±λ values. The choice $\lambda = 0.01$ is small enough that $O(\lambda^2)$ discretization artifacts in the derivative are negligible; compare to the results in Ref. [65]. We train two flows, B1 and B2 in Table I, which achieve an ESS of 84% at the evaluation volume, cf. the direct reweighting ESS of around 2% at the same values of λ. The target parameters are matched to Ref. [65], albeit at a smaller volume. The value of $\beta = 6$ corresponds to a lattice spacing of $a \simeq 0.09$ fm (using the Sommer radius to set the scale [69]), and the hopping parameter κ in the quenched Dirac operator—related to the bare quark mass as $\kappa = 1/(2m_0 + 4)$—is taken to be $\kappa = 0.132$. The lattice spatial and temporal extent are $L = 8$ and $T = 16$, such that $M_\pi L > 4$. For the purpose of this demonstration, we approximate the pion masses using the effective mass at the center of the lattice,

$$\cosh aM_\pi = \frac{C_\pi(T/2+1) + C_\pi(T/2-1)}{2C_\pi(T/2)}, \tag{30}$$



(a) (b)

FIG. 3. (a) Pion mass in lattice units as a function of the coupling to the gluonic energy-momentum tensor λ. Marker shapes denote how the ensembles were obtained: orange circles for heatbath ensembles at fixed values of λ, blue squares for ensembles flowed from $\lambda = 0$, and red triangles when using configurations generated at $\lambda = 0$ and reweighted to $\lambda = \epsilon = 10^{-4}$. The pion mass is evaluated in quenched lattice QCD at $\beta = 6.0$, $\kappa = 0.132$, $L = 8$, and $T = 16$. (b) Bare gluon momentum fraction of the pion from Eq. (28) using a finite-difference approximation computed using the three different methods: independent heatbath ensembles, ϵ reweighting, and correlated flowed ensembles.

where $C_\pi(t)$ is the pion correlator. In physical units, $M_\pi \simeq 1.2$ GeV.

For evaluation, 14k gauge-field configurations are generated using one heatbath step with five overrelaxation steps between measurements for each independent ensemble. Correlation functions are measured with four smeared sources per configuration with point sinks, using Chroma [70]. The pion mass as a function of λ is shown in Fig. 3a, as determined using ϵ reweighting, independent ensembles, and flowed ensembles. Since the flow model quality at the volume of interest is very high, uncertainties in the observables computed on flowed ensembles are very similar to those computed using ensembles generated with heatbath.

The physical quantity of interest, $\langle x\rangle_g^{\text{latt}}$, depends on the difference between the pion mass determined at different values of λ. When this difference is computed using independent ensembles, statistical uncertainties add in the usual way, and the error in the correlated difference is larger than that of each $M_\pi(\lambda)$ estimate. In contrast, for flowed ensembles or ϵ reweighting, cancellations of correlated fluctuations significantly reduce the variances. This can be seen in Fig. 3b, which shows $\langle x\rangle_g^{\text{latt}}$ computed following the different methods outlined in Sec. II. The use of flowed ensembles reduces the uncertainty by a factor of ≃7 with respect to independent ensembles, and ≃5 with respect to ϵ reweighting (using $\lambda = 10^{-4}$ with an ESS of 99.93%). Thus, incorporating flows into this calculation leads to a reduction of more than 20× in the number of configurations necessary to achieve the same statistical error.

It is also possible to compute the second derivative of M_π with respect to λ, which can be approximated as

$$\left.\frac{d^2M_\pi}{d\lambda^2}\right|_{\lambda=0} \simeq \frac{M_h(+\lambda) + M_h(-\lambda) - 2M_h(0)}{\lambda^2}. \quad (31)$$

While for the particular case of the gluon energy-momentum tensor this derivative is not physically relevant, second derivatives are related to matrix elements of two-current insertions—see for instance Compton scattering applications [71,72]. Using the same three methods as for the first derivative, we find:

$$\begin{aligned} \text{Flow:}\ \left.\frac{d^2M_\pi}{d\lambda^2}\right|_{\lambda=0} &= -6(15), \\ \epsilon\,\text{reweighting:}\ \left.\frac{d^2M_\pi}{d\lambda^2}\right|_{\lambda=0} &= -140(110), \\ \text{Indep ens:}\ \left.\frac{d^2M_\pi}{d\lambda^2}\right|_{\lambda=0} &= -120(150). \end{aligned} \quad (32)$$

All the determinations yield numbers that are zero within two standard deviations, but the relative magnitude of the uncertainties can nevertheless be used to assess the advantage of the flowed approach. In particular, for the second derivative, the error reduction when using flows is larger than for the case of the first derivative, a factor of 7–10 smaller than that obtained using ϵ reweighting or independent ensembles. This, in turn, leads to requiring one to two orders of magnitude fewer configurations to achieve some target statistical precision.

C. Mass dependence of QCD observables

As a third example, we compute derivatives with respect to the quark mass in QCD with $N_f = 2$ unimproved Wilson fermions. As a simple demonstration, we work directly with the action including the exact fermion determinant,

$$S(U) = S_g(U) - \log\det D_w[U] D_w^\dagger[U], \qquad (33)$$

where $S_g(U)$ is the plaquette gauge action and D_w is the discrete standard Wilson operator. The quark mass enters in the action via the hopping parameter κ. This target is referred to as "$N_f = 2$ QCD" in Table I.

For this test, we compute the derivative of some simple observables (generically labeled as X) with respect to κ, approximated via finite differences:

$$\frac{dX}{d\kappa} \simeq \frac{X(\kappa_2) - X(\kappa_1)}{\kappa_2 - \kappa_1}. \qquad (34)$$

Depending on the observable, such derivatives can be useful, e.g., to extract sigma terms or to constrain chiral extrapolations. Here we specifically consider the average plaquette, the squared topological charge measured using the gradient flow at flow time $t/a^2 = 2$, and gradient flow scales t_c.

We train a flow to map configurations from $\kappa = 0.1530$ to $\kappa = 0.1545$ at $\beta = 5.6$ (Model C in Table I). Such parameters are close to those in Ref. [73]. We use 9k configurations generated using standard HMC with pseudofermions. Note, however, the reweighting factor and KL divergence for each configuration are computed with Eq. (33); this is statistically consistent and introduces no approximations. At the evaluation volume of 8^4, the flow achieves ESS = 48%, which should be compared with the ESS = 28% obtained using direct reweighting to the same target parameters.

The results are given in Fig. 4, which compares the (normalized) values of several observables computed using the two methods, i.e., correlated flowed ensembles and ϵ reweighting (with $\Delta\kappa = 3 \times 10^{-4}$ for an ESS of 95%). At these statistics and for these choices of κ, independent ensembles result in statistical errors $\gtrsim 2\times$ larger than those attained with flows, and we do not display them. In all cases, flows provide a variance reduction and the central values are consistent within a standard deviation with those obtained with independent ensembles, which indicates that systematic errors in the finite-difference approximation of the derivative are not significant in this example. The error reduction varies between observables in the range ~20%–40%. In particular, the largest reduction is seen for the 1×1 plaquette loop, while the smallest is seen for the topological charge. Thus, depending on the observable of interest, one requires a factor of $1.5 - 2\times$ fewer configurations to obtain a comparable statistical error when using flows.

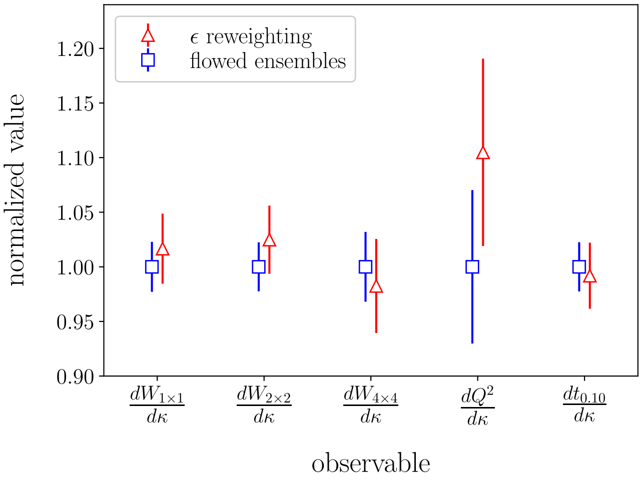


FIG. 4. Illustration of the error reduction in derivatives of observables with respect to the action parameter κ. $W_{n\times n}$ is the average square Wilson loop of size n, Q^2 is the squared topological charge defined via the gradient flow, and t_c labels gradient flow scales, as in Eq. (17). The y-axis shows the values of the observables and their statistical errors normalized to the value obtained with flows. Results that incorporate flows are shown as blue squares, while the errors with ϵ reweighting are denoted by red triangles.

IV. CONCLUSION

In this work, we present the application of machine-learned flows to the computation of observables involving derivatives. Specifically, we use flows to map ensembles between distributions defined by different parameters in the lattice action. By exploiting correlated cancellations of uncertainties between these ensembles, this application has the potential to provide a computational advantage in the evaluation of finite-difference approximations of derivatives.

To illustrate this idea, we showcase three numerical demonstrations in the context of lattice QCD: continuum limit extrapolations, matrix elements using the Feynman–Hellmann approach, and the mass dependence of observables. In all cases, flows provide a reduction of variance, which implies that fewer configurations are needed to achieve the same statistical error. The improvement factor for all demonstrations of this work, defined as the variance reduction in observables computed using flows with respect to ϵ reweighting, is summarized in Fig. 5. These values are in the range of 1.5× for observables in QCD to more than 20× for quantities in the Feynman–Hellmann approach. With higher-quality flow models, these factors can be improved.

This comparison does not account for the differing costs of the different steps in each method, namely generating the initial ensemble with heatbath, applying the flow (in the flowed case), and measuring correlation functions. Of course, the potential advantages of this approach depend sensitively on not only the model used, but on the particular application, the cost of evaluating observables, how autocorrelations are treated, and the precision goal. For a

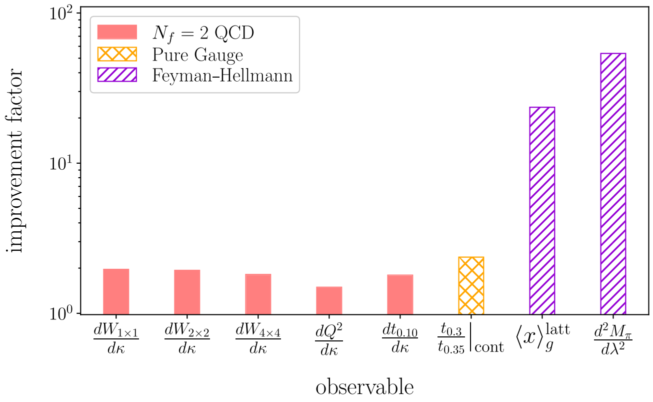


FIG. 5. Summary of the variance reduction in observables computed from derivatives with respect to the action parameters when using flows compared with ϵ reweighting. The improvement factor is defined as the ratio of variances of the observables computed with ϵ reweighting over flows. The label "$N_f = 2$ QCD" denotes derivatives of observables with respect to κ in two-flavor QCD, the label "Pure Gauge" corresponds to the result for the continuum limit extrapolation of gradient flow scales in the pure gauge theory, and the label "Feynman–Hellmann" indicates observables computed using the Feynman–Hellmann approach in quenched QCD.

ballpark comparison, consider the results for the computation of matrix elements in the Feynman–Hellmann approach. In this application, the cost of applying the flow is comparable to the cost of measuring correlation functions, while the cost of a heatbath update is less by an order of magnitude. This amounts to a factor of $\lesssim 3$ increase in computational cost to achieve a variance reduction by a factor of more than 20. This constitutes a real computational advantage of approximately one order of magnitude, neglecting the costs of training. Given expected further improvements through the continued development of flow architectures, these results are promising.

This work focuses on target actions that only depend on the gauge fields, e.g., pure gauge SU(3), quenched QCD, and exact-determinant QCD. To generalize these results to state-of-the-art lattice QCD scales, where the fermion determinant cannot be explicitly evaluated, one must combine these flows with pseudofermion flows for QCD, as explored in Refs. [18,41,42].

As flow model technology for lattice QCD continues to advance, applications of correlated ensembles could be extended to compute other interesting quantities, such as sigma terms of hadrons or observables in QED + QCD. If the success seen in the proof-of-principle applications of this work can be achieved in such contexts, it holds the potential to drive substantial advances in the field.

ACKNOWLEDGMENTS

We thank Michael Albergo, Kyle Cranmer, and Ross Young for useful discussions. RA, DCH, FRL, PES, and J. M. U. are supported in part by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under Grant Contract No. DE-SC0011090. PES is additionally supported by the U.S. DOE Early Career Award DE-SC0021006, by a NEC research award, and by the Carl G. and Shirley Sontheimer Research Fund. F. R. L. acknowledges support by the Mauricio and Carlota Botton Fellowship. G. K. was supported by the Swiss National Science Foundation (SNSF) under Grant No. 200020_200424. This manuscript has been authored by the Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. This work is supported by the U.S. National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, http://iaifi.org/) and is associated with an ALCF Aurora Early Science Program project, and used resources of the Argonne Leadership Computing Facility which is a DOE Office of Science User Facility supported under Contract DEAC02-06CH11357. The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center [74] for providing HPC resources that have contributed to the research results reported within this paper. Numerical experiments and data analysis used PyTorch [75], JAX [76], Haiku [77], Horovod [78], NumPy [79], and SciPy [80]. Figures were produced using Matplotlib [81].

APPENDIX: DETAILS OF MODELS

In this appendix, we provide some additional details of the models of this work and the scheme used to train them. It is important to stress that the hyperparameters and training schemes of these models have not been fine-tuned to be optimal, but they suffice for the present demonstration. It is therefore likely that the model quality can be increased with further training or simple modifications of the hyperparameters.

The layers considered in this work use a ratio of polynomials

$$f(x) = \frac{1}{1+2x}\frac{a_0 + a_1 x}{b_0 + b_1 x} \tag{A1}$$

to construct g_x in Eq. (13), where a_i and b_i are trainable parameters.

All models have $n_{\mathrm{pt}} = 6$, where n_{pt} is the number of iterations of Eq. (14) in each layer. This choice has been found to be empirically better than lower values of n_{pt}. In models A, B1, and B2 we alternate the masking pattern between mod 2 or mod 4, since empirically this results in slight improvements compared to just using the mod 2 masking at the same computational cost (a mod 4 stack is

TABLE II. Additional details of the flow models of this work. “M2” and “M4” refer to a masking pattern modulo 2 or modulo 4, respectively, as described in Sec. II C.

Model	Number of layers	Masking patterns	Number of params.	Gradient steps	Learning rate	Training batch size
A	96	$(\mathrm{M2}+\mathrm{M4})\times 4$	16k	12000	10^{-4}	2048
B1	72	$(\mathrm{M2}+\mathrm{M4})\times 3$	12k	2100	10^{-3}	512
B2	72	$(\mathrm{M2}+\mathrm{M4})\times 3$	12k	2100	10^{-3}	512
C	88	$\mathrm{M2}\times 11$	15k	1700	1.5×10^{-3}	960

computationally equivalent to two mod 2 stacks). The model architectures are shown in Table II.

The models are optimized by minimizing the reverse KL divergence:

$$D_{\mathrm{KL,rev}} = \frac{1}{B}\sum_{i=1}^{B}[\log q(U_i) + S(U_i)] + \mathrm{const}, \tag{A2}$$

where the sum runs over the B configurations in a batch and the (unknown) normalization constant need not be evaluated for optimization. Samples from the prior distribution are generated using heatbath/overrelaxation (pure gauge) or HMC (QCD). The training scheme consists of a constant learning rate for a fixed number of gradient steps. We use a constant batch size to train each model. Between gradient steps, each configuration in the batch is evolved independently using the corresponding update algorithm. These details are summarized in Table II.

In all cases, we use path gradients, which are implemented by computing the gradients for optimization using the path derivative rather than the total derivative:

$$\frac{d\log q(U)}{d\theta} \to \frac{\partial \log q(U)}{\partial U}\frac{dU}{d\theta}. \tag{A3}$$

This reduces the variance of the gradients without changing their expectation. See Ref. [51] for more details.

These models have been trained for different wall times: ten days using six nodes with eight NVIDIA A100 GPUs each for model A, two days using two nodes for models B1 and B2, and two days using four nodes for model C. Note that no attempts have been made to optimize either the training procedure nor implementation of the approach to reduce training times.

A sufficient condition to guarantee invertibility of the residual layers (Lipschitz condition) is

$$\|g_x(V_1) - g_x(V_2)\| < \|V_1 - V_2\|, \tag{A4}$$

where $\|\cdot\|$ denotes the matrix norm. This is not explicitly enforced in the transformations used in this work, but we have not detected any violations in trained models. See Appendix B of Ref. [82] for a discussion on the Lipschitz condition.

[1] P. Boyle *et al.*, in Snowmass 2021 (2022), 3, arXiv:2204.00039.

[2] A. S. Kronfeld *et al.* (USQCD Collaboration), arXiv:2207.07641.

[3] D. Boyda *et al.*, in 2022 Snowmass Summer Study (2022), 2, arXiv:2202.05838.

[4] K. Cranmer, G. Kanwar, S. Racanière, D. J. Rezende, and P. E. Shanahan, Nat. Rev. Phys. **5**, 526 (2023).

[5] D. Bachtis, G. Aarts, and B. Lucini, Phys. Rev. D **103**, 074510 (2021).

[6] S. Calì, D. C. Hackett, Y. Lin, P. E. Shanahan, and B. Xiao, Phys. Rev. D **107**, 034508 (2023).

[7] C. Lehner and T. Wettig, arXiv:2304.10438.

[8] L. Wang, G. Aarts, and K. Zhou, arXiv:2309.17082.

[9] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Phys. Rev. Lett. **126**, 032001 (2021).

[10] D. J. Rezende and S. Mohamed, arXiv:1505.05770.

[11] L. Dinh, J. Sohl-Dickstein, and S. Bengio, arXiv:1605.08803.

[12] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, J. Mach. Learn. Res. **22**, 1 (2021).

[13] S.-H. Li and L. Wang, Phys. Rev. Lett. **121**, 260601 (2018).

[14] M. S. Albergo, G. Kanwar, and P. E. Shanahan, Phys. Rev. D **100**, 034515 (2019).

[15] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Phys. Rev. Lett. **125**, 121601 (2020).

[16] D. Boyda, G. Kanwar, S. Racanière, D. J. Rezende, M. S. Albergo, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Phys. Rev. D **103**, 074504 (2021).

[17] D. C. Hackett, C.-C. Hsieh, M. S. Albergo, D. Boyda, J.-W. Chen, K.-F. Chen, K. Cranmer, G. Kanwar, and P. E. Shanahan, arXiv:2107.00734.

[18] M. S. Albergo, G. Kanwar, S. Racanière, D. J. Rezende, J. M. Urban, D. Boyda, K. Cranmer, D. C. Hackett, and P. E. Shanahan, Phys. Rev. D **104**, 114507 (2021).

[19] M. S. Albergo, D. Boyda, D. C. Hackett, G. Kanwar, K. Cranmer, S. Racanière, D. J. Rezende, and P. E. Shanahan, arXiv:2101.08176.

[20] M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, Phys. Rev. D **106**, 014514 (2022).

[21] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller, and P. Kessel, Phys. Rev. E **101**, 023304 (2020).

[22] S. Foreman, X.-Y. Jin, and J. C. Osborn, in *9th International Conference on Learning Representations* (2021), arXiv:2105.03418.

[23] S. Foreman, T. Izubuchi, L. Jin, X.-Y. Jin, J. C. Osborn, and A. Tomiya, Proc. Sci., LATTICE2021 (**2022**) 073.

[24] S. Foreman, X.-Y. Jin, and J. C. Osborn, Proc. Sci., LATTICE2021 (**2022**) 508.

[25] L. Del Debbio, J. M. Rossney, and M. Wilson, Phys. Rev. D **104**, 094507 (2021).

[26] M. Gabrié, G. M. Rotskoff, and E. Vanden-Eijnden, Proc. Natl. Acad. Sci. U.S.A. **119**, e2109420119 (2022).

[27] P. de Haan, C. Rainone, M. C. N. Cheng, and R. Bondesan, arXiv:2110.02673.

[28] S. Lawrence and Y. Yamauchi, Phys. Rev. D **103**, 114509 (2021).

[29] X.-Y. Jin, Proc. Sci., LATTICE2021 (**2022**) 600.

[30] J. M. Pawlowski and J. M. Urban, Phys. Rev. D **108**, 054511 (2023).

[31] J. Finkenrath, arXiv:2201.02216.

[32] M. Gerdes, P. de Haan, C. Rainone, R. Bondesan, and M. C. N. Cheng, SciPost Phys. **15**, 238 (2023).

[33] A. Singha, D. Chakrabarti, and V. Arora, Phys. Rev. D **107**, 014512 (2023).

[34] A. G. D. G. Matthews, M. Arbel, D. J. Rezende, and A. Doucet, arXiv:2201.13117.

[35] M. Caselle, E. Cellini, A. Nada, and M. Panero, J. High Energy Phys. 07 (2022) 015.

[36] D. Albandea, L. Del Debbio, P. Hernández, R. Kenway, J. Marsh Rossney, and A. Ramos, Eur. Phys. J. C **83**, 676 (2023).

[37] D. Albandea, L. Del Debbio, P. Hernández, R. Kenway, J. M. Rossney, and A. Ramos, Proc. Sci., LATTICE2023 (**2024**) 013.

[38] S. Bacchio, P. Kessel, S. Schaefer, and L. Vaitl, Phys. Rev. D **107**, L051504 (2023).

[39] K. A. Nicoli, C. J. Anders, T. Hartung, K. Jansen, P. Kessel, and S. Nakajima, Phys. Rev. D **108**, 114501 (2023).

[40] A. Singha, D. Chakrabarti, and V. Arora, Phys. Rev. D **108**, 074518 (2023).

[41] R. Abbott, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, S. Racanière, D. J. Rezende, F. Romero-López, P. E. Shanahan, B. Tian, and J. M. Urban, Phys. Rev. D **106**, 074506 (2022).

[42] R. Abbott, M. S. Albergo, A. Botev, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, A. G. D. G. Matthews, S. Racanière, A. Razavi, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, Proc. Sci., LATTICE2022 (**2023**) 036.

[43] R. Abbott, M. S. Albergo, A. Botev, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, A. G. D. G. Matthews, S. Racanière, A. Razavi, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, arXiv:2305.02402.

[44] R. Abbott, M. S. Albergo, A. Botev, D. Boyda, K. Cranmer, D. C. Hackett, A. G. D. G. Matthews, S. Racanière, A. Razavi, D. J. Rezende, F. Romero-López, P. E. Shanahan, and J. M. Urban, Eur. Phys. J. A **59**, 257 (2023).

[45] K. A. Nicoli, C. J. Anders, L. Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, and P. Stornati, Proc. Sci., LATTICE2021 (**2022**) 338.

[46] S. Bacchio, Phys. Rev. D **108**, L091508 (2023).

[47] G. Catumba, A. Ramos, and B. Zaldivar, arXiv:2307.15406.

[48] G. M. de Divitiis, R. Frezzotti, V. Lubicz, G. Martinelli, R. Petronzio, G. C. Rossi, F. Sanfilippo, S. Simula, and N. Tantalo (RM123 Collaboration), Phys. Rev. D **87**, 114505 (2013).

[49] N. Tantalo, Proc. Sci., LATTICE2022 (**2023**) 249.

[50] S. Kullback and R. A. Leibler, Ann. Math. Stat. **22**, 79 (1951).

[51] L. Vaitl, K. A. Nicoli, S. Nakajima, and P. Kessel, arXiv:2207.08219.

[52] P. Bialas, P. Korcyl, and T. Stebel, Comput. Phys. Commun. **298**, 109094 (2024).

[53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[54] W. K. Hastings, Biometrika **57**, 97 (1970).

[55] L. Tierney, Ann. Stat. **22**, 1701 (1994).

[56] M. Favoni, A. Ipp, D. I. Müller, and D. Schuh, Phys. Rev. Lett. **128**, 032003 (2022).

[57] A. Tomiya and Y. Nagai, arXiv:2103.11965.

[58] M. Creutz, Phys. Rev. D **21**, 2308 (1980).

[59] N. Cabibbo and E. Marinari, Phys. Lett. **119B**, 387 (1982).

[60] A. D. Kennedy and B. J. Pendleton, Phys. Lett. **156B**, 393 (1985).

[61] F. R. Brown and T. J. Woch, Phys. Rev. Lett. **58**, 2394 (1987).

[62] S. L. Adler, Phys. Rev. D **37**, 458 (1988).

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

[64] M. Lüscher, J. High Energy Phys. 08 (2010) 071.

[65] R. Horsley, R. Millo, Y. Nakamura, H. Perlt, D. Pleiter, P. E. L. Rakow, G. Schierholz, A. Schiller, F. Winter, and J. M. Zanotti (QCDSF UKQCD Collaborations), Phys. Lett. B **714**, 312 (2012).

[66] M. Batelaan, K. U. Can, R. Horsley, Y. Nakamura, P. E. L. Rakow, G. Schierholz, H. Stüben, R. D. Young, and J. M. Zanotti (QCDSF-UKQCD-CSSM Collaborations), Phys. Rev. D **108**, 034507 (2023).

[67] M. Batelaan *et al.* (QCDSF/UKQCD/CSSM, CSSM, UKQCD, QCDSF Collaborations), Phys. Rev. D **107**, 054503 (2023).

[68] A. Hannaford-Gunn, R. Horsley, H. Perlt, P. Rakow, G. Schierholz, H. Stüben, R. Young, J. Zanotti, and K. U. Can (CSSM/QCDSF/UKQCD Collaborations), Proc. Sci., LATTICE2021 (**2022**) 088.

[69] S. Durr, Z. Fodor, C. Hoelbling, and T. Kurth, J. High Energy Phys. 04 (2007) 055.

[70] R. G. Edwards and B. Joo (SciDAC, LHPC, UKQCD Collaborations), Nucl. Phys. B, Proc. Suppl. **140**, 832 (2005).

[71] K. U. Can *et al.*, Phys. Rev. D **102**, 114505 (2020).

[72] A. Hannaford-Gunn *et al.* (CSSM/QCDSF/UKQCD Collaborations), Proc. Sci., LATTICE2021 (**2022**) 028.

[73] R. Gupta, C. F. Baillie, R. G. Brickner, G. W. Kilcup, A. Patel, and S. R. Sharpe, Phys. Rev. D **44**, 3272 (1991).

[74] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell *et al.*, 2018 IEEE High Performance extreme Computing Conference (HPEC) (2018) pp. 1–6, arXiv:1807.07814.

[75] A. Paszke *et al.*, arXiv:1912.01703.

[76] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, http://github.com/google/jax (2018).

[77] T. Hennigan, T. Cai, T. Norman, and I. Babuschkin, http://github.com/deepmind/dm-haiku (2020).

[78] A. Sergeev and M. Del Balso, arXiv:1802.05799.

[79] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith *et al.*, Nature (London) **585**, 357 (2020).

[80] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, Nat. Methods **17**, 261 (2020).

[81] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).

[82] M. Lüscher, Commun. Math. Phys. **293**, 899 (2010).